

Masteroppgave i matematikdidaktikk  
Matematisk institutt  
Universitetet i Bergen

Programmering og algoritmisk  
tenkning i matematikkundervisning på  
videregående skole

Egil Johnsen Dahle

Hovedveileder: Førsteamanuensis Johan Lie

Bergen, Februar 2021



# Forord

Jeg vil først og fremst takke min veileder Johan Lie som foreslo et aktuelt og spennende masterprosjekt om hvordan en kan undervise i «algoritmisk tenkning» i den videregående skolen. Takk til Johan som har stilt opp og har gitt meg konstruktiv støtte og hjelp gjennom tiden jeg har jobbet med oppgaven.

Dette prosjektet hadde ikke vært mulig uten at jeg fikk samlet gode kvalitative data fra flere klasser. Jeg fikk sett på hvordan elever og lærere i skolen faktisk møter digitale problemstillinger i den nye læreplanen. Jeg ønsker å takke alle som bidro til datainnsamlingen i prosjektet: Takk til lærerne som har latt meg møte og bruke klassene sine og som stilte opp til intervju, men ikke minst takk til elevene i 1T som brukte tid av undervisningen til å delta i intervjuene jeg gjennomførte.

Møte med klassene som deltok var utrolig spennende og inspirerende. Jeg har lenge hatt et mål om å bli lærer. Den praksis jeg har fått som lektorstudent, og møtene med elevene gjennom dette prosjektet, har gjort meg sikrere på at jeg valgte rett da jeg startet på den integrerte lektorutdanningen. Jeg vil takke alle ved UiB som bidrar til det jeg synes er et veldig godt studieprogram. Spesielt takk til Marianne Jensen ved Matematisk institutt som alltid stiller opp når det er praktiske spørsmål knyttet til det å være lektorstudent!

Gjennom arbeidet med oppgaven har jeg også lært at det alltid er kjekt å ha gode hjelpere rundt seg. I mitt tilfelle har jeg to søstre som klarte å bli involvert i prosjektet mitt. Lillesøsteren min Ingrid, som selv er lektorstudent, hjalp til med å formulere brev til skolene, rettet på skrivefeil og kom med gode, konstruktive kommentarer gjennom hele skriveprosessen. Storesøsteren min, Tordis, har i slutten av prosjektet gitt hjelp med struktureringen og formateringen av oppgaven. Takk til dere begge!

Jeg vil også takke foreldrene mine som har hjulpet til i skriveprosessen og som har stilt opp med middagsmat som ikke startet som frossen pizza hver gang jeg har ønsket variasjon.

Sist, men ikke minst, vil jeg også takke alle vennene mine som har vært tålmodige og holdt ut med meg denne høsten. Dere har hørt på når jeg har fortalt om prosjektet og kommet med gode kommentarer og forslag. Spesielt vil jeg takke Lydia og Martin som har vist spesiell interesse for prosjektet mitt.



# Sammendrag

Etter at den nye læreplanen “Fagfornyelsen” har blitt iverksatt, skal algoritmisk tenkning og programmering bli en integrert del av skolehverdagen til elever og lærere i den norske skole. Dette innebærer at elever skal skrive sine egne algoritmer og lage dataprogram som en vesentlig del av matematikken de lærer. Denne oppgaven hadde som mål å få en forståelse for hvordan vi kan innlemme programmering og algoritmisk tenkning i matematikkundervisningen i faget 1T, slik at elevene skal få størst mulig læringsutbytte.

For å undersøke dette ble et undervisningsopplegg gjennomført i tre ulike klasser ved en videregående skole i Vestland fylke. Undervisningsopplegget var en del av algebraundervisningen i faget 1T, teoretisk matematikk for Vg1. Opplegget hadde fokus på at elevene skulle bruke programmeringsspråket Python, med IDLE eller Spyder/Anaconda som tekstredigeringsprogram, for å løse algebraiske oppgaver.

For å samle inn data rundt det gjennomførte opplegget ble kvalitative forskningsmetoder tatt i bruk. Fem av elevene som deltok i undervisningsopplegget og de to lærerne som deltok ble intervjuet i etterkant av opplegget. Under intervjuene ble elevene blant annet bedt om å løse to problem som inkluderte bruk av programmering. Observasjoner fra timen ble også brukt.

Dette prosjektet viser at programmering og algoritmisk tenkning kan innlemmes i dagens matematikkundervisning, men fra dette prosjektet så jeg at det bør innlemmes som et verktøy gjennom hele matematikkfaget. De fleste utfordringene elevene hadde oppsto nemlig når matematikken ble for innviklet, eller når kunnskapene de hadde om programmering og algoritmisk tenkning ikke strakk til. Lærere er viktige for at programmering skal fungere som et godt verktøy i undervisningen. Videre forskning kunne sett på hvordan man kan hjelpe dem å innlemme programmering og algoritmisk tenkning i undervisningen sin.



# Innhold

<b>Forord</b>	<b>i</b>
<b>Sammendrag</b>	<b>iii</b>
<b>Forkortelser</b>	<b>vii</b>
<b>Figurer</b>	<b>ix</b>
<b>Tabeller</b>	<b>xi</b>
<b>1 Introduksjon</b>	<b>1</b>
<b>2 Teori</b>	<b>5</b>
2.1 Datamaskin som hjelpemiddel i undervisningen . . . . .	7
2.2 Læreplanen LK06 og LK20 . . . . .	13
2.3 Grunnleggende Python . . . . .	16
2.4 Kompetanser . . . . .	20
2.4.1 Programmeringskompetanser . . . . .	20
2.4.2 Matematiske kompetanser . . . . .	22
2.4.3 Å tenke algebraisk . . . . .	25
2.4.4 Problemløsning i matematikk . . . . .	27
2.5 Kognitiv utvikling . . . . .	30
2.6 Forskningsintervju . . . . .	34
<b>3 Metode</b>	<b>35</b>
3.1 Undervisning . . . . .	35
3.2 Innsamling av data . . . . .	41
3.2.1 Oppgave 1 . . . . .	41
3.2.2 Oppgave 2 . . . . .	44
<b>4 Analyse og diskusjon</b>	<b>47</b>

<b>5 Konklusjon</b>	<b>59</b>
<b>Kildeliste</b>	<b>61</b>
<b>A Transkripsjon</b>	<b>65</b>
A.1 Intervju L <sub>1</sub> . . . . .	68
A.2 Intervju L <sub>2</sub> . . . . .	70
A.3 Intervju E <sub>1</sub> . . . . .	74
A.4 Intervju E <sub>2</sub> . . . . .	80
A.5 Intervju E <sub>3</sub> . . . . .	86
A.6 Intervju E <sub>4</sub> . . . . .	94
A.7 Intervju E <sub>5</sub> . . . . .	101
<b>B Svar fra NSD</b>	<b>107</b>
<b>C Brev til skole</b>	<b>113</b>
<b>D Intervju informasjonsskriv</b>	<b>117</b>
<b>E Lysbilder brukt i opplegget</b>	<b>123</b>
<b>F Intervjuguide lærer</b>	<b>131</b>
<b>G Intervjuguide elev</b>	<b>135</b>



# Forkortelser

**1T** Teoretisk matematikk for studiespesialiserende Vg1.

**CAS** Computer Algebra Systems, verktøy for å løse algebra-problem på datamaskinen.

**E<sub>1-5</sub>** Anonymisert forkortelse for Elev 1 til og med Elev 5.

**EDB** Elektronisk databehandling - Skolefag på 1980-tallet i programmering.

**IDE** Integrated Development Environment - Et tekstredigeringsverktøy for programmering.

**IT** Informasjonsteknologi-faget for Vg2 (IT1) og Vg3 (IT2).

**L<sub>1</sub>, L<sub>2</sub>** Anonymisert forkortelse for Lærer 1 og Lærer 2.

**LK06** Læreplan kunnskapsløftet, innført 2006.

**LK20** Læreplan fagfornyelsen, delvis introdusert fra høst 2020.

**R1** Realfagsmatematikk for Vg2.

**R2** Realfagsmatematikk for Vg3.

**TikZ** Vektorgrafikk program som kjører i L<sup>A</sup>T<sub>E</sub>X (tekstredigeringsprogrammet dette dokumentet ble skrevet i).

**UDIR** Utdanningsdirektoratet.

**Vg1** Første klasse på videregående.

**Vg2** Andre klasse på videregående.

**Vg3** Tredje klasse på videregående.



# Figurer

1.1	Eksempel på “først inn er først ut” . . . . .	2
1.2	Eksempel på “sist inn er først ut” . . . . .	2
2.1	Skjermklipp fra Scratch . . . . .	8
2.2	Intervju E <sub>4</sub> , linje 5 til 13 . . . . .	9
2.3	Bilde av RaspberryPi 3 model B . . . . .	11
2.4	Utklipp fra EDB lærebok . . . . .	12
2.5	Halveringsmetoden . . . . .	16
2.6	Eksempel på avrundingsfeil ved bruk av flyttall i programmering	17
2.7	Utklipp av Spyder IDE . . . . .	18
2.8	Eksempel på program som finner midtpunktet mellom to punkt.	19
2.9	Fire grunnleggende programmeringskompetanser . . . . .	21
2.10	Utgangspunkt for en mal. . . . .	21
2.11	Niss sine 8 matematiske kompetanser . . . . .	23
2.12	Modell av et koffeinmolekyl, laget i L <sup>A</sup> T <sub>E</sub> X . . . . .	24
2.13	MFA-spiralen (Manipulere, få-en-Forståelse, Artikulere) . . . . .	26
2.14	Tidslinje-graf som viser hvordan en (typisk) elev bruker tiden når hen prøver å løse ikke-standardisert problem . . . . .	28
2.15	Tidslinje-graf som viser hvordan en matematiker jobber med et vanskelig problem . . . . .	29
2.16	Tidslinje-graf som viser hvordan en elev bruker tiden når hen prøver å løse ikke-standardisert problem etter kurs i problem- løsning . . . . .	29
2.17	Fem kuler med ulike farger . . . . .	33
2.18	Første syklus i kombinasjons-oppgaven . . . . .	33
2.19	Andre syklus i kombinasjons-oppgaven . . . . .	33
3.1	Svar fra L <sub>2</sub> . . . . .	36
3.2	Svar fra L <sub>1</sub> . . . . .	36
3.3	Eksempel på bruk av matematiske operatører i kommandolinje til Python . . . . .	38

3.4	Oppgaver fra timen . . . . .	39
3.5	To oppgaver med lineære ligningsett . . . . .	40
3.6	Eksempel på løsning av lineære ligningsett . . . . .	41
3.7	Eksempel på løsning av Oppgave 1 . . . . .	42
3.8	Output fra program i Oppgave 1 . . . . .	43
3.9	Eksempel på løsning av Oppgave 2 . . . . .	45
4.1	Intervju L <sub>1</sub> , linje 7 og 8. . . . .	47
4.2	Intervju L <sub>2</sub> , linje 28 - 30. . . . .	48
4.3	Intervju E <sub>4</sub> , linje 127 - 146. . . . .	50
4.4	Intervju E <sub>3</sub> , linje 62 - 71. . . . .	51
4.5	Intervju E <sub>1</sub> , linje 29 - 38. . . . .	52
4.6	Intervju E <sub>1</sub> , linje 165 - 173. . . . .	53
4.7	Intervju E <sub>2</sub> , linje 181 - 199. . . . .	54
4.8	Intervju E <sub>2</sub> , linje 25 - 40. . . . .	55
4.9	Intervju E <sub>5</sub> , linje 67 - 70. . . . .	56
4.10	Intervju E <sub>3</sub> , linje 97 - 110. . . . .	57

# Tabeller

2.1	Utklipp fra kompetansemålene i LK06 og LK20 . . . . .	15
2.2	Eksempler på Python sine fire primitive typer . . . . .	17
2.3	Matematiske og logiske operasjoner og deres Python-syntaks .	19
2.4	De fire stadiene til Piaget . . . . .	31
3.1	Årsplan for en av klassene, uke 35 - 41. . . . .	37



# Kapittel 1

## Introduksjon

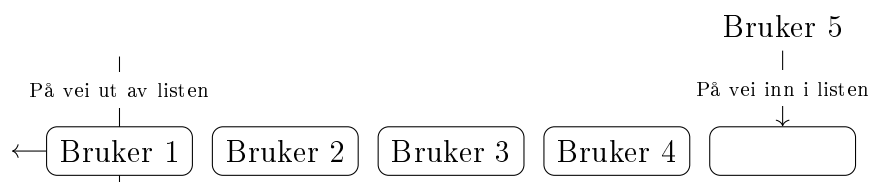
Et av de første minnene mine fra bruk av datamaskin i utdannings sammenheng er fra ca. 3. eller 4. trinn på barneskolen (2003-2004). Jeg skulle skrive en tekst inn på datamaskinen hjemme. Til frustrasjon for moren min leste jeg, og skrev inn en og en bokstav. Mamma spurte storesøsteren min om det var slik vi lærte dette på skolen. Hun verifisert mammas mistanker, at man lærer å skrive inn hele ord og setninger sammenhengende. Siden den gang har teknologien utviklet seg. Mitt forhold til og interesse for teknologi har utviklet seg i samme takt. På videregående valgte jeg derfor IT-faget i andre og tredje klasse (IT1 og IT2 i LK06). Som student ved UiB har jeg også tatt kurs der programmering ble brukt; hovedsakelig programmering i Python, Java og MatLab. Med denne bakgrunnen følte det derfor naturlig å skrive masteroppgave om programmering i matematikkundervisning.

Denne oppgaven ser på hvordan algoritmisk tenkning og programmering kan innlemmes i matematikkundervisningen i den videregående skolen. I læreplanverkets (LK20) overordnet del, som ble introdusert høsten 2020, er digital kompetanse definert som en av fem grunnleggende ferdigheter:

*Læreplanverket definerer fem grunnleggende ferdigheter: lesing, skriving, regning, muntlige ferdigheter og digitale ferdigheter. Disse ferdighetene er del av den faglige kompetansen og nødvendige redskaper for læring og faglig forståelse. De er også viktige for utviklingen av elevenes identitet og sosiale relasjoner, og for å kunne delta i utdanning, arbeid og samfunnsliv («UDIR: Grunnleggende ferdigheter», 2020).*

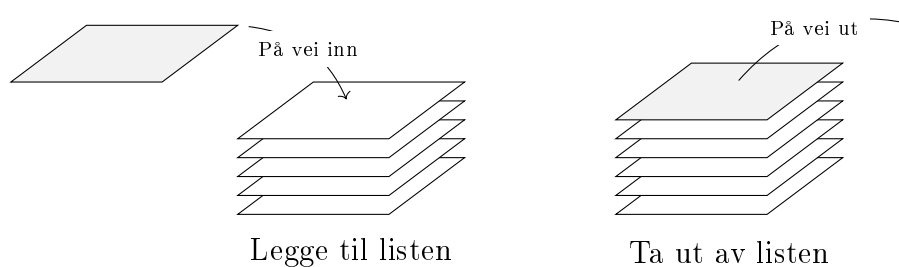
Dette gjenspeiles i matematikkfaget ved at det blant annet finnes kompetansemål om nettopp algoritmer og programmering. En algoritme kan defineres som en *steg-for-steg* oppskrift (Zelle, 2010, s. 4) eller som en prosedyre

for å løse et problem (Sedgewick & Wayne, 2011, s. 4). *Hvorfor* kan algoritmer og programmering ansees som digitale ferdigheter? All digital teknologi bruker algoritmer, ofte uten at vi legger merke til det eller tenker over det. For at teknologien skal kunne bruke disse algoritmene må noen lage programmer som kjører disse. De to neste eksemplene viser to algoritmer som bestemmer måter å lagre data på. Eksemplene viser at det ligger algoritmer bak alt vi kan gjøre i den digitale sfæren. Når prinsippet *førstemann til møll* er viktig lages en “kø” (queue) der første inn er første ut, som illustrert i Figur 1.1 (Sedgewick & Wayne, 2011, s. 126). Denne formen for lagring av data kan sammenlignes med å stå i kø, første personen som kom til køen er første som kommer gjennom. En nettside som selger billetter bruker denne metoden; brukerne plasseres i en kø når de prøver å kjøpe en billett.



Figur 1.1: Eksempel på “først inn er først ut”. Figur laget i TikZ.

Når vi er på nettet kan vi navigere tilbake til nettsider vi har vært på før. Dette er fordi nettleseren (som oftest) lagrer adressene til nettsidene vi har besøkt i en “bunke” (stack) der siste inn er første ut, som illustrert i Figur 1.2 (Sedgewick & Wayne, 2011, s. 127).



Figur 1.2: Eksempel på “sist inn er først ut”. Figur laget i TikZ.

Et program er en tekst med instruksjoner skrevet i et programmeringsspråk. Programmeringsspråket oversetter disse instruksjonene til koder som en datamaskin kan følge (Zelle, 2010, s. 7). Algoritmer er beskrivelser av en prosess; programmering og programmer gjør det mulig å kjøre disse algoritmene på en datamaskin.



Selv om denne oppgaven ser på matematikkundervisningen i videregående skole, er dette bare en fortsettelse av det som undervises i grunnskolen. Under vises to kompetansemål fra LK20 for henholdsvis 4. og 5. trinn på grunnskolen.

4.trinn *Mål for opplæringa er at eleven skal kunne lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer.*

5.trinn *Mål for opplæringa er at eleven skal kunne lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer*

Fagkode: MAT01-05, hentet fra [udir.no](http://udir.no)

Det er viktig å bemerke at LK20 ble innført høsten denne oppgaven ble skrevet, elevene som deltok i dette prosjektet hadde derfor LK06 som læreplan i grunnskolen, en læreplan som ikke inkluderte programmering eller algoritmisk tenkning i matematikkundervisningen. I dette prosjektet skal det prøves ut et opplegg relatert til algoritmisk tenkning og programmering i teoretisk matematikk på videregående. I begynnelsen av prosjektet var problemstillingen:

*Hvordan kan algoritmisk tenkning og programmering innlemmes i matematikkundervisning på videregående trinn?*

Etter hvert som prosjektet utviklet seg ble problemstillingen tilspisset. Det ble utviklet et opplegg som ble testet ut i tre 1T-klasser (Teoretisk matematikk i Vg1). Opplegget la seg tett opp mot det elevene hadde jobbet med tidligere. Dette var i hovedsak tema knyttet opp mot algebra. Under vises problemstillingen som oppgaven tar utgangspunkt i.

*Hvordan kan algoritmisk tenkning og programmering brukes som verktøy for å fremme elevenes forståelse for algebra?*

For å svare på dette, ble det utnyttet kvalitativ datainnsamling gjennom intervju med elever og lærer, i tillegg til observasjon av oppleggets gjennomførelse. Da opplegget ble utarbeidet ble det tatt hensyn til at elevene mest sannsynlig hadde få eller ingen forkunnskaper om programmering eller algoritmisk tenkning.



# Kapittel 2

## Teori

Dette prosjektet gjennomførte undervisning i 1T matematikk. En beskrivelse av det elevene skal lære gjennom dette faget er gitt som en liste med kompetansemål i læreplanen LK20 (UDIR, 2020). Steder markert i kursiv omhandler programmering og digitale hjelpemidler i kompetansemålene for faget.

Mål for opplæringa er at eleven skal kunne

1. formulere og løyse problem ved hjelp av algoritmisk tenking, ulike problemløysingsstrategiar, *digitale verktøy og programmering*
2. lese og forstå matematiske bevis og utforske og utvikle bevis i relevante matematiske emne
3. identifisere variable storleikar i ulike situasjonar, setje opp formlar og utforske desse ved *hjelp av digitale verktøy*
4. utforske strategiar for å løyse likningar, likningssystem og ulikskapar og argumentere for tenkjemåtane sine
5. forklare forskjellen mellom ein identitet, ei likning, eit algebraisk uttrykk og ein funksjon
6. utforske samanhengar mellom andregradslikningar og andregradsulikskapar, andregradsfunksjonar og kvadratsetningane og bruke samanhengane i problemløysing
7. modellere situasjonar knytte til ulike tema, drøfte, presentere og forklare resultatata og argumentere for om modellane er gyldige
8. lese, hente ut og vurdere matematikk i relevante tekstar om ulike tema og presentere relevante berekningar og analysar av resultatata
9. utforske og beskrive eigenskapane ved polynomfunksjonar, rasjonale funksjonar, eksponentialfunksjonar og potensfunksjonar
10. bruke gjennomsnittleg og momentan vekstfart i konkrete døme og gjere greie for den deriverte

11. forklare polynomdivisjon og bruke det til å omskrive algebraiske uttrykk, drøfte funksjonar og løyse likningar og ulikskapar
12. gjere greie for definisjonane av sinus, cosinus og tangens og bruke trigonometri til å berekne lengder, vinklar og areal i vilkårlige trekantar
13. grunngi sinus-, cosinus- og arealsetninga
14. bruke trigonometri til å analysere og løyse samansette teoretiske og praktiske problem med lengder, vinklar og areal

«UDIR: Matematikk T (MAT09-01): Kompetansemål og vurdering», 2020

I min tolkning av læreplanmålet “*formulere og løyse problem ved hjelp av algoritmisk tenking, ulike problemløysingsstrategiar, digitale verktøy og programmering*”, kan, og skal programmering innlemmes der det passer i de andre læreplanmålene. Jeg brukte derfor årsplanen til en av klassene som opplegget ble gjennomført i som utgangspunkt for hva undervisningen skulle omhandle. Dette viser jeg til senere i Kapittel 3.1. De to andre klassene, som hadde en annen lærer, hadde vært gjennom de samme matematiske temaene. Det var tydelig at elevene ikke hadde brukt programmering i 1T undervisningen tidligere, og fra årsplanen var det tydelig at opplegget kunne fokusere på kompetansemålene “*identifisere variable storleikar i ulike situasjonar, sette opp formalar og utforske desse ved hjelp av digitale verktøy*” og “*utforske strategiar for å løyse likningar, likningssystem og ulikskapar og argumentere for tenkjemåtane sine*”. Det elevene ble undervist før opplegget kan gå under de matematiske temaene algebra og aritmetikk.

I resten av Kapittel 2 vil det teoretiske grunnlaget for oppgaven bli dan-  
net. I Kapittel 2.1 vil programmering i matematikkundervisningen settes inn i et historisk perspektiv. Dette dekker teorier om programmering frem til idag. Videre i Kapittel 2.2 ser vi på læreplanene LK06 og LK20 og hvilke plass digitale hjelpemidler har/har hatt i norsk matematikkundervisning de siste årene. I Kapittel 2.3 går vi gjennom den programmeringen elevene skulle lære. Kapitlene 2.4.1 og 2.4.2 går gjennom matematiske kompetanser og kompetanser i programmering. Kapitlene 2.4.3 og 2.4.4 ser på algoritmisk tenkning og problemløsning i matematikkundervisning. Kapittel 2.5 tar for seg kognitive teorier utviklet av Piaget, mens det til slutt er et delkapittel, Kapittel 2.6, som viser hvordan et forskningsintervju kan struktureres.

## 2.1 Datamaskin som hjelpemiddel i undervisningen

Å bruke teknologi som hjelpemiddel i undervisningen er ikke en ny idé. Bruk av digitale verktøy i norsk skole startet på 1980-tallet, gjennom faget elektronisk databehandling (EDB), men dette var ikke en ny tanke på dette tidspunktet. Sitatet under viser at Vannevar Bush, vitenskapelig rådgiver til daværende amerikansk president Roosevelt, i 1945 skrev om datamaskin som et redskap til støtte for læring.

*Many technologies function as scaffolds and tools to help students solve problems. This was foreseen long ago: in a prescient 1945 essay in the Atlantic Monthly, Vannevar Bush, science advisor to President Roosevelt, depicted the computer as a general-purpose symbolic system that could serve clerical and other supportive research function in the sciences, in work, and for learning, thus freeing the human mind to pursue its creative capacities. (National Research Council, 2000, s. 213)*

I matematikkundervisning ble det i perioden fra midten av 1950-tallet til rundt 1970 gjort flere forsøk på å modernisere pensum. Disse forsøkene ble plassert under begrepet “New Math” (Kilpatrick, 2012, s. 563). Kilpatrick påpeker at

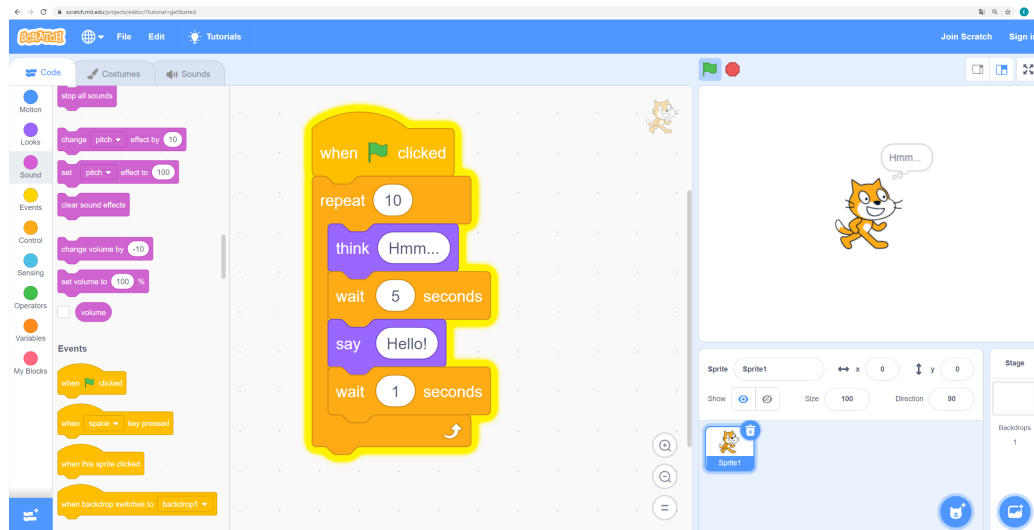
*By the end of the era, they had come to see that much more was required. At the crux of any curriculum change is the teacher. The teacher needs to understand the proposed change, agree with it, and be able to enact it with his or her pupils—all situated in a specific educational and cultural context. (Kilpatrick, 2012, s. 569)*

De fleste endringene som ble gjort overlevde ikke, men matematikkundervisningen etter “New Math” var ikke lik før disse endringene. Dette sitatet er også interessant i Norge idag da det viser til viktigheten av at lærere er med når man utfører endringer i pensum til et fag. Ett eksempel på påvirkning fra denne perioden som er sentralt i denne oppgaven er at før “New Math” var Euklids geometri sentralt i matematikkundervisningen på ungdomstrinnet (secondary school). Selv om mye fra “New Math” reformene ikke ble videreført, er ikke Euklids geometri en like sentral del av matematikkundervisningen lengre; den har blitt erstattet av transformasjons-geometri (transformation geometry), koordinat-geometri og vektorgeometri (Kilpatrick, 2012, s. 569). I boken *Mindstorms* skriver Papert dette om “New Math”:

*The name itself -“New Math”- was a misnomer. There was very little new about the mathematical content: It did not come from a process of invention of children’s mathematics but from a process of trivialization of mathematician’s mathematics.*

(Papert, 1980, s. 67)

Papert introduserte en ny form for geometri som han kalte for *Turtle geometry* som startet med mål om å passe for barn (Papert, 1980, s. 67). Denne geometrien står i kontrast til Euklid sin logiske geometri og Descartes sin algebraiske geometri ved at den er en programmeringsorientert geometri (computational style of geometry) (Papert, 1980, s. 69). På slutten av 1960-tallet ble programmeringsspråket LOGO utviklet, av blant annet Papert, et programmeringsspråk som rettet seg mot matematikkundervisning, og Turtle geometri. Dette språket, og ideene til Papert om Turtle geometri, har videre gitt opphav til programmeringsspråket Scratch (Lewis, 2017, s. 20) som kom på 2000-tallet og er laget for å enkelt lære barn og unge programmering. Fra Scratch sine nettsider linkes det til artikler skrevet om Scratch, den eldste er fra 2007: [BBC News - Free tool offers 'easy' coding](#). Dette programmeringsspråket får man tilgang til fra nettsiden <https://scratch.mit.edu/>, og det kan kjøres direkte i nettleseren.



Figur 2.1: Skjermbilde fra Scratch

Scratch er et programmeringsspråk der de ulike operasjonene er representert ved blokker, der noen blokk-operasjoner ”passer inn” i andre blokker; programmering blir da en byggeprosess. Figur 2.1 viser et bilde av en katt som beveger seg når ”programmeringsblokken” blir aktivert, når man trykker

på det grønne flagget. Dette kan gjøre at veien til å lage programmer blir kortere. Versjoner av Scratch har blitt brukt som undervisningsverktøy for barn helt ned til 5-6 års alderen (Forsström & Kaufmann, 2018, s.24). Kattens bevegelse er Scratch sin versjon av en skilpadde (turtle). Turtle Geometry låner mange ideer fra Euklids og Descarts sine geometrier. Katten er for eksempel definert ved et punkt, som i Euklid sin geometri, men den er samtidig dynamisk ved at den har en retning (Papert, 1980, s. 67). Det er vanskelig å si hvor mye Scratch eller programmering har blitt brukt i norsk skole frem til nå. I et notat fra *Senter for IKT i utdanningen* (nå en del av UDIR) fra 2016 blir det vist til at omtrent 150 ungdomskoler den høsten var med i forsøk der eleven kunne velge valgfag i programmering («Programmering i skolen», 2016, s. 25). Fra intervjuene jeg foretok, som er gjengitt i Tillegg A, kom det frem at tre av elevene hadde programmering/programmeringskurs som del av undervisningen på ungdomsskolen. Transkripsjon 2.2 viser utdrag fra hvordan E<sub>4</sub> beskrev sine tidligere erfaringer med programmering, der E<sub>4</sub> er anonymisert betegnelse for den fjerde eleven jeg intervjuet, M står for meg - intervjueren. Elevene gikk på ungdomsskole i perioden 2017-2020, før den nye læreplanen ble innført.

---

5	M:	Har du noen tidligere erfaring med programmering?
6	E <sub>4</sub> :	Programmerte tre år på ungdomsskolen.
7	M:	Så alle årene.
8	E <sub>4</sub> :	Som valgfag.
9	M:	Som valgfag ja.
10	E <sub>4</sub> :	Ja.
11	M:	Hvilke språk brukte du då? Python.
12	E <sub>4</sub> :	Pythhon, Idle, javascript, Scraeth, micro:bit
13	M:	Ja. Du var innom en del da.
14	E <sub>4</sub> :	Ja.

---

Transkripsjon 2.2: Intervju E<sub>4</sub>, linje 5 til 13, fra Tillegg A.6

*Maker Movement* er en samlebetegnelse på en bevegelse innen bruk av teknologi som har gitt opphav til mange teknologiske hjelpemidler ment for undervisning (Eisenberg, 2017, s.39). Denne bevegelsen oppfordrer til å lage egne løsninger, åpne opp teknologiske gjenstander (“void the warranty” - gjøre garantien ugyldig) og eksperimentere med teknologi og programmering. *Void the warranty* refererer til det å åpne opp teknologien og bruke den på måter den ikke var ment til. Innen dette kommer begrepet *tinkering* som kan beskrives som følgende:

*... tinkering as 'to repair, adjust, or work with something in an unskilled or experimental manner' with synonyms as 'to fiddle, fool, or mess', to 'play, monkey or toy' and finally as 'to handle thoughtlessly, ignorantly, or mischievously'.*  
(Lamers mfl., 2014, s. 139)

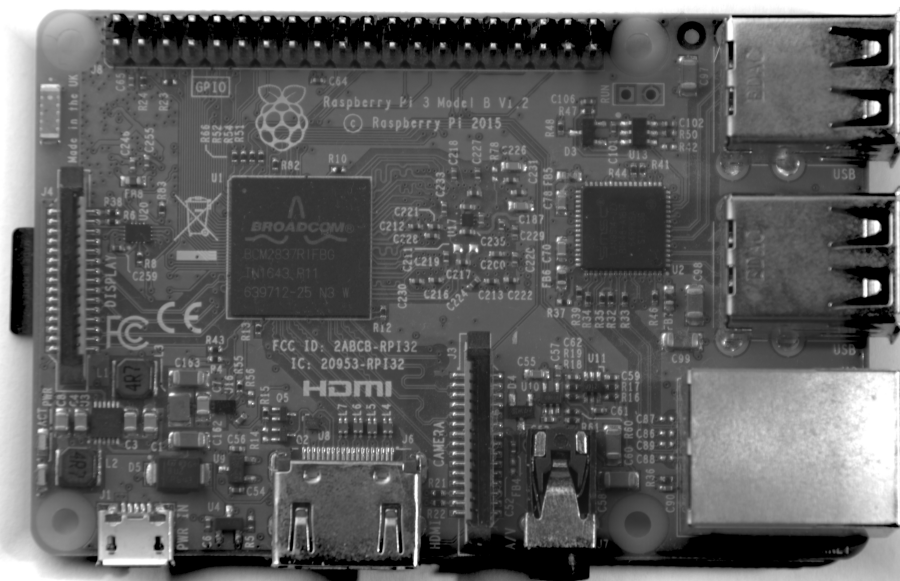
Forsström og Kaufmann (2018) går gjennom litteratur om programmering i matematikkundervisningen. Mange av oppleggene det henvises til har som mål at elevene skal gi ordre til roboter gjennom programmering. Som lektorstudent har jeg møtt lærere som har fått elevene til å bruke ettkorts-datamaskiner som for eksempel Arduino, RaspberryPi og BBC micro:bits. Med ettkorts-datamaskiner menes det at alle komponentene er på ett kort, i motsetning til en datamaskin med separate kort for de ulike komponentene (RAM, HDD/SDD, grafikkort, "motherboard", osv...). Arduino, RaspberryPi og micro:bit kan programmeres ved hjelp av Scratch eller Python. Figur 2.3 viser bilde av en RaspberryPi. I en av praksisperiodene mine opplevde jeg en lærer på videregående som fikk elevene sine til å lodde sensorer til en RaspberryPi for å måle luftkvalitet utendørs. I «Observations on Tinkering in Scientific Education» trekkes det frem et eksempel på et suksessfullt eksperiment der Arduino ble brukt av studenter (academic students). I dette prosjektet ble det utviklet et billig Arduino-basert diagnostisk-polymerasekjedereaksjon-verktøy for å påvise malaria (Lamers mfl., 2014, s. 140). BBC Micro:bit beskrives som en mikrokontroller med et 5×5 led-lys "skjerm", to knapper og den har mulighet til å spille av musikk (Abonyi-Tóth & Pluhár, 2019, s. 191). En micro:bit inneholder akselerometer, kompass og kan kobles til flere sensorer, for eksempel temperatursensor. I tillegg kan en micro:bit kommunisere trådløst med en annen micro:bit. Dette betyr blant annet at man kan ha prosjekt der elever programmerer en micro:bit til å telle skritt ved hjelp av akselerometeret og en annen micro:bit til å vise antall skritt på skjermen (Abonyi-Tóth & Pluhár, 2019, s. 191).

BBC micro:bit er et av mange prosjekt for å lære barn å kode. En av partnerne til dette prosjektet er norske *Lær Kidsa Koding* («Om Lær Kidsa Koding», udatert). På hjemmesiden til *Lær Kidsa Koding* står det

*Lær Kidsa Koding kan legges ned når alle barn kan få sjansen til å bli skapere og ikke bare forbrukere av digitale tjenester. Og da må det ordentlig inn i skolen, hvilket betyr politisk vilje. Vi har en lang vei foran oss.*

«Hvordan Lær Kidsa Koding! ble til», 2015



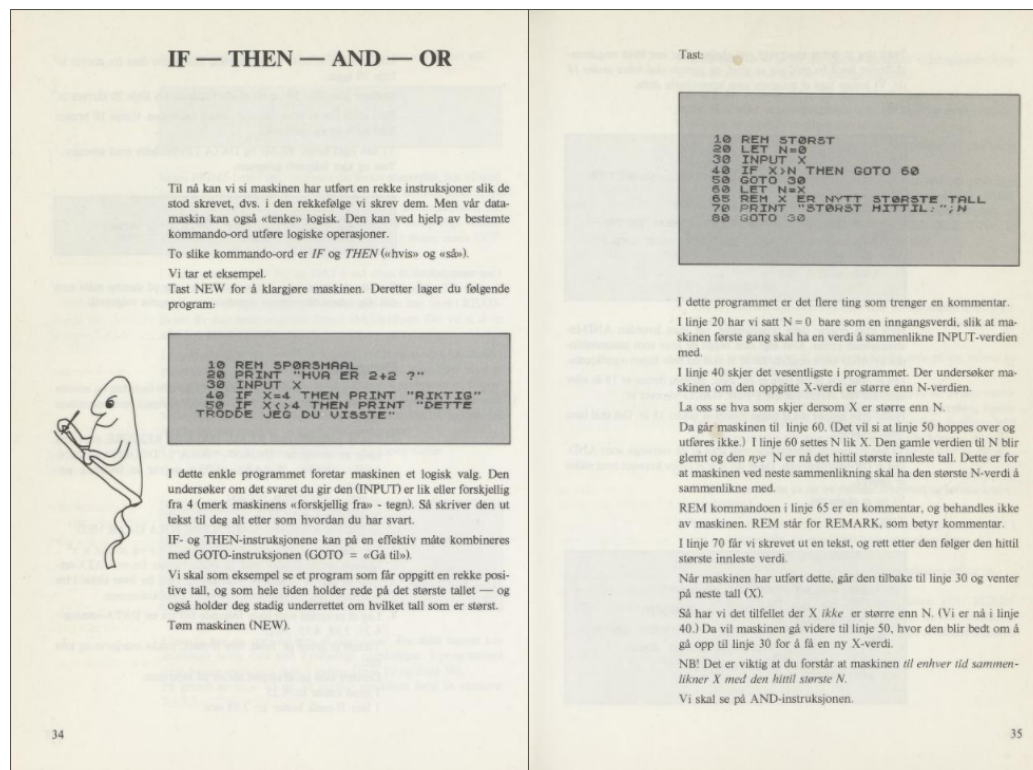


Figur 2.3: Bilde av RaspberryPi 3 model B

I motsetning til Arduino og micro:bit fungerer Raspberry Pi som en fullstendig datamaskin, dersom den er koblet til tastatur, mus og skjerm. Hovedpoenget til Raspberry, foruten lav pris, er at det er enkelt å koble disse maskinene til elektriske motorer og sensorer som kan måle temperatur, bevegelse (f. eks. måle G-kraft, skritteller, m.m.), fuktighet, osv. Det er mulig å få disse maskinene til å kjøre programmer skrevet i Scratch og Python, eksempelvis kommer Raspberrian (det linux-baserte brukersystemet til Raspberry) med begge disse programmeringsspråkene installert. For å få en lekebil til å bevege seg fremover gjennom Python, kan vi tenke oss at kommandoen `output1 = True` sender kommando om at `output1` skal være på, som medfører at et hjul beveger seg. Hva bevegelse betyr gir ikke mening før vi vet hvor lang tid en rotasjon tar, og omkretsen til hjulet. Dette kan eksempelvis brukes i undervisning for å lære om omkrets. Jeg har selv programmert Raspberry Pi til ulike formål: temperaturlogger, sjakklokke og som et kamera. For temperaturmåleren, ble programmering i Python også brukt for å behandle og presentere resultatene på en skjerm.

Bruk av digitale verktøy i norsk skole startet på 80-tallet. Da gjennom faget EDB (elektronisk databehandling). Figur 2.4 viser utklipp av to sider fra en lærebok i EDB. Her er det mange elementer vi kjenner igjen idag, som

bruk av logiske operatører.



Figur 2.4: Side 34 og 35 fra lærebok i EDB fra 1980-tallet (Lervik, 1984)

Fee mfl. (2017) begrunner bruk av algoritmisk tenkning og programmering (computational science), på kryss av de faglige disiplinene, med at denne formen for kunnskap kan føre til en mer effektiv utforskning og løsning av problemer (Fee mfl., 2017, s. 3). Når vi legger vekt på programmering og algoritmisk tenkning i undervisningen er ikke formålet at eleven skal studere datavitenskap senere, men at hen skal få noen grunnleggende ferdigheter i bruk av digitale hjelpemidler. I *Just Give me a Computer and I Will Move the Earth* av Sfard og Leron (1996) viser de hvordan programmering kan endre undervisningen. I vanlig matematikkundervisning har elevene vaner og uvaner som kan begrense hva de får til (Sfard & Leron, 1996, s. 191). En uvane er at elevene ofte tror at det finnes bare *en* korrekt måte å løse et problem på; den som er oppgitt i læreboken eller metoden læreren har vist (Schoenfeld, 1992, s. 343). Når elever skal løse et problem tror de at (a) de skal ha en metode for å løse problemet klart, og (b) metoden skal gi resultater med en gang (Schoenfeld, 1992, s. 343). Schoenfeld, 1989, sier også at de fleste elevene tror at dersom de ikke kan løse et problem på noen få minutter

så kan de ikke løse dette problemet i det hele tatt (Sfard & Leron, 1996, s. 191). Når fokuset er rettet på programmering og algoritmisk tenkning skyves problemet fra å finne løsninger - svar med to streker under -, til å finne metoder som kan løse et problem. Sfard og Leron (1996) rapporterer at endringen mellom "vanlig" matematikkundervisning og matematikkundervisning med programmering er; hvordan elevene prøver å overkomme hindre, antall forsøk der de prøver å løse oppgaven, hvor iherdig de prøver å finne og rette problem og hvor lang tid eleven bruker på et problem (Sfard & Leron, 1996, s. 191). Papert sier også mye av det samme i *Mindstorms*. Han mener at mange barn holdes tilbake i læringen sin, fordi deres syn på læring handler om at man har rett eller feil. Linjen Papert trekker til programmering blir at man nesten aldri har rett første gang man gjør noe når man programmerer (Papert, 1980, s. 37), og at man i prosessen med å programmere - forteller en datamaskin hvordan den skal tenke - ender opp med et mer reflektert syn på hvordan man selv tenker (Papert, 1980, s. 36 - 37).

## 2.2 Læreplanen LK06 og LK20

Opplæring i norsk skole er styrt av læreplanverket. Dette er delt i tre deler; overordnet del, fag- og timefordeling og læreplaner i fag. I denne delen sammenlignes læreplanene LK06 og LK20 i matematikk med hverandre for å se på hvordan den nye læreplanen, LK20, endrer synet på bruk av digitale verktøy i forhold til den utgående læreplanen LK06. Det er læreplanene som bestemmer de spesifikke kompetansene en elev skal ha etter fullført emne.

De siste årene har digitale verktøy i undervisningen vært mye omtalt. Et kjapt søk på stikkordet "nettbrett" på NRK.no leder til en serie med artikler der bruk av digitale hjelpemidler i skolen blir diskutert. Under er et lite utvalg av disse artiklene, hentet 25.09.2020. [Må brukes med fornuft](#) (Melby, 2020), som handler om bruk av nettbrett i barneskolen; [Et naivt og utdatert prosjekt](#) (Falch, 2020) som handler om bruk av digitale hjelpemidler i norsk skole og problem med at barn bruker for mye tid foran skjerm; [Digitale prøvekaniner](#) (Brochmann, 2019), som handler om problemer med bruk av nettbrett i barn sin skolegang; [Nettbrett uten vett](#) (Machenbach, 2019), om skolens ukritiske bruk av digitale verktøy. Disse artiklene tar hovedsaklig for seg bruk av digitale verktøy i grunnskolen.

Bruk av digitale verktøy er mye diskutert, ofte rettet mot hvilke verktøy som brukes. I «Computational/Algorithmic Thinking» blir disse verktøyene ikke bare sett på som verktøy i undervisningen og læringsprosessen, men de er også tett forbundet med digital kyndighet som skal bygges opp for samfunnsfaglige og økonomiske grunner (Stephen & Kadjevich, 2020, s. 117). Et

av argumentene som brukes i LK20 er at de fem grunnleggende ferdighetene, nevnt i Kapittel 1 er viktige ... *for å kunne delta i utdanning, arbeid og samfunnsliv* («UDIR: Grunnleggende ferdigheter», 2020). Lewis bruker eksempelet med at voksne har en god forståelse for hvordan man skal beskytte fysiske eiendeler og privatliv; når vi forlater hjemmene våre låser vi dører og lukker vinduer, og at det er viktig å kunne gjøre det samme med vår digitale sikkerhet og vårt digitale privatliv (Lewis, 2017, s.23). I «Re-envisioning Computing Across Disciplines» begrunner Fee mfl. bruk av digitale verktøy i undervisningen med:

*The application we are looking for is not a specific career path. Instead, we wish to offer computing education that can illuminate to students from any discipline how developing a robust understanding of computing will make them more effective at exploring and solving problems wherever their studies or career lead them.*  
(Fee mfl., 2017, s. 2-3)

Målet er altså at bedre kjennskap til digitale hjelpemidler kan gjøre at man blir mer effektiv i utforskning og å løse problemer innen fagfeltet eller karriere man velger.

Ett uttrykk for “digitalisering” i skolen så vi i LK06. I MAT1-01, koden for faget fellesmatematikk (1 - 10 klasse og i tillegg Vg1 T, Vg1 P, Vg2 T og Vg2 P), som gjaldt fra 2006 til 2010 ble det ikke vist til bruk av digitale verktøy i kompetansemålene, mens MAT1-04, som er en senere versjon av samme fag hadde formuleringer som “bruk av digitale verktøy”. Utdrag fra disse to versjonene av fellesmatematikk i LK06 er vist i Tabell 2.1. Dette utdraget viser hva elevene skulle lære i geometri i 1T. Vi ser at det ble lagt til et punkt i MAT1-04, der bruk av digitale verktøy er eksplisitt nevnt. Kompetansemål som omhandler digitale verktøy ble lagt til flere av læreplanmålene i matematikkfaget underveis i LK06, der det passet seg. Disse målene la da til et ønske om at elevene fremdeles skulle kunne den samme grunnleggende matematikken, men da også bruke datamaskinen som et verktøy for å løse lignende oppgavetyper som tidligere. Behovet for digital kompetanse ble altså forstått, men fra egen erfaring ble datamaskinen ofte tatt i bruk som en mer avansert kalkulator. Med dette mener jeg at mye av potensialet, for eksempel for dataprogrammet GeoGebra, ikke ble brukt i undervisningen.

Tabell 2.1: Utklipp fra ett kompetansemål i LK06, fra 2006 og fra 2013

<b>Geometri 1T</b> (MAT1-01) Mål for opplæringa er at eleven skal kunne	<b>Geometri 1T</b> (MAT1-04) Mål for opplæringa er at eleven skal kunne
<ul style="list-style-type: none"> <li>• gjere greie for definisjonane av sinus, cosinus og tangens og bruke trigonometri til å berekne lengder, vinklar og areal i vilkårlige trekantar</li> <li>• bruke geometri i planet til å analysere og løyse samansette teoretiske og praktiske problem knytte til lengder, vinklar og areal</li> </ul> <p>«UDIR: MAT1-01», 2006</p>	<ul style="list-style-type: none"> <li>• gjere greie for definisjonane av sinus, cosinus og tangens og bruke trigonometri til å berekne lengder, vinklar og areal i vilkårlige trekantar</li> <li>• bruke geometri i planet til å analysere og løyse samansette teoretiske og praktiske problem med lengder, vinklar og areal</li> <li>• lage og bruke skisser og teikningar til å formulere problemstillingar, i oppgaveløysing og til å presentere og grunngje løysingane, med og utan bruk av digitale verktøy</li> </ul> <p>«UDIR: MAT1-04», 2013</p>

LK20 er en helt ny læreplan som fokuserer mer på digital kompetanse. I introduksjonen til dette kapitlet så vi kompetansemålene for 1T i denne læreplanen. Det første kompetansemålet nevner eksplisitt at eleven skal kunne bruke programmering og algoritmisk tenkning. At disse er nevnt i kompetansemålet som handler om å løse matematiske problem, tolker jeg som at flere av de resterende kompetansemålene implisitt krever programmering og algoritmisk tenkning. Elever som tar 1T matematikk kan blant annet gå videre med realfagmatematikk (R-matematikk). Om vi antar at de gjør dette vil de i den nye læreplanen for R1 ha kompetansemålet.

*[Mål for opplæringen er at eleven skal kunne] bestemme den deriverte i et punkt geometrisk, algebraisk og ved numeriske metoder, og gi eksempler på funksjoner som ikke er deriverbare i gitte punkter*

«UDIR: MAT03-02», 2020

Et eksempel på en numerisk metode er halveringsmetoden. En algoritme for denne numeriske metoden er gitt i Figur 2.5. Halveringsmetoden finner nullpunkt (roten) til en funksjon  $f(x)$ . Metoden halverer intervallet  $[a, b]$  i hver syklus inntil størrelsen på intervallet  $((b-a))$  er liten nok, og da estimeres nullpunktet til å være plassert midt mellom punktet  $a$  og  $b$ .

```

Gitt et intervall  $[a, b]$  slik at  $f(a)f(b) < 0$ 
mens  $(b - a)/2 > \text{TOL}$ 
   $c = (a + b)/2$ 
  hvis  $f(c) = 0$ , avslutt
  hvis  $f(a)f(c) < 0$ 
     $b = c$ 
  ellers
     $a = c$ 
Intervallet  $[a, b]$  inneholder nullpunktet
Det tilnærmede nullpunkt er  $(a + b)/2$ 

```

Figur 2.5: Halveringsmetoden, oversatt fra Sauer, 2018, s. 28

I kompetansemålene for R2 finner vi numeriske metoder igjen, denne gangen skal de programmeres:

*[Mål for opplæringen er at eleven skal kunne] utvikle algoritmer for å beregne integraler numerisk, og bruke programmering til å utføre algoritmene*  
 «UDIR: MAT03-02», 2020

Det er helt tydelig at det er en endring fra tidligere læreplan i hvordan digitale ferdigheter skal undervises. Hovedvekten av denne endringen ser ut til å ligge i at elevene skal lære programmering og algoritmisk tenkning, og dermed bruke digitale verktøy som et faglig verktøy, i motsetning til for eksempel en kalkulator som bare gir svar på aritmetiske spørsmål.

## 2.3 Grunnleggende Python

Programmering i Python betyr å bruke et språk som kan oversettes til datakode - “språket” til en datamaskin. I dette avsnittet blir de mest grunnleggende elementene introdusert.

Mens vi i dagligtale blander tall og tekst må vi, når vi programmerer, skille disse fra hverandre. I Python har vi fire *primitive typer*; heltall, flyttall, tekst-streng og boolsk verdi. Tabell 2.2 viser eksempler på dette.

Tabell 2.2: Python sine fire primitive typer

Type	Eksempel			
Heltall	0	-5	3	
Flyttall	0.0	5.3	10.52	
Tekst-streng	"5 + 5"		'Tekst'	
Boolsk verdi	True	False	0	1

Alt som er skrevet mellom `' '` eller `" "` i Python tolkes som en tekst-streng. Videre skiller Python mellom flyttall og heltall, der forskjellen er hvordan tallene lagres. Et flyttall lagres på en måte som kan føre til avrundingsfeil. Sauer (2018) kommer med eksempel på dette s. 15 i *Numerical Analysis*, som er vist i Figur 2.6. Heltall derimot er lagret på en måte som ikke kan føre til avrundingsfeil.

```

>>> a = 9.4
>>> a - 9
0.40000000000000036
>>> |

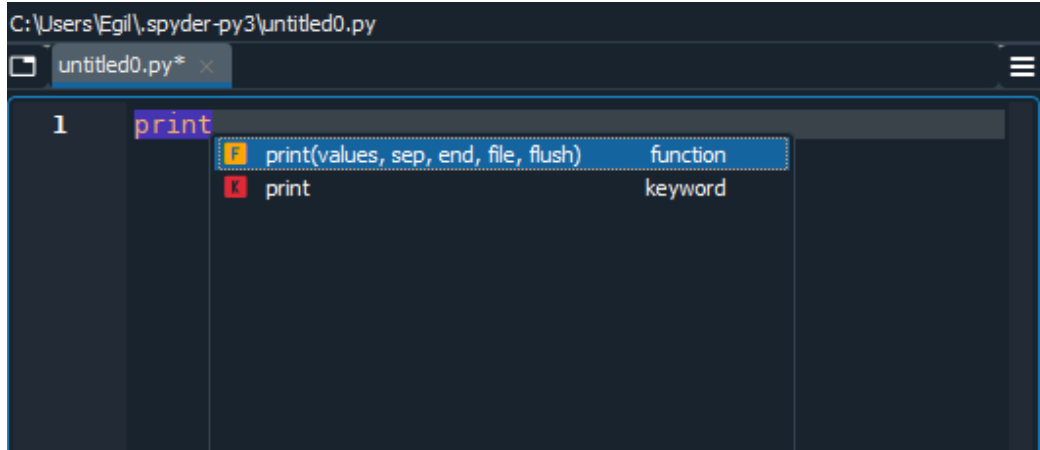
```

Figur 2.6: Eksempel på avrundingsfeil ved bruk av flyttall i programmering

I Python er en variabel noe man har tildelt en verdi som har formen til en av de fire primitive typene. For å gjøre denne tildelingen skriver man `<variabelnavn> = <type>`. Variabelnavnet kan være nesten hva som helst. I eksempelet gitt ser vi at variabelnavnet er `a`. Det er viktig å presisere at `"="` ikke betyr *er lik*, men betyr *tildeles*. En variabel tildeles altså en verdi eller tekststreng. Dette kan senere endres. I Figur 2.6 har vi skrevet at `a=9.4`, men hvis neste linje definerer at `a=5`, vil den gamle `a`-verdien `9.4` bli erstattet med `5`.

Python Shell er der kommandoene blir hentet inn og oversatt til datakode av Python. Eksempelet i Figur 2.6 viser regnestykke i Python Shell. `>>>` betyr bruker-input, mens en linje uten dette er output fra Python. Å skrive programmer er en "omvei" til kommandolinjen (Python Shell), en omvei som gjør det enklere å teste ut og lagre programmene våre. Når vi lager programmer skriver vi tekstfiler med `.py` som filsuffix. Filsuffix bestemmer

hvilket program som skal åpne filen, og `.py` betyr at det er Python som skal kjøre programmet. Et IDE er et tekstredigeringsverktøy som hjelper oss i prosessen med å skrive programmer, på lik linje med at Word, Google Docs, L<sup>A</sup>T<sub>E</sub>X, LibreOffice, osv. hjelper oss med å skrive tekster. Når man laster ned Python kommer dette med IDLE som IDE. I Windows er det også mulig å lage programmer i Notisblokk (engelsk: NotePad), og så lagre disse og endre filsuffixen fra `.txt` til `.py`. Elevene som deltok brukte IDLE eller Spyder som tekstredigeringsverktøy. Det er ikke noen forskjell på hva man kan få til mellom disse to verktøyene, men Spyder kommer med flere tillegg allerede installert, som mulighet for å bruke `pyplot` - et verktøy for å tegne grafer i Python. Den største forskjellen mellom de ulike IDE-ene er hvor mye hjelp man får av tekstredigeringsprogrammet. Figur 2.7 viser at Spyder kommer med forslag til hva man kan gjøre, i dette tilfellet når kommandoen `print` har blitt skrevet inn, noe IDLE ikke gjør. Det begge IDE-ene har til felles er at de fargelegger det vi skriver for å gjøre koden enklere å lese. I denne oppgaven brukes IDLE sin fargelegging når det vises eksempler på programmering.



Figur 2.7: Utklipp av Spyder IDE

Python kan også håndtere matematiske operatører, de mest grunnleggende operatorene er vist i Tabell 2.3 med navn, matematisk symbol og symbol brukt av Python.



Tabell 2.3: Matematiske og logiske operasjoner og deres Python-syntaks

	Navn	Symbol	Python syntaks
1	Addisjon	+	+
2	Subtraksjon	-	-
3	Multiplikasjon	*	*
4	Divisjon flyttall	:	/
5	Divisjon heltall		//
6	Rest		%
7	Eksponent		**
8	Mindre enn	<	<
9	Mindre enn eller lik	≤	<=
10	Er lik	=	==
11	Større enn eller lik	≥	>=
12	Større enn	>	>
14	Ulik	≠	!=

For å få Python til å gi oss et resultat brukes `print(<variabel/verdi>)`. For å be en bruker om å oppgi tekst som Python skal bruke kan `input(<tekst-streng>)` brukes, som igjen kan oversettes til tall ved å bruke `eval(<tekst-streng>)`, `int(<tekst-streng>)` eller `float(<tekst-streng>)`.

Dette er det nivået undervisningsopplegget i denne oppgaven ble lagt til, og vi vil derfor ikke komme inn på ting som for eksempel løkker (`for`, `while`) og spørsmål (`if`, `else`, `elif`). Jeg vil derfor avslutte dette avsnittet ved å vise et eksempel på bruk av variabler og operasjoner i Python. Eksempelet, vist i Figur 2.8, viser en måte å lage et program som finner midtpunktet mellom to punkt.

```
punkt1 = eval(input('Oppgi verdiene til det første punktet: '))
punkt2 = eval(input('Oppgi verdiene til det andre punktet: '))
midtpunkt = (punkt1 + punkt2)/2
print('Midtpunkt til disse verdiene er:')
print(midtpunkt)
```

Figur 2.8: Eksempel på program som finner midtpunktet mellom to punkt.

Dette eksempelet viser hva jeg forventer at en elev skal få til etter undervisningsopplegget, med tanke på kommandoer og rekkefølge i programmet.

## 2.4 Kompetanser

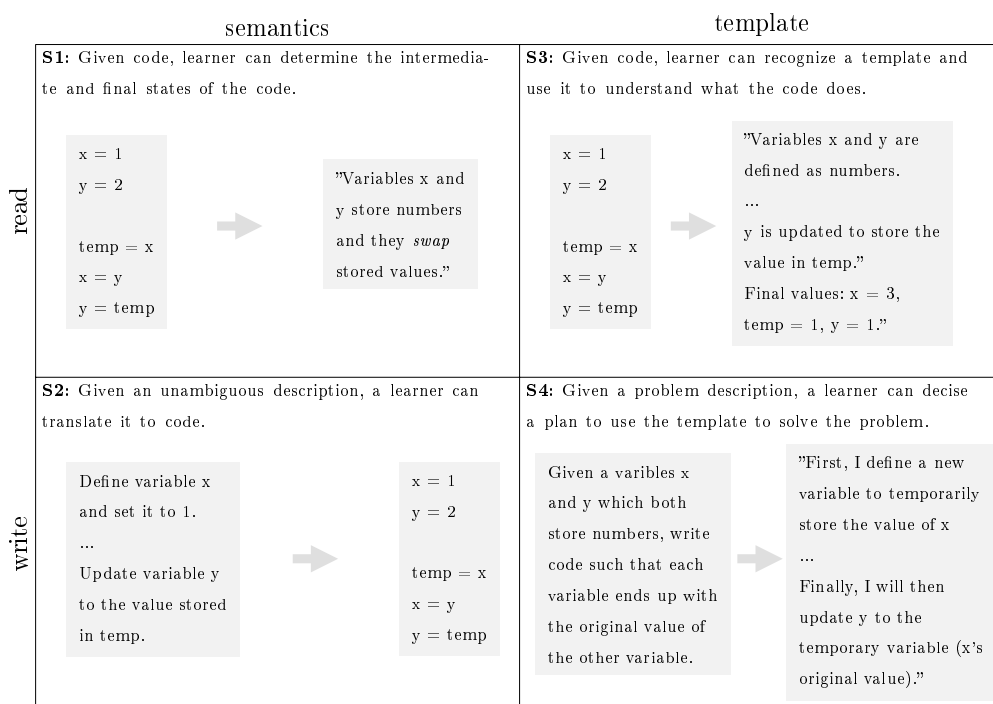
### 2.4.1 Programmeringskompetanser

Begrepet *Computational Thinking* (algoritmisk tenkning) ble skapt av Papert (Stephen & Kadijevich, 2020, s. 117). Algoritmisk tenkning er beskrevet i «Computational/Algorithmic Thinking» som en fundamental ferdighet, på lik linje med å kunne skrive, lese og regne. Et av verktøyene innen algoritmisk tenkning er programmering. På Utdanningsdirektoratet sine nettsider står det dette om algoritmisk tenkning:

*Algoritmisk tenkning innebærer å bryte ned komplekse problem til mindre, mer håndterlige delproblemer som lar seg løse. Det inkluderer å organisere og analysere informasjon på en logisk måte og å lage fremgangsmåter (algoritmer) for å komme fram til ønsket løsning.*

(«UDIR: Algoritmisk tenkning», 2019)

Videre i dette delkapittelet vil det bli presentert en måte programmering kan introduseres til elever. Dette er presentert i artikkelen «A theory of instruction for introductory programming skills» der grunnleggende programmering blir delt inn i to ferdigheter; lese (read) og skrive (write), og inn i to former for kunnskap; semantisk (semantics) og mal (template). Figur 2.9 gir en oversikt over disse to ferdighetene og de to formene for kunnskap. Semantikk handler om språk; et programmeringsspråk som Python krever en bestemt måte å skrive på for at et program skal fungere. Semantikk innen programmering handler om hva en kode betyr (å lese kode) eller hvordan den skal skrives (å skrive kode). Begrepet *template* blir her oversatt til mal, og handler om å manipulere tidligere kodesnutter (maler) slik at de kan brukes til å løse nye problem.



Figur 2.9: Xie mfl. (2019) sine grunnleggende ferdigheter. Figur laget gjennom TikZ, basert på figur fra Xie mfl., 2019, s. 11

Disse ferdighetene er delt inn i S1 - S4. Xie mfl. mener at i starten så bør lesing komme før skriving og at det skriftlige (semantikken) bør være fokus før mal (Xie mfl., 2019, s. 10). S1 handler om å kunne lese koden og å kunne beskrive hva den gjør, mens det i S2 krever at man kan gå fra en utvetydig beskrivelse til å skrive et program. S3 handler om å gjenkjenne en mal. I Figur 2.10 vises eksempelet fra Figur 2.8 i Delkapittel 2.4.1 som utgangspunkt for en mal.

```
<variabelnavn1> = eval(input('<tekst1>'))
<variabelnavn2> = eval(input('<tekst2>'))
<variabelnavn3> = [regnestykke]
print('<tekst3>')
print(<variabelnavn>)
```

Figur 2.10: Utgangspunkt for en mal.

S4-ferdigheten viser til at en elev skal kunne gå fra en beskrivelse av et

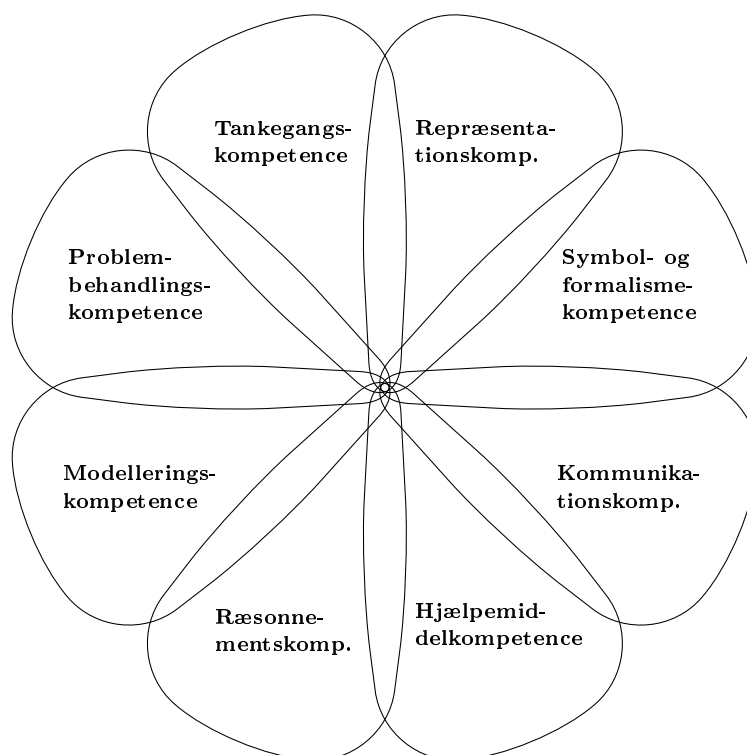
problem til å gi en beskrivelse av en løsning. Problemet som ble løst i Figur 2.8 kan bli gitt på følgende måte: “gitt to punkt, **a** og **b**, som begge er oppgitt av en bruker, skriv en kode som finner og skriver ut midtpunktet mellom disse”. Når man jobber i S4 vil man gi en beskrivelse av rekkefølgen og av hva man gjør; *først ber vi brukeren om to tall, som lagres som a og b. Videre definerer vi en variabel, midtpunkt, ved hjelp av a og b. Til slutt ber vi programmet skrive ut midtpunktet.* Å faktisk skrive programmet hører til i S2.

Vi ser at Xie mfl. gir en metode for å introdusere programmering som starter med kjennskap til programmeringsspråket (S1 og S2). Videre skal eleven jobbe med den algoritmiske tenkningen (S3 og S4). Etter hvert som elever blir bedre til å programmere og tenke algoritmisk kommer de til å måtte lære å lese og å skrive nye ting, med påfølgende læring i S3 og S4.

## 2.4.2 Matematiske kompetanser

Algoritmisk tenkning og programmering antas å kunne være faktorer som kan hjelpe elever til å bygge opp matematiske kompetanser og knytte ulike matematiske kompetanser sammen. Niss mfl. deler matematiske kompetanser inn i to kategorier; *Å kunne spørre og svare i, med og om matematikk* og *Å omgås språk og redskaper i matematikk* (Niss mfl., 2002, s. 44). Under disse to kategoriene er det igjen åtte kompetanser, ofte ser vi disse som en åttebladet blomst, som vist i Figur 2.11.

Disse kompetansene blir delt inn under de to kategoriene. Den første kategorien *Å kunne spørre og svare i, med og om matematikk* inneholder tankegangskompetanse, problemløsningskompetanse, modelleringskompetanse og resonneringskompetanse. Den andre kategorien *Å omgås språk og redskaper i matematikk* inneholder representasjonskompetanse, symbol- og formalismekompetanse, kommunikasjonskompetanse og hjelpemiddelskompetanse. Selv om dette er tilfellet betyr ikke dette at kompetansene kun støtter sin egen kategori, eller at gode egenskaper innen en av kategoriene ikke overføres til gode egenskaper i den andre kategorien. Eksempelvis er det ikke slik at problemløsningskompetanse kun hjelper med å utvikle elevenes evne til å kunne spørre og svare i, med og om matematikk, elevene bruker for eksempel symboler og kommunikasjon for å løse et problem i matematikk. Disse åtte kompetansene vil videre bli utdypet. Som Niss mfl. skriver, så er ikke målet med en slik beskrivelse å *selvstendigjøre* disse kompetansene. Det å kategorisere matematiske kompetanser i ulike delkompetanser er nyttig for både lærer og elev, nettopp fordi vi da kan utdype hva de går ut på (Niss mfl., 2002, s.47).



Figur 2.11: Niss sine 8 matematiske kompetanser, figur laget gjennom TikZ, basert på figur fra Niss mfl., 2002 s. 45.

Tankegangskompetanse handler, som navnet skulle tilsi; om å kunne tenke matematisk. I matematikk finnes det bestemte tradisjoner for hvordan spørsmål ser ut, hvilke begreper som brukes og hvordan man formulerer svar. Tankegangskompetanse handler om å være klar over hvilke spørsmål som er karakteristisk for matematikk, hvordan stille slike spørsmål selv og å være klar over hvordan svar på disse spørsmålene kan se ut (Niss mfl., 2002, s. 47).

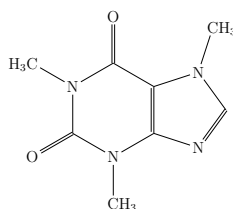
Problembehandlingskompetanse handler om å formulere og å kunne løse matematiske problem. Denne kompetansen går ut på å kunne oppdage, formulere, avgrense og presisere forskjellige matematiske problem (Niss mfl., 2002, s. 49). Et matematisk problem er et problem som krever matematisk undersøkelse for å finne et svar. Ikke alle matematiske spørsmål er et matematisk problem, for noen kan et matematisk spørsmål kreve kjente rutiner og derfor ikke utgjøre et problem (Niss mfl., 2002, s. 49). Noe som kan være med å komplisere det hele, er at noe som en elev oppfatter som et matematisk problem trenger ikke å bli oppfattet som et problem av en annen elev.

Modelleringskompetanse handler om å kunne bruke matematiske model-

ler. Fra UiB sin ordbok, <https://ordbok.uib.no/>, blir ordet *modell* definert som følgende:

modell m1 (fra italiensk, av latin *modulus* 'mål, målestokk', av *modus* 'måte')

En modell i moderne forstand er en representasjon som viser noe. Ett modellfly prøver å være tro til hvordan noe faktisk ser ut, bare skalert ned i størrelse, mens en modell for det kjemiske molekylet koffein kan ta formen vist i Figur 2.12, som ikke representerer en fysisk modell av molekylet, men heller en modell som illustrerer funksjonen eller prinsippet bak molekylet.



Figur 2.12: Modell av et koffeinmolekyl, laget i L<sup>A</sup>T<sub>E</sub>X

Figurer, tabeller, grafer og formler er noen eksempler på måter en matematisk modell kan representeres. Dersom jeg kjører fra Bergen til Voss og holder 60km/t hele veien kan en modell for dette beskrives ved  $f(x) = 60\text{km/t} \cdot x$  der  $x$  er tiden som ble brukt og  $f(x)$  er avstand kjørt. Niss mfl. beskriver modelleringskompetanse som evnen til å avgjøre om og når en modell er gyldig, evnen til å avgjøre hva modellen beskriver og evnen til å lage sine egne matematiske modeller (Niss mfl., 2002, s. 52).

Resonnementskompetanse handler ikke bare om å produsere matematiske resonnement, men også evnen til å *følge* og å *bedømme* et matematisk resonnement (Niss mfl., 2002, s. 54).

Representasjonskompetanse i matematikk handler om hvordan elevene forstår og bruker ulike matematiske representasjoner. Dette kan være representasjoner av matematiske objekter, fenomener, problemer eller situasjoner (Niss mfl., 2002, s. 56). Eksempelet med en kjøretur fra Bergen til Voss bruker en formel som representerer bilen sin bevegelse, en representasjon som blant annet krever symbol-forståelse. Representasjonskompetanse handler også om å forstå forbindelsen som kan være mellom ulike representasjoner og å velge den best egnede representasjonen (Niss mfl., 2002, s. 57).

Hjelpemiddelskompetanse handler om å ha kjennskap til ulike redskap man har i matematikk og egenskapene disse har (Niss mfl., 2002, s. 62). Python, GeoGebra og Excel er eksempler på digitale redskap i matematikk.

Kommunikasjonskompetanse handler om å kunne sette seg inn i, og tolke andre sine matematiske utsagn og tekster, og å kunne uttrykke seg selv matematisk. Matematikk kan uttrykkes muntlig, visuelt eller skriftlig (Niss mfl., 2002, s. 60).

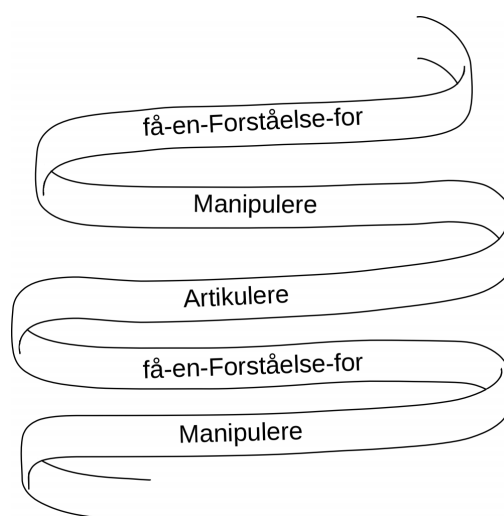
Symbol- og formalismekompetanse handler om å kunne avkode symbol- og formelspråk, å kunne oversette matematisk språk til naturlig språk og motsatt, og å kunne bruke og behandle utsagn og uttrykk med symbol i seg (Niss mfl., 2002, s. 58).

I denne oppgaven legges vekt på *symbol- og formalisme-kompetanse* og *problembehandlingskompetanse*. Opplegget som ble utviklet, og tilhørende intervju brukte algebra som utgangspunkt for å bruke programmering som problemløsningsverktøy. *Symbol- og formalisme-kompetanse* kan for eksempel handle om hvordan en elev tar et algebraisk problem og “oversetter” dette til en form Python kan bruke, mens *problembehandlingskompetanse* ser på hvordan eleven løser dette problemet.

### 2.4.3 Å tenke algebraisk

Begrepet *algebra* kommer fra Islamsk matematikk, ca. 800 e.kr. En av de tidligste muslimske tekstene om algebra ble skrevet av Muhammad ibn Mūsā al-Khwārizmī (Katz, 2014, s. 272). Denne boken handlet om al-jabr, opphavet til ordet algebra, og al-muqābala. Begge disse begrepene tar for seg prosesser innen det vi i dag kjenner som algebra. Begrepet al-jabr refererte til prosessen med å flytte en negativ verdi på en side av en ligning til den andre siden, der den ble positiv. Begrepet al-muqābala refererte til prosessen med å finne to like verdier på hver sin side av en ligning, for så å trekke vekk disse to verdiene. Eksempelet Katz gir på dette er  $3x + 2 = 4 - 2x$ . Ved al-jabr flytter vi  $-2x$  over på den andre siden og får  $3x + 2x + 2 = 4$  som blir  $5x + 2 = 4$ . Ved al-muqābala ser vi at begge sider har verdien 2 i seg, og trekker dette vekk på begge sider. Da får vi  $5x = 2$ . Det kan også være verdt å nevne, med tanke på tema i denne oppgaven, at Muhammad ibn Mūsā al-Khwārizmī sitt navn gav opphav til begrepet algoritme (al-Khwārizmī  $\Rightarrow$  algoritme) (Katz, 2014, s. 272).

En del av algebraisk tenkning handler om generalisering. Dette er prosessen der symbol blir brukt for å representere alle mulige tall man kan tenke på. I boken *Å lære algebraisk tenkning* som er skrevet av Mason mfl. og oversatt til norsk av veilederen min, Johan Lie, blir en tilnærming til prosessen med å generalisere beskrevet ved hjelp av en spiral. MFA-spiralen (Manipulere, få-en-Forståelse-for, Artikulere), som er vist i Figur 2.13, beskriver prosessen med at man begynner med å manipulere kjente objekt eller spesielle eksempler.



Figur 2.13: MFA-spiralen hentet fra Mason mfl., 2014, s. 39.

Dette er noe som hjelper eleven med å gjøre seg opp en mening og få kjennskap til det som skal generaliseres. Manipuleringen fører til at eleven får en forståelse for det som skal generaliseres. Til slutt skal eleven prøve å artikulere-beskrive med egne ord en generell sammenheng. Dette er beskrevet ved en spiral fordi første artikulering ikke trenger å være helt korrekt og heller ikke perfekt (Mason mfl., 2014, s. 39).

I «Algebra Teaching and Learning» i *Encyclopedia of Mathematics Education* blir forskning på elevers overgang fra aritmetikk til algebra i undervisningen oppsummert i flere punkt. Under vises tre av disse punktene.

1. students' visual imagery of inequalities and of equations that involve linear functions on both sides of the equal sign is greatly assisted by graphical representations and by digital software featuring such representations
2. students' difficulties with recognizing structure in algebraic expressions and equations are a reflection of the difficulties they have with recognizing structure in number and arithmetical operations;



3. students' representations of functional problem situations evolve from first using numbers as the only means of modeling, then to intensively working with graphs and tables, and lastly to using more symbolic representations

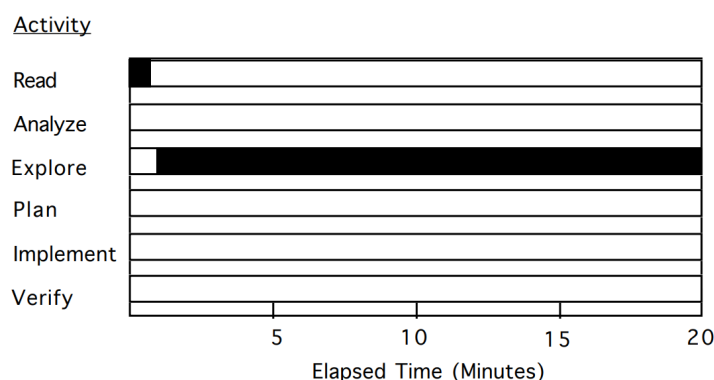
(Kieran, 2020, s. 39)

Disse tre punktene indikerer hvordan elever tenker når de jobber algebraisk. I Punkt 1 trekkes bruk av digital teknologi frem som et konseptuelt verktøy som fremmer læring i matematikk. Kieran trekker frem at digital teknologi, etter hvert som eleven blir eldre, også spiller en rolle som et verktøy for og i matematisk arbeid (Kieran, 2020, s. 39), og ikke bare for læringen sin del.

#### 2.4.4 Problemløsning i matematikk

For å kunne hjelpe en elev til å utvikle og forbedre sine matematiske ferdigheter er det viktig å vite hvordan hen løser problemer. Niss mfl. har problemløsningskompetanse som en av sine åtte matematiske kompetanser. Fordi den største delen av datainnsamlingen i dette prosjektet er transkripsjon av elever som løser oppgaver ser vi nærmere på problemløsning i matematikk her. Schoenfeld sin tekst «Learning to Think Mathematically: Problem Solving, Metacognition, and Sense Making in Mathematics» har tre figurer som er aktuelle for denne oppgaven, Figur 2.14 viser hvordan et elev-par løser ikke-standardisert (matematisk) problem. Figur 2.15 viser hvordan en matematiker jobber med vanskelige problem. Figur 2.16 viser hvordan elev-par jobber med ikke-standardiserte problem etter kurs i problemløsning. Disse figurene representerer over 100 videopptak fra situasjonene beskrevet der deltakerne hadde 20 minutt på å løse en oppgave.

Elevene i Figur 2.14 leste problemet raskt og valgte en fremgangsmåte som de fulgte. Elevene brukte denne fremgangsmåten selv når det var tydelige tegn på at de ikke hadde fremgang i å løse problemet. Etter tiden var over (20 minutter), ble elevene spurt om hvordan deres fremgangsmåten kunne hjelpe med løse det originale problemet, noe elevene ikke kunne svare på (Schoenfeld, 1992, s. 355-356).



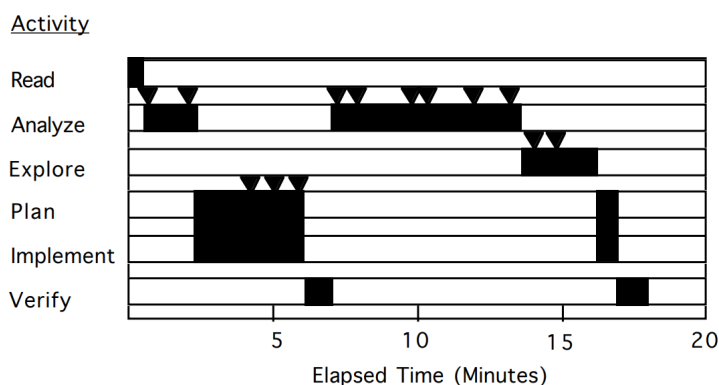
Figur 2.14: Tidslinje-graf som viser hvordan en (typisk) elev bruker tiden når hen prøver å løse ikke-standardisert problem, figur hentet fra Schoenfeld, 1992, s. 356

Schoenfeld understreker at Figur 2.14 ikke representerer vanlig tidsbruk til en elev når hen løser (vanlige) oppgaver i matematikk (Schoenfeld, 1992, s. 356). Dette er fordi elever som oftest løser rutineoppgaver; oppgaver der de har en kjent fremgangsmåte. Schoenfeld mener at problemløsningsoppgaver vanligvis blir separert fra “matematikken” elevene skal lære, enten ved at disse oppgavetyperne blir markert som vanskeligere eller at de er plassert i egne avsnitt dedikert og markert som problemløsningsoppgaver (Schoenfeld, 1992, s. 354). Denne separeringen gir et implisitt signal om at denne oppgavetyper er *i tillegg til*, eller som Schoenfeld skriver

*The implicit message contained in this format is “You may take a breather from the real buisness of doing mathematics, and enjoy yourself for a while.”*

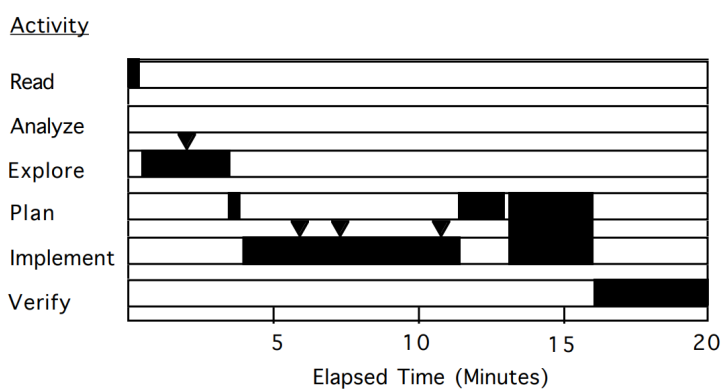
(Schoenfeld, 1992, s. 354)

Til sammenligning viser Figur 2.15 hvordan en matematiker går frem for å løse et vanskelig matematisk problem. Det er ikke et mål i seg selv å få elever til å løse problem likt som en matematiker. Fra Figur 2.15 ser vi at matematikerne brukte neste halvparten av tiden på å forstå (analysere) problemet. Matematikeren låste seg heller ikke til en fremgangsmåte, men prøvde ut flere. ▼ markerer steder der matematikeren kom med kommentarer på fremgangen, eksempelvis “OK. All I need to be able to do is [a particular technique] and I’m done.” (Schoenfeld, 1992, s. 356).



Figur 2.15: Tidslinje-graf som viser hvordan en matematiker jobber med et vanskelig problem, figur hentet fra Schoenfeld, 1992, s. 356

Den siste figuren, Figur 2.16, viser effekten av det Schoenfeld kaller kurs i problemløsning (problem solving classes).



Figur 2.16: Tidslinje-graf som viser hvordan en (typisk) elev bruker tiden når hen prøver å løse ikke-standardisert problem etter kurs i problemløsning, figur hentet fra Schoenfeld, 1992, s. 356

I disse timene jobbet elevene i grupper på tre eller fire og arbeidet med matematiske problem. Læreren (the instructor) gikk mellom gruppene, og stilte spørsmål (Schoenfeld, 1992, s. 356). Det var i hovedsak tre spørsmål som ble stilt.

What (exactly) are you doing?

(Can you describe it precisely?)

Why are you doing it?

(How does it fit into the solution?)

How does it help you?

(What will you do with the outcome when you obtain it?)

Schoenfeld, 1992, s. 356

I artikkelen «Relational Understanding and Instrumental Understanding» peker Skemp på at begrepet *å forstå* har to betydninger. Når en elev sier at hen har forstått noe kan det bety at hen mestrer metoden som får rett svar, som er *instrumentell forståelse*. Den andre betydningen, *relasjonell forståelse*, handler om å forstå hva man skal gjøre (for å løse et problem) og hvorfor vi gjør dette (Skemp, 1976). Det var Mellin-Olsen som først påpekte for Skemp at begrepet *å forstå* (understanding) hadde to ulike betydninger (Skemp, 1976, s. 2). Mellin-Olsen viser til at det er to drivkrefter for instrumentell læring. Det første er læreren sin undervisning, og fokuset læreren har på å forstå formler versus å bruke formler. Selv om læreren ikke underviser med mål om instrumentell forståelse kan det hende eleven sine mål og syn på læring fører til dette uansett (Mellin-Olsen, 1984, s. 41 - 45). Schoenfeld sine spørsmål legger vekt på å utvikle en elev sin relasjonelle forståelse i matematikk. Schoenfeld påpeker at poenget (med det figurene illustrerer) ikke er at elevene løser et problem, ikke-standard problem krever forkunnskap og hell. Poenget Schoenfeld vil frem til er at elevene gir seg selv mulighet til å løse et vanskeligere problem (Schoenfeld, 1992, s. 357). Etter kurs i problemløsning så Schoenfeld en medfølgende (concomitant) økning i elever som klarte å løse ikke-standardisert problem.

## 2.5 Kognitiv utvikling

For å kunne utvikle og strukturere et undervisningsopplegg er det nyttig å vite noe om hvem som skal undervises. Hvordan tenker eleven og hvordan kan informasjon og kunnskap overføres til elevene? Kognisjon er her et sentralt begrep:

*Cognition refers to the inner processes and products of the mind that lead to "knowing". It includes all mental activity-attending,*

*remembering, symbolizing, categorizing, planning, reasoning, problemsolving, creating, and fantasizing.* (Berk, 2013, s.225)

Kognisjon handler altså om all mental aktivitet, alt fra tilstedeværelse til resonnering. I dette avsnittet går vi gjennom Piaget sine teorier om kognitiv utvikling. Jean Piaget (1896-1980) utviklet teori om barns kognitive utvikling, i Tabell 2.4 vises stadiene i kognitiv utvikling som Piaget mente mennesker går gjennom. I det konkret-operasjonelle (*Concrete Operational*) stadiet nærmer den kognitive prosessen seg det vi finner hos voksne. Tanker blir mer logiske, fleksible og organiserte (Berk, 2013, s. 249). Barn i dette stadiet er fortsatt dårlig med abstrakte ideer, altså ideer som ikke er åpenbare i den virkelige verden (Berk, 2013, s. 252). De elevene denne oppgaven fokuserer på tilhører det siste stadiet; det formelt-operasjonelle stadiet (*Formal Operational*). Tabell 2.4 viser oversikt og beskrivelse over Piaget sine fire stadier av kognitiv utvikling.

Tabell 2.4: De fire stadiene til Piaget (hentet fra Mooney (2013), s. 64)

Piaget's Stages of Cognitive Development		
Age	Stage	Behaviors
Birth to age 2	Sensorimotor	Learn through the senses; learn through the reflexes; manipulate materials.
2-7 years	Preoperational	Form ideas based on their perceptions; can only focus on one variable at a time; overgeneralize based on limited experience.
7-11 or 12 years	Concrete Operational	Form ideas based on reasoning; limit thinking to objects and familiar events.
11 or 12 years and older	Formal Operational	Think conceptually; think hypothetically

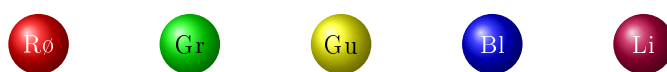
I det formelt-operasjonelle stadiet trenger barna ikke lengre konkrete ting eller hendelser som objekt for de kognitive prosessene (Berk, 2013, s. 253). Piagets pendelproblem er et eksempel på dette.

**Piaget sitt pendelproblem** går ut på å gi barn hyssinger med ulik lengde, objekter med ulik vekt, slik at de kan lage en pendel av disse; og et stativ å feste pendelen i. Spørsmålet barnet skal svare på er *hva som påvirker farten til pendelen*.

Barn som har nådd det formelt-operasjonelle stadiet klarer å separere inn i variabler som kanskje påvirker farten; lengden til hyssingen, objektets vekt, høyden objektet slippes fra og om objektet blir skubbet, og å teste ut effekten

av de forskjellige variablene. Dette er i motsetning til barn i det konkrete operasjonelle stadiet, som kan teste ut ulike påvirkninger på pendelen, men som ikke er i stand til å skille mellom og teste enkeltvariabler. Eksempel på dette kan være å teste ut påvirkningen lengden på hyssingen har, men at de samtidig også endrer vekten på objektet (Berk, 2013, s. 254). I en metaanalyse gjort av John Hattie rangeres undervisningsprogram som tar utgangspunkt i Piagets teorier (Piagetian programs) som nummer to med stor effektstørrelse,  $d=1.28$  (Hattie, 2009, Appendix B). Dette var en metaanalyse av mer en 800(!) andre analyser relatert til oppnåelse i utdanning, som siden har blitt oppdatert i senere utgivelser til å inkludere enda flere. På nettsiden Visible-Learning («Visible learning», 2020) kan man finne en oppdatert liste. Hva betyr  $d=1.28$ ? Dette kan være måte å vise effektstørrelse på, der en økning på  $d=1.0$  i en elev kan sies å tilsvare ca. to til tre år, eller å øke læringseffekten med ca. 50% (Hattie, 2009, kap. 2).  $d=0.4$  blir ansett som nøytral med tanke på kognitiv utvikling. At Hattie bruker flere beskrivelser på dette er fordi han sammenligner mange ulike element som kan påvirke en elev sin læring, og Hattie har plassert *Piagetian programs* nesten helt øverst i sin metaanalyse. Det er en sammenheng mellom stadiet en elev er på i Piaget sin teori og oppnåelse på skole, spesielt i matematikk (Hattie, 2009, s. 43). *Piagetian programs* refererer til program som baserer seg på at man skal undervise på det nivået elevene er på i Piaget sin teori.

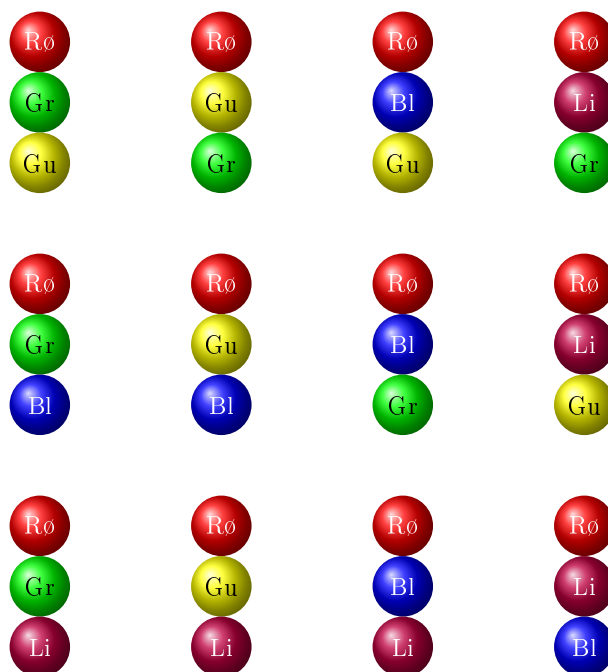
Papert sier selv at han brukte mange av Piaget sine ideer da han utviklet Turtle geometri (Papert, 1980, s. 34 - 36). Et av poengene med Turtle geometri er at barn kan identifisere seg med skilpadden, og trekke inn erfaringer fra hvordan de selv beveger seg mens de lærer formell geometri (Papert, 1980, s. 69). Et eksempel Papert bruker for å forsvare bruk av programmering i matematikkundervisning er fra når barn blir undervist kombinatorikk. Når et barn i det konkret-operasjonelle stadiet blir bedt om å finne alle kombinasjonsmuligheter til ett sett med ulik-fargede kuler er de ikke i stand til dette. Papert mener at dette kan ha med "kulturen" barnet vokser opp i; dersom barnet har lært å bruke løkker i programmering krever ikke dette problemet noen form for abstrakt tenkning. I stedet kan barnet sette opp en løkke som starter med en farge, og som går gjennom fargene som kan plasseres som nummer to, og gjenta denne prosessen til alle kombinasjonene er funnet (Papert, 1980, s. 36). Figur 2.17, Figur 2.18 og Figur 2.19 viser fem ulik-fargede kuler, den første og den andre syklusen i en løkke som finner alle kombinasjonsmulighetene.



Figur 2.17: Fem kuler med ulike farger, figur laget gjennom TikZ



Figur 2.18: Første syklus i kombinasjons-oppgaven, figur laget gjennom TikZ



Figur 2.19: Andre syklus i kombinasjons-oppgaven, figur laget gjennom TikZ

I Figur 2.19 begynner kombinasjonene i første kolonne med første kombinasjon i Figur 2.18, Rød, Grønn osv. Papert viser også til at problemet med å fjerne gjentakende kombinasjoner, for eksempel Rød-Grønn-Gul og Rød-Gul-Grønn også blir trivielt dersom man kan programmere og algoritmisk tenkning (Papert, 1980, s. 36).

## 2.6 Forskningsintervju

I masterprosjektet mitt har jeg gjennomført forskningsintervju for å samle inn kvalitative data.

Kvale og Brinkmann (2019) skisserer en idealisert vei for en intervjuundersøkelse som består av syv faser; tematisering av et intervjuprosjekt, design, selve intervjuet, transkribering, analyse, verifikasjon og rapportering. Disse syv fasene blir forklart på s. 137 i boken, og er gjengitt under.

1. **Tematisering** Formuler formålet med undersøkelsen, og beskriv hvordan du oppfatter emnet som skal undersøkes, før intervjuarbeidet begynner. Du bør klarlegge undersøkelsens hvorfor- og hva-spørsmål før du stiller spørsmålet hvordan - det vil si, før du velger metoder.
2. **Planlegging** Planlegg studien, og ta hensyn til alle syv stadier, før du tar fatt på intervjuarbeidet. Planlegg med henblikk på å innhente den kunnskapen du ønsker, og med tanke på studiens *moralske* implikasjoner.
3. **Intervjuing** Utfør intervjuene på grunnlag av en intervju-guide, og med en reflektert tilnærming til kunnskapen som søkes og intervjuets kontekst, samt ta hensyn til intervju-situasjonens mellommenneskelige relasjoner.
4. **Transkribering** Klargjør intervjumateriale for analyse, noe som vanligvis medfører transkribering fra tale til skriftlig tekst.
5. **Analysering** På grunnlag av undersøkelsens formål og emneområde, og i samsvar med intervjumaterialets natur, bestemmer du hvilke analysemetode som er best egnet for intervjuene.
6. **Verifisering** Undersøk intervjufunnenes generaliserbarhet, pålitelighet og validitet. Reliabilitet henviser til hvor pålitelige resultatene er, og validitet vil si hvorvidt en intervjustudie undersøker det den er ment å skulle undersøke.
7. **Rapportering** Undersøkelsefunnene og metodebruken formidles i en form som overholder vitenskapelige kriterier, tar hensyn til undersøkelsens etiske sider, og resulterer i et lesbart produkt.

*Hentet fra Kvale og Brinkmann (2019) s. 137*



# Kapittel 3

## Metode

I dette prosjektet ble et undervisningsopplegg laget og gjennomført for tre Vg1 klasser på en studiespesialiserende i Vestland fylke, for å se hvordan programmering kan innlemmes i matematikkundervisning. Det matematiske temaet som ble brukt som bakgrunn var algebra, fordi det var dette temaet elevene hadde jobbet med tidligere i 1T. Før opplegget ble utført ble det gjennomført en forberedelsestime i klassene, opplegget ble presentert og det ble sjekket at alle elevene hadde nødvendig programvare på PC-ene sine. Under og rett etter gjennomføringen av opplegget ble observasjoner notert ned. Etter opplegget var ferdig ble lærerne, samt noen elever intervjuet om opplegget. I tillegg ble elevene bedt om å løse to oppgaver, og dette ble til slutt transkribert. I de følgende delkapitlene er hvert punkt beskrevet nøyaktig.

Lydopptak av en person blir ansett som persondata, noe som gjorde at vi måtte søke om tillatelse fra NSD - *Norsk senter for forskningsdata* (<https://www.nsd.no/>). Søknaden ble godkjent, og i Tillegg B er svaret fra NSD gjengitt. Så ble brev sendt rundt til skoler for å se om noen var interesserte i delta i undervisningsopplegget. Brevet er vist i Tillegg C. For å kunne gjennomføre intervjuet måtte intervjuobjektene få informasjon, informasjonsskrivet er gjengitt i Tillegg D. I dette skrevet er det også en samtykkeerklæring intervjuobjektene måtte signere i forkant av intervjuet.

### 3.1 Undervisning

Ettersom opplegget kun inngikk i en undervisningsøkt, var det viktig å vite hva lærerne hadde gått gjennom i tidligere undervisning. I tillegg ble det tydelig fra starten at lærerne ikke hadde mye erfaring med programmering selv. Figur 3.1 viser en av e-postene jeg fikk fra  $L_2$ .  $L_1$  og  $L_2$  er anonymisert kallenavn for lærerne.

Heisann!

Dette blir bra ☹ Eg har svært liten kunnskap om programmering dessverre, så dette vert lærerikt for meg også.

Eg har undervisning i 1T:

Oddetalsveker:  
[REDACTED]  
[REDACTED]

Partallsveker:  
[REDACTED]  
[REDACTED]

Mvh  
[REDACTED]

Figur 3.1: Svar fra L<sub>2</sub>

Figur 3.2 viser en e-post fra L<sub>1</sub> som forklarte hva elevene hadde gått gjennom i matematikkundervisningen. Som jeg viste til i starten av Kapittel 2, mener jeg at programmering og algoritmisk tenkning skal innlemmes i de andre kompetansemålene der det passer. Innen matematikk er det helt egne retninger dedikert til utvikling av CAS (Choen, 2002, s. 2). CAS, som står for *Computer Algebra Systems*, handler om at datamaskinen bruker algoritmer og formler for å løse algebraproblem. Ett eksempel på dette er abc-formelen som brukes til å faktorisere  $ax^2 + bx + c$  til  $a(x - x_1)(x - x_2)$ . Denne formelen gir at  $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ . En av oppgavene i elev-intervjuet gikk ut på at elevene skulle lage et CAS som fant og gav ut  $x_1$  og  $x_2$ .

Hei!

Jeg har to grupper i 1T som du kan komme inn i.  
Begge gruppene har til nå jobbet med algebraiske uttrykk, kvadratsetninger, faktorisering og forkorting av rasjonale uttrykk.  
Er det noe du kan bruke?

Vennlig hilsen  
[REDACTED]

Figur 3.2: Svar fra L<sub>1</sub>

Tabell 3.1 viser årsplanen til klassen som tilhørte L<sub>2</sub>. Dette ble brukt som utgangspunkt for undervisningsopplegget. Jeg møtte elevene i alle tre klassene i uke 40 for å forsikre meg om at eleven hadde nødvendig dataprogram installert. Opplegget ble gjennomført i uke 42 som en del av klassenes vanlige undervisningstid, som var mellom 2 timer og 2 timer og 15 minutt (medregnet friminutt på ca. 10-15 minutt).

Tabell 3.1: Årsplan for en av klassene, uke 35 - 41.

Ukenummer	
35	<ul style="list-style-type: none"> <li>• Regnerekkefølge</li> <li>• Brøkrekning</li> <li>• Bokstavuttrykk</li> </ul>
36	<ul style="list-style-type: none"> <li>• Mengdelære</li> <li>• Kvadratsetningene</li> <li>• Faktorisering</li> <li>• Forenkle rasjonale uttrykk</li> </ul>
37	<ul style="list-style-type: none"> <li>• Forenkle rasjonale uttrykk</li> <li>• Prosentregning</li> <li>• Førstegradsligninger</li> </ul>
38	<ul style="list-style-type: none"> <li>• Formler</li> <li>• Rette linjer</li> </ul>
39	<ul style="list-style-type: none"> <li>• Ettpunktsformelen</li> <li>• Vurdering</li> </ul>
40	<ul style="list-style-type: none"> <li>• Egenvurdering + Fagsamtaler</li> <li>• Lineære ligningsett</li> <li>• Ettpunktsformelen</li> <li>• Andregradsligninger med to ledd</li> <li>• Arbeide med CAS/GeoGebra</li> <li>• Grafisk løsning av ligninger</li> <li>• Grafisk løsning av andregradsligninger</li> </ul>
41	Høstferie

I Tillegg E er lysbildene fra økten gjengitt. De første minuttene av timen ble brukt til å hjelpe elevene med å åpne programmet. De elevene som ikke hadde Spyder installert ble bedt om å installere Python direkte fra [python.org](https://python.org), dette fordi det er raskere å laste ned vanlig Python; i skrivende stund har Spyder ca. 169MB som må lastes ned, mens vanlig Python ca. 27MB. Valgene med nedlasting ble tatt før UDIR hadde lagt ut den første prøveeksamen (eksempeloppgaver for lærere om hvordan eksamen kom til å se ut) i dette faget og vi så at de krever bruk av tilleggspakker til Python. Å laste ned slike pakker er ikke avansert, men kan være krevende når en lærer skal hjelpe femten elever med ulike PC-er å laste de ned. Nedlasting av tilleggspakker er forskjellig på Windows dersom man har lastet ned Python fra Windows Store enn fra [python.org](https://python.org), og det er forskjellig mellom Windows, Mac og Linux. Spyder kommer med alle tilleggspakkene installert fra før, som fjerner dette problemet. Tilleggspakken som ble brukt på prøveeksamen var *numpy*.

En kort beskrivelse av de ulike vinduene, IDE og kommandolinje (Python Shell), ble gitt før elevene selv begynte å teste ut koding. Det første elevene skulle prøve ut var de ulike operasjonene. Dette skulle de gjøre i kommandolinjen. De ble også bedt om å komme frem til en forklaring på operatoren sammen med en medelev. Eksempel på dette er vist i Figur 3.3.

```
>>> 5 + 7
      12
>>> 6-7
      -1
>>> 5*2
      10
>>> 5%2
      1
>>> |
```

Figur 3.3: Eksempel på bruk av matematiske operatører i kommandolinje til Python

Elevene ble vist hvordan man definerer variabler i Python. Dette gjøres

ved å skrive `<Variabelnavn> = <Verdi>`. Jeg viste også at man kan få Python til å be om input fra brukeren, og at dette kunne bli gjort ved hjelp av koden `<Variabelnavn> = eval(input('<Tekst>'))`. '<Tekst>' er informasjon Python gir til brukeren av programmet, og kan for eksempel se ut som `'a = '`. `<Variabelnavn>` vil uansett ta verdien av det brukeren oppgir etter dette. Underveis fikk elevene mulighet til å teste disse tingene ut selv.

Etter at eleven hadde blitt vist, og fått prøvd å skrive og kjøre sitt første program, ble oppgavene som er vist i Figur 3.4 gitt. Tradisjonen tro var det første programmet til elevene `print('Hello, World!')`.

1. Skriv et program som printer differansen mellom to tall en bruker har oppgitt.
2. Skriv et program som printer summen av to tall en bruker har oppgitt.
3. Et partall er et heltall som har null i rest når det deles med 2; hva gjør programmet under? Skriv programmet inn og test det ut.

```
1 a = eval(input("Oppgi et heltall: "))
2 print(a%2==0)
```

4. Skriv et program som finner ut om  $a - b$  er større enn 0.  $a$  og  $b$  skal være bruker-input.
5. Skriv en kode der en bruker oppgir høyde og bredde til en trekant; programmet skal printe ut arealet til trekanten.

Figur 3.4: Oppgaver fra timen

Etter at elevene hadde gjort disse oppgavene gikk jeg gjennom dem på tavlen. Deretter fikk elevene oppgaven vist i Figur 3.5. De ble først vist stykket til venstre, et lineært ligningsett, og de ble bedt om å finne punktet der disse to linjene krysset. Dette skulle de gjøre på papir, ikke ved programmering. Dette temaet var noe de hadde jobbet med noen uker tidligere. Etter dette skulle de finne samme løsning, bare for de generelle ligningsettene, som er vist til høyre i figuren. Dette ble så brukt som utgangspunkt for et Python-program.

$\begin{aligned} 2x + 11y &= 4 \\ 3x + 14y &= 1 \end{aligned}$	$\begin{aligned} ax + by &= c \\ dx + ey &= f \end{aligned}$
--	--

Figur 3.5: To oppgaver med lineære ligningsett

Under vises et eksempel på hvordan man kan løse problemet til høyre. Fremgangsmåten er lik for det generelle ligningsettet som det er for det ikke-generelle ligningsettet, men i det ikke-generelle kan man utføre flere mellomregninger som “forenkler” prosessen. Eksempel på dette kan være  $(ae - bd)y = af - cd$  som i løsningen til ligningsettet til venstre ville hatt formen  $-5y = -10$ .

$$\begin{aligned} ax + by &= c \\ dx + ey &= f \\ \\ ax &= c - by \\ x &= \underline{\underline{\frac{c}{a} - \frac{b}{a}y}} \end{aligned} \tag{3.1}$$

$$\begin{aligned} d\left(\frac{c}{a} - \frac{b}{a}y\right) + ey &= f \\ cd - bdy + aey &= af \\ (ae - bd)y &= af - cd \\ y &= \underline{\underline{\frac{af - cd}{ae - bd}}} \end{aligned}$$

Selv om  $x$  i dette tilfellet er definert ved hjelp av en ukjent,  $y$ , trenger vi ikke gjøre mer med dette nå. Dette kommer av at Python leser fra toppen av et dokument og nedover, så vi kan definere  $y$  før vi prøver å definere  $x$ . Figur 3.6 viser hvordan Python-koden kan se ut.

```
print('Dette programmert finner krysningspunkt mellom to linjer gitt på formen')
print('ax + by = c')
print('dx + ey = f')

a = eval(input('a = '))
b = eval(input('b = '))
c = eval(input('c = '))
d = eval(input('d = '))
e = eval(input('e = '))
f = eval(input('f = '))

y = (a*f - c*d)/(a*e - b*d)
x = (c/a) - (b/a)*y

print('x = ')
print(x)
print('y = ')
print(y)
```

Figur 3.6: Eksempel på løsning av lineære ligningsett

## 3.2 Innsamling av data

Intervjuene er gjengitt i Tillegg A, intervjuguide for intervju med lærerne er gitt i Tillegg F, mens intervjuguide for intervju med elev er gitt i Tillegg G. Intervjuene med lærer ble gjennomført før intervjuene med elevene, og nummereringen som er gjengitt gir også rekkefølgen på intervjuene. Lærerne refereres til som  $L_1$  og  $L_2$ , mens elevene refereres til med E etterfulgt av tilsvarende suffiks. Hovedfokus for intervjuene med elevene var delen der de skulle løse to oppgaver. Disse to oppgavene vil bli beskrevet i de to neste delkapitlene.

### 3.2.1 Oppgave 1

Oppgaven kan bli funnet i Tillegg G. Denne oppgaven går ut på å lage et program som finner punktet der to rette linjer krysser hverandre. Ved å ta utgangspunkt i det elevene lærte i undervisningen vil det bli gitt en fasit på oppgaven her. I oppgaven ble det også gitt ett eksempel. Under vises derfor også løsning på eksempelet.

$$\begin{array}{l|l} y = ax + b & y = 3x - 10 \\ y = cx + d & y = -0,5x + 7,5 \end{array} \quad (3.2)$$

De generelle formene på ligningene (til venstre) ble ikke oppgitt til elevene, mens linjene til høyre ble vist i et koordinatsystem der kryssingspunktet ble markert med  $P$ . Neste steget er å sette  $y$ -verdiene til linjene lik hverandre, for så å utføre det som ofte omtales som *flytte-bytte*.

$$\begin{array}{l|l} ax + b = cx + d & 3x - 10 = -0,5x + 7,5 \\ ax - cx = d - b & 3x + 0,5x = 7,5 + 10 \\ x(a - c) = d - b & 3,5x = 17,5 \\ x = \frac{d-b}{a-c} & x = 5 \end{array} \quad (3.3)$$

Dersom  $y$ -verdiene til linjene er like, må  $x$ -verdiene være det vi har kommet frem til i Ligning 3.3. Siden vi er ute etter kryssingspunktet vet vi at dette er tilfellet. For å avslutte venstre side setter vi  $x = 5$  og finner en  $y$ -verdi.

$$\begin{array}{l} y = 3 \cdot 5 - 10 \\ y = 15 - 10 = 5 \end{array} \quad (3.4)$$

Linjene i eksempeloppgaven krysses dermed i punkt  $(x, y) = (5, 5)$ . Programmet elevene ble bedt om å skrive kan se ut som vist i Figur 3.7.

```
print('Dette programmert finner kryssningspunkt mellom to linjer gitt på formen')
print('y = ax + b')
print('y = cx + d')

a = eval(input('a = '))
b = eval(input('b = '))
c = eval(input('c = '))
d = eval(input('d = '))

x = (d - b)/(a-c)
y = a*x + b

print('x = ')
print(x)
print('y = ')
print(y)
```

Figur 3.7: Eksempel på løsning av Oppgave 1



Dersom vi kjører dette programmet får vi resultatet som vist i Figur 3.8. Her er det som er skrevet i blått automatisk skrevet ut av programmet, mens det i svart er bruker-input. Etter ett bruker-input må brukeren trykke Enter. Siden programmet kjørte i Python-Shell kommer `>>>` etter at programmet er ferdig å kjøre. Dette betyr at Python har kjørt programmet og er klar til å ta imot nye kommandoer.

```

    Dette programmert finner krysningspunkt mellom to
    linjer gitt på formen

    y = ax + b

    y = cx + d

    a = 3

    b = -10

    c = -0.5

    d = 7.5

    x =

    5.0

    y =

    5.0

    >>>
```

Figur 3.8: Output fra program i Oppgave 1

Det er mulig å presentere denne teksten bedre ved å formatere teksten. De første tre `print`-ene kunne vært én ved å bruke `\n`:

```
print('Dette programmert finner krysningspunkt mellom to
linjer gitt på formen\ny = ax + b\ny = cx + d').
```

Denne kodesnutten ville gitt samme resultat (output) som den der tre `print`-er ble brukt. Resultatet kunne blitt printet på én linje ved å bruke kommandoen `.format()`:

```
print('x = {}, y = {}'.format(x, y)).
```

Denne kodesnutten gjør at resultatet ville blitt printet på formen:

```
x = 5.0, y = 5.0
```

Grunnen til at dette ikke er ”fasiten” her er fordi det bare var en av elevene som brukte dette av alle elevene i de tre klassene. Dette er noe som godt kan introduseres når man begynner å tenke på å presentere resultat på en fin måte, noe som først krever kjennskap til hvordan man skal komme frem til resultat.

### 3.2.2 Oppgave 2

I den siste oppgaven elevene skulle gjennomføre, oppgave 2, som også kan finnes i Tillegg G, ble elevene gitt tre funksjoner:

$$\begin{aligned} f(x) &= ax^2 + bx + c \\ x_1 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\ x_2 &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} \end{aligned} \tag{3.5}$$

Oppgaven ville at elevene skulle skrive et program som tar imot verdier fra  $f(x)$  og som deretter printer ut  $x_1$  og  $x_2$ . For å hjelpe elevene ble det gitt en figur over en generell andregradsfunksjon, og jeg hadde med ett transparentark med koordinatsystem printet på. Elevene ble bedt om å bruke denne figuren og transparent arket til å vise omtrent hvor nullpunktene var plassert på figuren. Eksempel på fasit er gitt i Figur 3.9.

```
from math import sqrt

print('Dette programmert finner nullpunkt til f(x) = ax^2+bx+c')

a = eval(input('a = '))
b = eval(input('b = '))
c = eval(input('c = '))

x1 = (-b+sqrt(b**2 - 4*a*c))/(2*a)
x2 = (-b-sqrt(b**2 - 4*a*c))/(2*a)
print('x1 =')
print(x1)
print('x2 = ')
print(x2)
```

Figur 3.9: Eksempel på løsning av Oppgave 2



# Kapittel 4

## Analyse og diskusjon

Før de fem elevene som ble intervjuet begynte å løse oppgavene ble de spurt om motivasjon for matematikk og tidligere erfaring med programmering. Alle fem mente de var motivert for faget. Dette kom enten av muligheter som gode karakterer gir, eller at de så for seg utdanninger som krevde matematisk kompetanse.  $E_4$  sin motivasjon kom av at matematikk var favorittfaget. Med unntak av  $E_1$  og  $E_5$  hadde elevene som ble intervjuet tidligere erfaringer med programmering fra ungdomsskole. Selv om lærerne tidligere hadde hatt kurs om Python var det tydelig at de mente at de ikke mestret programmering. Transkripsjon 4.1 viser hva  $L_1$  svarte på spørsmål om hen hadde fått noe ut av programmeringskurs.

- 
- 4  $L_1$ : Jeg har jo gått på masse kurs.  
5 M: Masse kurs med programmering da?  
6  $L_1$ : Mhm, ja.  
7 M: Og det var der du sa du ikke hadde fått så mye ut av det eller?  
8  $L_1$ : Nei altså, nivået ligger litt for høyt på alle kursene, så jeg klarer ikke å forstå språket. Så jeg forstår ikke hvorfor jeg skal skrive det jeg gjør.
- 

Transkripsjon 4.1: Intervju  $L_1$ , linje 4 og 8, fra Tillegg A.1

I linje 7-8 kommer det frem at læreren ikke klarte å forstå Python-språket, selv med tidligere erfaringer, og ikke ser helt poenget med bruk programmering i matematikkundervisningen. Disse to elementene henger sammen, det er vanskelig å se poenget i noe man ikke helt forstår. Transkripsjon 4.2 fra  $L_2$  viser at læreren prøvde å følge med på hva som ble undervist i timen, og klarte det til en viss grad. Her påpeker læreren at det var vanskelig når elevene brukte IDLE og læreren selv brukte Anaconda. Begge disse er IDE-er,

og selve Python-koden skal være lik, men det er noe forskjeller på programmet sitt utseende. I tillegg kan oppsettet av de to ulike programmene gjøre funksjonaliteten forskjellig fra maskin til maskin.

- 
- 28 L<sub>2</sub>: Ehhh, sånn generelt, læringsmessig så har du nok den fordelingen der noen sitter veldig og følger med og noen som ikke er så interessert. Sånn som du ville fått i en vanlig undervisningsituasjon. Eh, men når du først fikk dei som på en måte, jeg satt jo kanskje ved siden av noken som ikke var så interessert i å lære seg dette her, ehh, men når de da begynte å pushe på de og sånn så opplevde jeg at de stod fast på de samme tingene som jeg stod fast på.
- 29 M: Ja, og det var?
- 30 L<sub>2</sub>: Ehm, det meste, hehe, neida, det var ikke, det var ikke det meste. No har jeg på en måte skrivi inn og sånn men og for meg var det nesten en utfordring at de skrev i Python og jeg hadde anaconda og sånne ting. For meg er det liksom nok til å vippe lasset littegran. Men jeg så jo da at de, at de kanskje ikke var flinke nok til å følge skrivemåten da. Sant.
- 

Transkripsjon 4.2: Intervju L<sub>2</sub>, linje 28 - 30, fra Tillegg A.2

Fra observasjonene som ble gjort i oppleggene kom det frem feil som gikk igjen blant flere elever. I listen under har jeg kategorisert de feilene som har vist seg å oppstå mange ganger i løpet av prosjektet.

1. Bytte bokstaver/bruker feil bokstav, eksempelvis a ble til d, e ble til c
2. Blande mellom store og små bokstaver, eksempelvis Print/print, X/x
3. Bruke udefinerte variabler/variabler før de ble definert.
4. Regne på begge sider av "=="
  - Utregning skal se ut som `<Regnestykke> == <Regnestykke>`
  - og er enten sant (`True`) eller usant (`False`)
  - (`<Regnestykke>` har her formen `<Float/Int> <Operator> <Float/Int>...`)
5. Bytte side av "==" (variabelnavn på feil side)
  - Skal være `<Variabelnavn> = <Verdi (type)>`

Punkt 1 og 2 viser til kodefeil. Dette er en vanlig kilde til feilmelding. For å gjøre det enklere for elevene å rette slike feil kan det være nyttig å vise at feilmeldingene i Python alltid viser linjenummer på hvor feilen oppstod. Inntrykket jeg fikk fra timen var også at Anaconda ga bedre feilmeldinger enn IDLE. At elevene bytter bokstaver betyr enkelt at elevene på ett tidspunkt brukte en bokstav, for eksempel a, og senere ved en feil brukte d der det skulle vært en a. Dette kom ofte fra at elevene gjorde utregninger med penn og papir først eller leste av fra tavlen og overførte resultater fra dette til sin datamaskin. Punkt 3, 4 og 5 omhandler utførelse av grunnleggende regning med variabler i Python. Den formen for programmering elevene lærte i opplegget handlet i stor grad om å mestre dette.

I Transkripsjon 4.3 ser vi at det ikke er automatikk i at elevene klarer overgangen fra å regne på papir til å føre dette inn i Python. Her har eleven løst eksempeloppgaven og skal generalisere problemet slik at man kan få Python til å finne kryssing mellom to vilkårlige rette linjer. Eksempelet som var gitt i oppgaven er

$$\begin{aligned} y &= 3x - 10 \\ y &= -0.5x + 7,5 \end{aligned} \quad (4.1)$$

I linje 138 påpeker jeg for eleven det som punkt 4 henviser til, hvordan Python kan utføre regnestykker. Eleven vet hvordan slike problem skal løses, hen hadde akkurat løst eksempeloppgaven, men får problemer med å finne en måte å “definere x på for Python”. Dette ser vi i linje 128, 132, 139, 142 og 144. E<sub>4</sub> “oppsummerer” tankegangen bra i linje 146: “*Men det er anderledes på papir enn det er på python.*” På papir kan man “hoppe” fra en del til en annen, mens Python jobber i rekkefølge og får problemer når noe er udefinert. Løsningen på E<sub>4</sub> sitt problem er at det går an å definere y uten bruk av ukjente etter at man har funnet  $x = \frac{-10-y}{3}$  eller lignende. Når man da skriver programmet bryter man *rekkefølgen* oppgaven ble løst med; Python må finne y først. Fra Schoenfeld, 1992, s. 355-356, vet vi at elever som ikke trent på problemløsningsstrategier lett kan sette seg fast på en framgangsmåte, selv når hen ikke har noe fremgagn. Hadde eleven “tatt hintene” jeg kom med - “*.. så du må gjøre noen utregninger først, før du kan skrive det inn i python.*” linje 127 og “*Ja. mhm så da må du regne litt ut først kanskje. ...*” linje 135 hadde hen kanskje endret strategi. På den andre side kan det hende at eleven fokuserer på den *relasjonelle forståelsen*, mens jeg som lærer i dette tilfellet “skubbet” eleven i retning *instrumentell forståelse*. Tolkningen av hva jeg sa da blir “*Bare gjør dette, så får vi rett svar!*” i motsetning til å gå inn i en samtale som “*det vi gjør nå er å lage system (CAS) utviklet av mennesker slik at vi kun trenger å løse algebraiske problem én gang, ..., men det er vi som faktisk må løse problemene.*”. Det er en forskjell på å utvikle CAS og

å bruke CAS. Eleven kan for eksempel huske tilbake to uker, da de jobbet med CAS/GeoGebra. I dette tilfellet løste datamaskinen problemet uten at eleven så hvorfor programmet klarte dette.

- 
- 127 M: .. så du må gjøre noen utregninger først, før du kan skrive det inn i python.
- 128 E<sub>4</sub>: Ja... men jeg kan ikke bare sette inn hvilke som helst x for jeg har ikke definert x.
- 129 M: Så du må finne en x? Det du gjorde i denne oppgaven, her fant du en x
- 130 E<sub>4</sub>: Jeg brukte x og flyttet disse sammen..
- 131 M: Prøv det
- 132 E<sub>4</sub>: men jeg har fortsatt ikke definert x-en her.
- 133 M: Men det gjorde du ikke i denne oppgaven heller. Du flyttet disse dit og så var x alene. Så kom du til at x er lik fem.
- 134 E<sub>4</sub>: Men man trenger ikke det når man skriver det på ark. Men for at programmet skal forstå det må man definere en variabel.
- 135 M: Ja. mhm så da må du regne litt ut først kanskje. For her er x den ukjente, så du kan flytte litt rundt. Så får du at x er noe kjent, men ikke med tanke på tall men bokstaver..
- 136 E<sub>4</sub>: Ja ..  
[Elev skriver inn i Python og prøver å kjøre program]
- 137 E<sub>4</sub>: Dette fungerte ikke.
- 138 M: For python klarer å gjøre det.. utregning på begge sider av erlikhetstegn?
- 139 E<sub>4</sub>: Nei... men jeg mangler noe.
- 140 M: Ja! Mitt råd er å regne ut forhånd, som du gjorde her, nedover på arket her og se hva du får. Finne noe python kan klare.  
[Elev skriver inn i Python]
- 141 M: Husk gangetegn mellom to ting som her.
- 142 E<sub>4</sub>: Og så må jeg snu om på formelen, sånn at jeg får x alene.
- 143 M: Du kan jo prøve å kopiere det du gjorde her men bytte ut tallene med bokstaver.
- 144 E<sub>4</sub>: Men vi må jo definere x-en.
- 145 M: Men her definerte du ikke x-en når du gjorde dette.
- 146 E<sub>4</sub>: Men det er anderledes på papir enn det er på python.
- 

Transkripsjon 4.3: Intervju E<sub>4</sub>, linje 127 - 146, fra Tillegg A.6

Transkripsjon 4.4 viser et eksempel på en elev som fokuserer på det instrumentelle for å løse oppgaven.



---

62	M:	Sånn at vi får en verdi på x, den ene i dette punktet.
63	E <sub>3</sub> :	Så då ville du fått xen på en side og tallene over på en annen side. [Elev regner på ark] Då må vi flytte å bytte. [Elev skriver i ca. 1 min og 40 sek]
64	M:	Kan jeg bare få se på linjene nå. Det ser ikke ut som at det er rett. Du ser du har minus der.
65	E <sub>3</sub> :	Åå, ja då blir jo det pluss da.
66	M:	Ja, ja litt fortegnsfeil, men ellers rett.
67	E <sub>3</sub> :	Men jeg må fortsatt flytte den over.
68	M:	Ja.
69	E <sub>3</sub> :	Så den over der. ... Men blir de, siden de blir liksom ... Blir den pluss da.
70	M:	Nei, ikke med denne, minus og pluss i addisjon blir minus.
71	E <sub>3</sub> :	Så må vi dele det.

---

Transkripsjon 4.4: Intervju E<sub>3</sub>, linje 62 - 71, fra Tillegg A.5

“Så då ville du fått xen på en side og tallene over på en annen side. [Elev regner på ark] Då må vi flytte å bytte.” fra linje 63 viser at eleven husker flytte-bytte regelen fra tidligere. Denne regelen sier at man kan “flytte” fra en side av erlighetstegnet til det andre, og at man da må “bytte” fortegn. Samtidig ser vi at eleven ikke har full kontroll over denne regelen: I linje 64 peker jeg på en fortegnsfeil i utregningen ... *Det ser ikke ut som at det er rett. Du ser du har minus der.* E<sub>3</sub> svarer i linje 65 med “Åå, ja då blir jo det pluss da.” og videre i linje 67 med “Men jeg må fortsatt flytte den over.”

Å lage dataprogram krever at eleven mestrer syntaksen i programmeringsspråket. Fra tidligere har elevene blitt kjent med å løse problem i matematikk. Siden elevene har erfaring med de matematiske problemene man møter i klasserommet, kan det med første øyekast virke som at det er enklere for dem å løse matematiske problemer enn å skrive programmer som en del av problemløsningprosessen. Dette står i kontrast til hva Sfard og Leron, observeerte, nemlig at programmering kan endre problemløsningprosessen slik at elevene jobber mer iherdig og bruker mer tid på å løse et problem. Dette kan ha noe med tidligere erfaringer til elevene. Sfard og Leron, 1996 ble skrevet før bruk av verktøy som GeoGebra ble en del av læreplanen. Et annet poeng kan være at elevene i de klassene jeg besøkt ikke hadde programmert i matematikkundervisning før opplegget, dermed ble intervjuene møte nummer to med programmering i matematikk for elevene. Dette vises i det neste eksempelet, fra intervjuet med E<sub>1</sub>. Transkripsjon 4.5 viser første reaksjon E<sub>1</sub> hadde på oppgave 1 (Tillegg G).

- 
- 29 M: Også får du hjelp fra meg og jeg kommer til å være støtte her, også kan du lese oppgaven først. Du kan lese inni deg. [Elev begynner å lese oppgave 1]  
[Ca. 12 sek med lesing]
- 30 E<sub>1</sub>: Ææ, lage et program?
- 31 M: Du skal lage et program som skal finne, du får oppgitt to linjer, og så skal du lage et program som finner punktet der disse to rette linjene krysser.. I x-y koordinater.
- 32 E<sub>1</sub>: Ja, mm.
- 33 M: Hva tenker du først om det?
- 34 E<sub>1</sub>: Æ, ja, æ, jeg er litt usikker på hva det menes med å lage et program.
- 35 M: Du skal få en datamaskin til å løse det for deg.
- 36 E<sub>1</sub>: Åja, så ved hjelp av programmering da liksom?
- 37 M: Ja, ved hjelp av programmering.
- 38 E<sub>1</sub>: Åja ja. Ja det er vel noen formler å sånn man må bruke da. Litt usikker. Jeg tenker umiddelbart på det der GeoGebra, så bruker man jo sånn funksjon da, der man velger begge linjer så kommer skjæringspunktet fram. Men jeg vet ikke helt hvordan jeg skal programmere for å finne det.
- 

Transkripsjon 4.5: Intervju E<sub>1</sub>, linje 29 - 38, fra Tillegg A.3

Her trenger eleven hjelp til å komme i gang med selve programmeringen. Det er ikke gitt at eleven forstår hva oppgaven går ut på. I linje 35; “du skal få en datamaskin til å løse det for deg” svarer eleven “Åja, så ved hjelp av programmering”. Her trenger eleven hjelp til å forstå hva det er programmeringen egentlig går ut på. Eleven peker så på at det er mulig å gjennomføre oppgaven ved å ta i bruk Geogebra, noe som absolutt er et viktig poeng. I GeoGebra kan man skrive inn formler for to linjer og få programmet til å finne krysningspunkt ved å bruke kommandoen *Skjæring mellom to objekt*. Eleven ser kanskje ikke fordelene med å løse problemet med Python, og har allerede en strategi for å ta i bruk Geogebra. Dette er det naturlig å forvente når en starter med programmering som verktøy i undervisningen. Eleven tar med seg kunnskaper og ferdigheter fra andre digitale verktøy inn i problemløsningen. Foreløpig har denne eleven ikke innsett fordelene det å lage sine egne programmer kan ha, det at man bruker digitale verktøy for å jobbe matematisk.

Vi kan ikke helt avskrive det Sfard og Leron, 1996 skriver i «Just Give me A Computer and I Will Move the Earth». I Transkripsjon 4.6 ser vi noe av det de beskrev; at en elev ikke er ferdig med en oppgave etter første forsøk.

---

165	M:	Ja. Hvordan får du programmet til å vise hva som er funnet?
166	E <sub>1</sub> :	Var ikke det bare man skulle trykke på run?
167	M:	Prøv det. [Programmet kjører]
168	M:	Oj, en feilmelding.
169	E <sub>1</sub> :	Ja, skal jeg inn og sjekke da. Det står name e is not defined. Og da på grunn av jeg skrev jo e i stedet for c i formelen min. Så bare endre den til c. [Programmet kjører på nytt]
170	M:	Hvorfor viser ikke programmet noe svar?
171	E <sub>1</sub> :	Det vet jeg ikke.
172	M:	Husker du print kommandoen.
173	E <sub>1</sub> :	Ja print ja. ... Hvordan var det igjen?

---

Transkripsjon 4.6: Intervju E<sub>1</sub>, linje 165 - 173, fra Tillegg A.3

Her leser eleven en feilmelding “Det står name e is not defined.” og videre finner eleven hvor denne feilen er ”Og da på grunn av jeg skrev jo e i stedet for c i formelen min. Så bare endre den til c.”. En av de åtte matematiske kompetansene handler om å tenke matematisk. I problemløsning elevene har møtt tidligere er det forventet at dette ikke er ett problem, de eneste verdiene som brukes er  $a$ ,  $b$ ,  $c$  og  $d$ . Dersom  $e$  også ble brukt kan dette føre til at variabler blandes uavhengig om det gjøres på datamaskin eller gjennom vanlig problemløsning.

Transkripsjon 4.7 viser E<sub>2</sub> under løsning av Oppgave 2. I denne oppgaven fikk elevene formlene  $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$  og  $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$  og skulle bruke dette til å finne nullpunktene til funksjonen  $f(x) = ax^2 + bx + c$ . Her ser vi at eleven jobber med å finne ut hva som skal inn i programmet. Det fremstår som at eleven har kontroll på hva hen vil programmere og at fokuset er hvordan algoritmen skal se ut, for eksempel med *hva* som skal inn i programmet og hvordan.

- 
- 181 M: Og hva trenger datamaskinen for å løse problemet?
- 182 E<sub>2</sub>: Åsså no har vi jo allerede, noe har vi jo allerede a b og c.
- 183 M: Ja.
- 184 E<sub>2</sub>: så no må vi jo liksom finne forskjellen mellom disse to. Og det er jo at det står pluss her og minus her. Også må jeg ikke skrive inn denne i programmet da? Skrive inn abc-formellen inn i programmet?
- 185 M: Du kan jo prøve og se hva som skjer. [Elev skriver ca. 28 sek]
- 186 E<sub>2</sub>: Jeg jo, jeg kan jo gjøre slik at mmmm.. at i hvertfall denne blir en egen variabel.
- 187 M: Det kan hende.
- 188 E<sub>2</sub>: Ja. [Elev skriver/tenker i ca 22 sek]
- 189 M: Hva er det du... Jeg kan komme rundt for å se skjermen [går rundt bordet].
- 190 E<sub>2</sub>: Men no er jo spørsmålet om den oppfatter det riktig eller på en måte.
- 191 M: Hva mener du.
- 192 E<sub>2</sub>: Eller at liksom, eller jo denne er jo, bare glem det, det går sikkert. .. Sånn. ... x1 er lik ... Eller kanskje vi må ta det som er oppe her først også liksom.
- 193 M: Ser at timen deres er ferdig nå, dette er det siste, så håper det går fint at vi fullfører det.
- 194 E<sub>2</sub>: Ja, ja. [Elev jobber på pc, ca. 15 sek]. Oki, da blir det... [elev jobber ca 5 sek]
- 195 M: Når du gjør det på denne måten må du ha parentes rundt dette stykket, eller så deler den kun på denne med denne.
- 196 E<sub>2</sub>: Ja, selvfølgelig, oki.
- 197 M: Og gangetegn mellom tall. Nå har du skrevet for x1, siden x2 bare har en minus i stedet kan du bare copy-paste.
- 198 E<sub>2</sub>: Ja oki, ..., skal jeg printe det ut også?
- 199 M: Ja.
- 

Transkripsjon 4.7: Intervju E<sub>2</sub>, linje 181 - 185, fra Tillegg A.3

En betraktelig del av dette utklippet viser eleven i prosess med å analysere problemet. Fra et programmeringsperspektiv er Oppgave 1 og Oppgave 2 like. Dette betyr at elevene kan bruke Oppgave 1, som hen akkurat hadde løst som mal for å løse problemet i Oppgave 2.

Fra intervjuene kommer det også frem at bruk av generalisering også er et tema som bør analyseres og diskuteres. Argumentet som blir brukt her er at programmering kan hjelpe med å utvikle symbol- og formalismekompetansen til elevene. Som vist i metodedelen krever løsningen på oppgaven at elevene

jobber med de generelle formlene for å kunne løse oppgaven. Transkripsjon 4.8 er et eksempel på dette.

- 
- 25 M: Ja. Da har vi komnt til oppgavene. Jeg kan gjenta at dette ikke er en test eller prøve. Det er også litt nye ting. Om du kan snakke om hva du tenker eller skrive på ark. [Elev leser på ark]
- 26 E<sub>2</sub>: Ehm okei. Ehh.
- 27 M: Kan bare si at figuren bare er ett eksempel.
- 28 E<sub>2</sub>: Jajaja! To linjer skal finne hverandre. Der de krysser. Altså to punkter då må de jo være sånn, .., eller to linjer. Er de to linjer som ikke stopper på en måte?
- 29 M: Mhm.
- 30 E<sub>2</sub>: Da må de jo. Ehm då må de jo være to formler ikke sant, på en måte.
- 31 M: Hvordan tror du formlene ser ut?
- 32 E<sub>2</sub>: Ehm. Eh ja, for dette er bare et eksempel.
- 33 M: Ja.
- 34 E<sub>2</sub>: Jeg er litt usikker men.
- 35 M: Bare prøv. Du kan skrive ned det du tenker også. ... Hva er du usikker på?
- 36 E<sub>2</sub>: Ehm. Jeg tenkte to linjer. Liksom at man vet disse to forskjellige linjene. Å da kan man jo si at det blir liksom  $a x$  minus  $b$ .
- 37 M: Ja. Har du vært borti denne formelen før?
- 38 E<sub>2</sub>: Ehm, jeg har vel det, men jeg husker ikke.
- 39 M: Om du skrev  $ax$  pluss  $b$  da?
- 40 E<sub>2</sub>: Ja, jajaja! Ja oki. Det at den har negativ verdi betyr jo at den begynner under. [Indikerer under x-aksen i eksempelet]
- 

Transkripsjon 4.8: Intervju E<sub>2</sub>, linje 25 - 40, fra Tillegg A.4

I dette utdraget skjer det mye. I linje 28 - 30 ser vi eleven prøve å finne en strategi for å løse oppgaven. “*To linjer skal finne hverandre. Der de krysser. Altså to punkter då må de jo være sånn, .., eller to linjer. Er de to linjer som ikke stopper på en måte?*” og “*Da må de jo. Ehm då må de jo være to formler ikke sant, på en måte.*”. Her ser vi at eleven prøver å forstå oppgaven ved å stille spørsmål til egenskaper til linjene og videre kommer frem til at å løse dette krever bruk av formler. Det er også tydelig at eleven vet at de to formlene som er oppgitt ( $y = 3x - 10$  og  $-0,5x + 7,5$ ) bare er et eksempel på to linjer som krysser: “*Ehm. Eh ja, for dette er bare et eksempel.*”. Videre klarer denne eleven å gjenkjenne formelen  $y = ax - b$ . E<sub>5</sub> finner samme

generalisering, som vi ser i Transkripsjon 4.9.

---

67	M:	Nå har vi brukt disse eksemplene, det oppgaven spør om er
68	E <sub>5</sub> :	Lage en..
69	M:	At linjene ikke har oppgitt verdier der.
70	E <sub>5</sub> :	Mmm. At den tre x på en måte blir a og y er lik ax minus b og y er lik cx minus d.
71	M:	Ja. Kan si at jeg ville brukt pluss ikke minus, men det fungerer dette også.
72	E <sub>5</sub> :	Sånn at det blir som en formel på en måte.

---

Transkripsjon 4.9: Intervju E<sub>5</sub>, linje 67 - 70, fra Tillegg A.7

Dette viser at begge disse elevene ser overgangen  $y = 3x - 10 \rightarrow y = ax - b$ . E<sub>2</sub> påpeker også “Ja, jajaja! Ja oki. Det at den har negativ verdi betyr jo at den begynner under.” i linje 40. E<sub>2</sub> kom frem til dette etter at hen ble vist den vanlige formen  $y = ax + b$ . Dette kan vise mot at E<sub>2</sub> har en mer relasjonell forståelse av formelen. E<sub>5</sub> gjenkjenner dette som *en formel* i linje 72. Prosessen med generalisering hos elevene går altså til utbytting av tall til bokstaver. I et forsøk på å hjelpe kommer jeg med at det er mer hensiktsmessig å generalisere fortegnet i b-leddet til +.

I Transkripsjon 4.10 ser vi E<sub>3</sub> i samme del av Oppgave 1.

---

97	M:	Vet du hvilke formel du kan bruke for å beskrive en linje, der du ikke vet disse verdiene?
98	E <sub>3</sub> :	Da var det a ... ganger b?
99	M:	Ikke helt, ax pluss b.
100	E <sub>3</sub> :	a x pluss b, ja så da må jeg ... y er lik ax pluss b
101	M:	Ja. Og du vil finne kryssning mellom to ukjente linjer som ser sånn ut.
102	E <sub>3</sub> :	Jo, ja.
103	M:	... Så det du ser da, du har a og b i denne formelen sant. De står for...
104	E <sub>3</sub> :	Det vet vi, null koma fem og syv komma fem.
105	M:	Ja det var denne linjen. ... .. så kanskje vi kan bruke andre tall for en ny linje, men ha x og y med der..
106	E <sub>3</sub> :	Hæ?
107	M:	Altså at en av linjene er y er lik ax pluss b, så da kan vi bruke y er lik cx pluss d kanskje?
108	E <sub>3</sub> :	Ja, åja det var jo den delt på ...
109	M:	Du husker litt av den oppgaven fra timen altså?
110	E <sub>3</sub> :	Ja du brukte ax pluss b delt på ... da gikk den helt til f da?

---

Transkripsjon 4.10: Intervju E<sub>3</sub>, linje 97 - 110, fra Tillegg A.5

Etter oppmuntring fra meg prøver eleven å huske formelen for rette linjer. Linje 98, “*Da var det a ... ganger b?*” viser at eleven ikke husker formelen. Denne transkripsjonen viser hvordan jeg, som lærer, prøvde å få eleven til å løse oppgaven uten mye hell. Jeg gikk rett fra å gi formelen  $ax + b$  til krysningspunkt mellom to linjer. Det er ikke tydelig for meg (i ettertid) at eleven forstod formelen. Dette kommer til uttrykk i linje 105-106 der jeg sier “*Ja det var denne linjen. ... .. så kanskje vi kan bruke andre tall for en ny linje, men ha x og y med der.*” noe eleven reagerer på med “*Hæ?*”. Om vi tar utgangspunkt i Schoenfeld sine tre spørsmålene for å lære elever problemløsning - *What are you doing?* - *Why are you doing it?* - *How does it help you?* - kunne jeg formulert bedre spørsmål. Eksempelvis må eleven se poeng med å generalisere linjene. I linje 97 peker jeg på formlene gitt i eksempelet. Hvordan vil det hjelpe eleven å oversette  $y = 3x - 10$  til  $y = ax - b$ ?

Over har vi sett flere transkripsjoner fra intervjuene som ble gjennomført etter undervisningsøkten. Målet med intervjuene var å se om programmering kunne være et godt verktøy i matematikkundervisningen i 1T. Den første beslutningen man må ta i et forskningsprosjekt dersom forskningsintervju blir vurdert som metode, er om dette er den mest adekvate måten å besvare

forskningsspørsmålet på (Kvale & Brinkmann, 2019, s. 135). Spørsmålet er derfor, hvorfor ikke bruke spørreundersøkelse? Intervjuet brukes som støtte til observasjonene som blir gjort; det er ikke alltid slik av vi kan stole blindt på egne observasjoner og antagelser. Transkripsjonene representere en objektiv fremstilling for hvordan elevene taklet matematiske oppgaver som krever programmering. Jeg diskuterer flere steder med *jeg som lærer*. Det vil derfor være viktig å påpeke at jeg ikke er læreren til elevene, de kjente meg bare i to timer før intervjuet. Når elever deltar i et intervju er det en unaturlig situasjon for dem. Elevene har ikke nødvendigvis erfaring med å sette ord på tankene sine mens de jobber med matematikk. Det er også en unaturlig situasjon for elevene å sitte alene med en lærer, mens de løser oppgaver. Selv om det er mange ting som gjør at dette ikke er en “normal” undervisnings-situasjon, gir disse intervjuene en mye bedre forståelse for hvordan elevene forholder seg til programmering og algoritmisk tenkning enn vi hadde fått ved en spørreundersøkelse.



# Kapittel 5

## Konklusjon

Denne oppgaven tok utgangspunkt i problemstillingen

*Hvordan kan algoritmisk tenkning og programmering brukes som verktøy for å fremme elevenes forståelse for algebra?*

For å svare på dette spørsmålet ble det gjennomført ett undervisningsopplegg og samlet inn data fra elever i 1T. Flere av problemene som oppsto for elevene da de skulle bruke programmering som verktøy kan, og forhåpentligvis vil, forsvinne i de kommende årene, når den nye læreplanen er blitt implementert på alle nivåer i skolen. Den nye læreplanen sier at elever skal begynne å lære programmering tidlig i grunnskolen, og elevene vil da ha mer erfaring med å lese og skrive kode når de begynner med matematikk på videregående. Samtidig skal det være verdt å nevne at flere av elevene som deltok i opplegget hadde tidligere erfaringer med programmering fra grunnskolen. Uavhengig av elevene sine tidligere erfaringer med programmering vil det uansett være en overgang for elevene fra grunnskole til videregående. I denne overgangen, ifølge Piaget, går de fra det konkrete operasjonelle stadiet, til det formelle operasjonelle stadiet. Til å begynne med er programmering bare et verktøy for å undervise matematikk. Etter hvert blir programmering en del av matematikkfaget, der de må tenke mer abstrakt. I transkripsjonene ser vi elevenes tankegang når de blir bedt om å jobbe med abstrahering gjennom generelle formler.

I denne oppgaven har det blitt vist at programmering kan brukes sammen med algebraundervisningen i 1T. Transkripsjonene fra elevintervjuene viser flere aspekter ved denne innlemmingen. For eksempel så trengs det flere timer enn den ene elevene fikk for at de skal lære å programmere i matematikk. Samtidig så er det uproblematisk å inkorporere programmering og algoritmisk tenkning i matematikkfaget. Elevene skal uansett ha en overgang fra aritmetikk og mer uformell algebra til den algebraen som undervises i 1T. Fra

årsplanen til den ene klassen kommer det frem at de alt hadde jobbet med blant annet *Regnerekkefølge*, *Bokstavuttrykk*, *Faktorisering*, *Formler*. Disse punktene er alle sentrale for å lære seg å programmere løsninger til problem i algebra. Dersom elevene blir bedt om å skrive et program som finner krysningspunktet mellom to linjer må eleven bruke bokstaver i stedet for tall hele veien. Programmering vil da være et godt verktøy for å hjelpe eleven å tenke abstrakt.

Denne oppgaven samlet inn data over en veldig kort periode. Tankegang og ferdigheter utvikles over lengre perioder. Det er et godt tegn at elevene klarer å løse problemene som ble gitt i intervjuet, selv om det var med støtte. Vi ser også at flere av elevene valgte programmeringsfag på ungdomsskolen. Det er lite trolig at elevene vil miste interessen i matematikkundervisningen, når programmering blir innlemmet i faget. Mitt inntrykk fra elevene i undervisningsøktene og intervjuene var at de fikk bedre forståelse for hvorfor og hvordan vi generaliserer algebraiske uttrykk.

For å få til å innføre et nytt undervisningsopplegg på en god måte, må i tillegg lærerne ha troen på opplegget og være i stand til å innføre endringene. Ikke alle lærere tror de har den nødvendige kompetansen for å undervise *programmering og algoritmisk tenkning* enda. De to lærerne som deltok i dette prosjektet ga uttrykk for at de ikke trodde de hadde nok kunnskap om programmering til å undervise i dette.

Programmering og algoritmisk tenking vil være en viktig del av undervisningen fremover. De fleste elevene greide å forholde seg til programmering som del av matematikkundervisningen, men det var et behov for støtte fra lærer. En viktig ting å gjøre kan derfor være at man må utvikle undervisningsopplegg for programmering og algoritmisk tenkning i matematikkundervisningen, slik at lærene får det de trenger for å innføre den nye læreplanen. Da får elevene mer ut av å bruke programmering som verktøy, enn de ville fått av undervisning der programmeringen står separat fra de andre målene i læreplanen.

# Bibliografi

- Abonyi-Tóth, A. & Pluhár, Z. (2019). Wandering Micro:bits in the Public Education of Hungary [New Ideas in School Informatics]. *Informatics in Schools*. Springer. <https://doi.org/10.1007/978-3-030-33759-9>
- Berk, L. E. (2013). *Child Development* (Ninth Edition). Pearson.
- Brochmann, G. (2019). *Digitale prøvekaniner*. Hentet 25. september 2020, fra <https://www.nrk.no/ytring/digitale-provekaniner-1.14702337>
- Choen, J. S. (2002). *Computer Algebra and Symbolic Computation*. CRC Press.
- Eisenberg, M. (2017). Approaching Computer Science Education Through Making. I S. B. Fee, A. M. Holland-Minkley & T. E. Lombardi (Red.), *New Directions for Computing Education* [Embedding Computing Across Disciplines] (s. 35–44). Springer. <https://doi.org/10.1007/978-3-319-54226-3>
- Falch, C. G. (2020). *Et naivt og utdatert prosjekt*. Hentet 25. september 2020, fra <https://www.nrk.no/ytring/et-naivt-og-utdatert-prosjekt-1.14851532>
- Fee, S. B., Holland-Minkley, A. M. & Lombardi, T. E. (2017). Re-envisioning Computing Across Disciplines. I S. B. Fee, A. M. Holland-Minkley & T. E. Lombardi (Red.), *New Directions for Computing Education* [Embedding Computing Across Disciplines] (s. 1–11). Springer. <https://doi.org/10.1007/978-3-319-54226-3>
- Forsström, S. E. & Kaufmann, O. T. (2018). A Literature Review Exploring the use of Programming in Mathematics Education. *International Journal of Learning, Teaching and Educational Research*, 17(12), 18–32. <https://doi.org/10.26803/ijlter.17.12.2>
- Hattie, J. (2009). *Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement*. Routledge.
- Hvordan Lær Kidsa Koding! ble til*. (2015). Hentet 4. februar 2021, fra <https://www.kidsakoder.no/2015/03/24/hvordan-laer-kidsa-koding-ble-til/>

- Katz, V. J. (2014). *History of Mathematics* (Pearson New International). Pearson.
- Kieran, C. (2020). Algebra Teaching and Learning. I S. Lerman (Red.), *Encyclopedia of Mathematics Education* (Second Edition, s. 36–44). Springer. <https://doi.org/10.1007/978-3-030-15789-0>
- Kilpatrick, J. (2012). The new math as an international phenomenon. *ZDM Mathematics Education*. <https://doi.org/10.1007/s11858-012-0393-2>
- Kvale, S. & Brinkmann, S. (2019). *Det kvalitative forskningsintervju* (3. utgave). Gyldendal.
- Lamers, M. H., van der Putten, P. & Verbeek, F. J. (2014). Observations on Tinkering in Scientific Education. *Entertaining the Whole World*. Springer. <https://doi.org/10.1007/978-1-4471-6446-3>
- Lervik, J. O. (1984). *EDB: DATA VALGFAG I*. Aschehough. Hentet 8. september 2020, fra <https://www.nb.no/nbsok/nb/0c2271912932e607312e9404a7e898de?lang=no>
- Lewis, C. M. (2017). Good (and Bad) Reasons to Teach All Students Computer Science. I S. B. Fee, A. M. Holland-Minkley & T. E. Lombardi (Red.), *New Directions for Computing Education* [Embedding Computing Across Disciplines] (s. 15–34). Springer. <https://doi.org/10.1007/978-3-319-54226-3>
- Machenbach, I. (2019). *Nettbrett uten vett*. <https://www.nrk.no/ytring/nettbrett-uten-vett-1.14533370>
- Mason, J., Graham, A. & Johnston-Wilder, S. (2014). *Å lære algebraisk tenkning* (J. Lie, Overs.). Caspar forlag.
- Melby, G. (2020). *Må brukes med fornuft*. Hentet 25. september 2020, fra <https://www.nrk.no/ytring/ma-brukes-med-fornuft-1.15133276>
- Mellin-Olsen, S. (1984). *Eleven, matematikken og samfunnet* (1. utgave, 1. opplag) [En undervisningslære]. NKI-forlaget.
- Mooney, C. G. (2013). *Theories of Childhood: An Introduction to Dewey, Montessori, Erikson, Piaget, and Vygotsky* (Second Edition). Redleaf Press.
- National Research Council. (2000). *How People Learn* [Brain, Mind, Experience, and School].
- Niss, M. mfl. (2002). Kompetencer og matematiklæring: Ideer og inspiration til utvikling af matematikundervisning i Danmark (M. Niss & T. H. Jensen, Red.). *Uddannelsesstyrelsens temahæfteserie*, (18).
- Om Lær Kidsa Koding*. (udatert). Hentet 4. februar 2021, fra <https://www.kidsakoder.no/om-lkk/>
- Papert, S. A. (1980). *Mindstorms* [Children, Computers, and Powerful Ideas]. Basic Books.

- Programmering i skolen. (2016). *Utdanningsdirektoratet*. Hentet 4. februar 2021, fra [https://www.udir.no/globalassets/filer/programmering\\_i\\_skolen.pdf](https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf)
- Sauer, T. (2018). *Numerical Analysis* (Third Edition). Pearson.
- Schoenfeld, A. H. (1992). Learning to Think Mathematically: Problem Solving, Metacognition, and Sense Making in Mathematics. *Handbook of Research in Mathematics Teaching and Learning* (s. 334–370). National Council of Teachers of Mathematics.
- Sedgewick, R. & Wayne, K. (2011). *Algorithms* (Fourth Edition). Addison-Wesley.
- Sfard, A. & Leron, U. (1996). Just Give me A Computer and I Will Move the Earth [Programming as a Catalyst of a Cultural Revolution in the Mathematics Classroom]. *International Journal of Computers for Mathematics Learning*, 185–195.
- Skemp, R. R. (1976). Relational Understanding and Instrumental Understanding. *Mathematics Teaching*, (77), 20–26.
- Stephen, M. & Kadijevich, D. M. (2020). Computational/Algorithmic Thinking. I S. Lerman (Red.), *Encyclopedia of Mathematics Education* (Second Edition, s. 117–123). Springer. <https://doi.org/10.1007/978-3-030-15789-0>
- UDIR: *Algoritmisk tenkning*. (2019). Hentet 17. januar 2021, fra <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/#:~:text=Algoritmisk%20tenkning%20inneb%C3%A6rer%20%C3%A5%20bryte,komme%20fram%20til%20%C3%B8nsket%20%C3%B8sning>.
- UDIR: *Grunnleggende ferdigheter*. (2020). Hentet 29. september 2020, fra <https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/grunnleggende-ferdigheter/>
- UDIR: *MAT03-02*. (2020). Hentet 15. januar 2021, fra <https://www.udir.no/lk20/mat03-02>
- UDIR: *MAT1-01*. (2006). Hentet 15. januar 2021, fra <https://www.udir.no/kl06/mat1-01>
- UDIR: *MAT1-04*. (2013). Hentet 15. januar 2021, fra <https://www.udir.no/kl06/MAT1-04#>
- UDIR: *Matematikk T (MAT09-01): Kompetansemål og vurdering*. (2020). <https://www.udir.no/lk20/mat09-01/kompetansemal-og-vurdering/kv42>
- Visible learning*. (2020). Hentet 10. september 2020, fra <https://visible-learning.org/>

- Xie, B., Nelson, G. L. & Loksa, D. (2019). A theory of instruction for introductory programming skills. *Computer Science Education*. <https://doi.org/10.1080/08993408.2019.1565235>
- Zelle, J. (2010). *Python Programming: An Introduction to Computer Science* (2. utg.). Franklin, Beedle & Associates Inc.

Tillegg A

Transkripsjon





Kodene i transkripsjonen er;  $M$  for meg (intervjueren),  $L_1/L_2$  for de to lærerne (intervjuobjekt) og  $E_n$  ( $n = 1, 2, \dots, 5$ ) for de ulike elevene (intervjuobjekt). Kommentarer kommer inni [ ]. Disse viser til hva som skjedde der det ikke er helt tydelig i dialogen hva som skjer.

## A.1 Intervju L<sub>1</sub>

- 
- 1 M: Ja, det første jeg tenkte å spørre om er om du har noen tanker om programmering sin plass i undervisningen, for eksempel matematikkundervisning. Har du gjort noen tanker om det?
- 2 L<sub>1</sub>: Em - ja jeg har jo det. Jeg klarer ikke å forstå hvorfor vi må gjøre det når vi har sånne ting som GeoGebra og CAS og sånt. Men jeg kan jo godt forstå at de har lyst til at vi skal utvide liksom perspektivet da.
- 3 M: Det neste spørsmålet er om du har gjort noen forberedelser i forhold til den nye læreplanen, spesielt med tanke på digital kompetanse og programmering.
- 4 L<sub>1</sub>: Jeg har jo gått på masse kurs.
- 5 M: Masse kurs med programmering da?
- 6 L<sub>1</sub>: Mhm, ja.
- 7 M: Og det var der du sa du ikke hadde fått så mye ut av det eller?
- 8 L<sub>1</sub>: Nei altså, nivået ligger litt for høyt på alle kursene, så jeg klarer ikke å forstå språket. Så jeg forstår ikke hvorfor jeg skal skrive det jeg gjør.
- 9 M: Ja, ok. Jeg er også interessert i klassen. Jeg har jo hatt to klasser med deg, og lurte på om du kan beskrive klassene sånn nivåmessig.
- 10 L<sub>1</sub>: Mhm, det er to ganske forskjellige klasser. Den ene er liten med få elever og der er det, ja sånn jevnt fordelt da med alle liksom, hva skal jeg si, karakterene. Det er noen høye, noen lave, veldig sånn jevnt. De er veldig ivrige, alle.
- 11 L<sub>1</sub>: Og den andre klassen der er det... det er en stor gruppe, der er det, ja hva skal jeg si. De sliter med mye grunnleggende regning, men det var jo sånn som du sa, noen som hadde forkunnskaper i programmering, og det, det visste ikke jeg.
- 12 M: Nei, jeg var overasket over det jeg og, at jeg så noen som helt klart hadde programmert før. For det jeg så, hvertfall den siste klassen var at elevene jobbet med programmering hele tiden med oppgavene og sånt. Er det normalt?
- 13 L<sub>1</sub>: Ja, det er veldig normalt.
- 14 M: Men de er likevel ikke alle som har gode karakterer?
- 15 L<sub>1</sub>: Nei, det er mange der som faktisk ikke ville fått bestått. De er med på ett sånt prosjekt hvor de har vurdering uten karakter, så de vet ikke at de ikke ville fått bestått. Men de er alle ivrige, alle der er veldig ivrige.
- 16 M: Så det er normalt at de jobber, eller prøver å jobbe?
- 17 L<sub>1</sub>: Absolutt.
- 18 M: Men da, når de ikke får til oppgaver, gir de opp eller fortsetter de å prøve?
- 19 L<sub>1</sub>: De fortsetter å prøve
- 20 M: De forsetter å prøve ja. Mhm.
- 21 M: Hvilket fokus har dere hatt på digital kompetanse i undervisningen?
- 22 L<sub>1</sub>: Frem til nå?
- 23 M: Frem til nå ja. Og ikke bare programmering, har dere hatt Geogebra ...
- 24 L<sub>1</sub>: Veldig lite
- 25 M: Veldig lite ja.
- 26 L<sub>1</sub>: Vi har hatt litt GeoGebra, men vi har først fokusert på brøkgregning rett å slett.
- 27 M: Ja, mhm. Har du noen andre kommentarer fra undervisningøkten jeg var i. For du har jo vært i undervisningen før med disse elevene. Var det noe du observerte som var anderledes?
- 28 L<sub>1</sub>: Med de? Nei.
- 29 M: Nei
- 30 L<sub>1</sub>: Nei
- 31 M: Så helt vanlig time
- 32 L<sub>1</sub>: Ja

33 M: Mhm

[Intervju avsluttes]

---

## A.2 Intervju L<sub>2</sub>

- 
- 1 M: Det første som jeg tenkte å spørre om er om du har noen tanker om programmering sin plass i matematikkundervisning.
- 2 L<sub>2</sub>: Mmm
- 3 M: Vi har snakket litt om det allerede men..
- 4 L<sub>2</sub>: Ja, jeg tenker jo at programmering har sin plass i matematikfaget, men litt som vi snakket om tidligere at kanskje det skulle vært et eget fag sånn at vi fikk litt mer kontinuitet gjennom hele året.
- 5 M: Ja.
- 6 L<sub>2</sub>: I steden for at no blir det på en måte lagt på toppen på et fag som i tillegg har en ny læreplan. Også vet jeg at det er mange lærere, inkludert meg selv som ikke har programmeringskompetanse i det hele.
- 7 M: Mhm.
- 8 L<sub>2</sub>: So, akkurat no så syns jeg det er litt tungt og litt uoverkommelig med at programmering skal inn, fordi jeg syns det er vanskelig.. eller jeg har ikke kapasitet til å på en måte, eh lære meg det då. Så skal du i tillegg stå foran en klasse lære no du ikke selv kan. Og det gjør jo noke med deg som lærer. Sånn i forhold til.. Det er ikke så gøy og stå der og være inkompetent då.
- 9 M: Nei. Men du er i hvertfall positiv til programmering i matematikkundervisning eller?
- 10 L<sub>2</sub>: Jaa, da tenker jeg at vil være.. eh litt sånn både og. Jeg syns det er bra det kommer, men samtidig så tenker jeg og litt at da kan være litt for spesielt interreserte. Jeg kan veldig lite om programmering så jeg vet ikke. Men i utgangspunktet så tenker jeg at det er for de som er litt mer matematisk .. eller realfaglige så. Og lt kan du i og for seg ta selv om du går for litt mer samfunsfaglig matte da.
- 11 M: Ja.
- 12 L<sub>2</sub>: Etter hvert. Så litt sånn både og.. Blir spennende og se.
- 13 M: Ja. Og det jeg også lurer på er hvordan du har forberedt deg til .. altså både dem nye læreplanen, men også rettet mot programmering.
- 14 L<sub>2</sub>: Har ikke fått forberedt oss i det hele. Men det er jo mest sansynlig på grunn av korona-situasjonen
- 15 M: Ja
- 16 L<sub>2</sub>: Rett og slett det blei jo en gangske heftig avslutning på skoleåret i fjor. So da er jo på en måte ett av mine anker-punkter, jeg har rett og slett ikke tid til å forberede meg.
- 17 M: Har du tatt noe kurs?
- 18 L<sub>2</sub>: Nei.. Me har hatt to kurs på skolen med noen lærere..
- 19 M: Noen lærere som kan programmering.
- 20 L<sub>2</sub>: Ja
- 21 M: Ja.
- 22 L<sub>2</sub>: Så da er alt jeg kan. Og jeg fulgte jo med på ditt kurs og jeg føler jo at jeg ser hva som skal gjøres når noen står over meg og sier hva som skal gjøres, men når jeg er på egenhånd så mister jeg tråden.
- 23 M: Du deltok jo i timen
- 24 L<sub>2</sub>: Ja
- 25 M: Og da fikk du hjelp av noen elever.
- 26 L<sub>2</sub>: De hadde hverfall bedre peiling en meg, hehe, for å si det sånn, men sånn som jeg på en måte tolka det så var det to stykker som på en måte hadde hatt litt programmering fra før mens resten av klassen, sånn som jeg oppfattet det der å då, hadde hadde liten erfaring
- 27 M: Ja, det stemmer nok... Har du noen andre observasjoner fra klassen?
- 28 L<sub>2</sub>: Eh, sånn generelt, læringsmessig så har du nok den fordelingen der noen sitter veldig og følger med og noen som ikke er så interessert. Sånn som du ville fått i en vanlig undervisnings situasjon. Eh, men når du først fikk dei som på en måte, jeg satt jo kanskje ved siden av noken som ikke var så interessert i å lære seg dette her, eh, men når de da begynte å pushe på de og sånn så opplevde jeg at de stod fast på de samme tingene som jeg stod fast på.
- 29 M: Ja, og det var?

- 30 L<sub>2</sub>: Ehm, det meste, hehe, neida, det var ikke, det var ikke det meste. No har jeg på en måte skrivi inn og sånn men og for meg var det nesten en utfordring at de skrev i Python og jeg hadde anaconda og sånne ting. For meg er det liksom nok til å vippe lasset littegran. Men jeg så jo da at de, at de kanskje ikke var flinke nok til å følge skrivemåten da. Sant.
- 31 M: Ja.
- 32 L<sub>2</sub>: Og da utløste på en måte at de ikke fikk til. For eksempel stor c i stede for liten sant.
- 33 M: Ja, hadde den, det programmet jeg skrev på tavlen hadde jo også en sånn feil, en e som ble en c
- 34 L<sub>2</sub>: Ja, det er jo sant. Det er fort gjort. .. Og jeg tenker jo at det er, det er en veldig naturlig ting når du skal gå gjennom et digitalt verktøy i timen så vil du få utfordringer. Og dette var en stor klasse, så jeg vet ikke om du merket noen forskjell på å ha færre elever enn å ha flere.
- 35 M: Det var forskjeller mellom klassene ja.
- 36 L<sub>2</sub>: For to av klassene er veldig store og en litt liten.
- 37 M: Men så fikk jeg mulighet til å forbedre småting mellom da.
- 38 L<sub>2</sub>: Ja, ja det er jo veldig flott å kunne forbedre da. Men sånn personlig så synes jeg opplegget fungerte veldig godt. Og problemet er å få gå rundt å hjelpe alle da. Når du har såpass mange elever som i den klassen jeg hadde.
- 39 M: Mhm
- 40 L<sub>2</sub>: Og det ville vært lettere om du hadde halvparten eller noe
- 41 M: Ja
- 42 L<sub>2</sub>: eller noe sånt.
- 43 M: Ja, for når vi snakker om klassen, .. , kan du beskrive hvordan..., altså var det anderledes oppførsel eller
- 44 L<sub>2</sub>: Det var helt normal
- 45 M: Kan du beskrive klassen?
- 46 L<sub>2</sub>: Ja, altså sånn som de vanligvis er så har du en todeling du har en gruppe hvis du står framme og ser bak så har du den til høyre som jobber veldig jevnt og trutt og snakker matematikk og ikkje på en måte bråker. Også har du en gruppe til høyre og.. det er høyt nivå, hvertfall første prøven viste at det er høyt nivå i klassen, ehh, de kan prate mer og tulle mer, men det virker som de får det med seg. Hvertfall enn så lenge.
- 47 M: Ja
- 48 L<sub>2</sub>: Men jeg så jo at, hvertfall programmering er helt nytt for de, såå de satt no og jobbet så jeg, men det kan jo være de tar det fort og mister fort konsentrasjonen når skal gå rundt å hjelpe. Sant.. så blir det en del dødtid sant. Og da,.. jeg vet ikke om du merket det med volumet på den ene siden blir fort høyere.
- 49 M: Ja
- 50 L<sub>2</sub>: Ja.
- 51 M: Ikke unormalt
- 52 L<sub>2</sub>: Ikke unormalt i det hele. Men det er høyt nivå,.., første prøven viste at det er høyt nivå i klassen.
- 53 M: Ja, så høyt nivå, og det er to ulike grupper i klassen
- 54 L<sub>2</sub>: To veldig ulike grupper mhm
- 55 M: Men ikke noe nivåforskjell mellom gruppene eller?
- 56 L<sub>2</sub>: Nei. Det er det ikke.. hvertfall ikke etter første vurdering og sånn jeg har oppfattet det.
- 57 M: Det jeg endret på mellom timene var den lineære ligningsett oppgaven. Hvor tydelig jeg var på hva som skulle gjøres. Observerte du noe da elevene prøvde å løse denne oppgaven?
- 58 L<sub>2</sub>: Nei. Det tror jeg ikke jeg gjorde. Annet en at de skrev inn, at det ble noe feil når de skrev det inn,.. men vi kom vel ikke helt gjennom hva vi skulle gjøre. Eller forsvant det bare?
- 59 M: Timen ble ferdig, så jeg gikk gjennom slutten på tavlen.
- 60 L<sub>2</sub>: Jeg observerte i hvertfall ikke noe så. Annet enn at det, når de skulle skrive inn disse a b c d og e så ble det noen innskrivingsfeil og sånn. Men det er godt mulig.. de er jo ikke så vant med å omforme formler og tenke.. de tenker gjerne tall og ikke bokstaver.
- 61 M: Ja. Ikke vant med generelle ligninger, sånn som i dette tilfellet.
- 62 L<sub>2</sub>: Nei, det har ikke vi,.., vi har jobbet veldig lite med dette enda, selv om dette VIL ta en større plass.. etter hvert

- 63 M: Det kommer jo i hvertfall med algoritmisk tenking..
- 64 L<sub>2</sub>: Ja det vil komme med da og så i tillegg så virker det som at oppgavene skal gå mer på forståelse no sant, så.
- 65 M: Ja.
- 66 L<sub>2</sub>: I de tidligere årene har vært veldig mye lærestoff og veldig liten tid og du har på en måte måttet komme deg igjennom metodene sant, så du har ikke fått fordypet deg sant. Og det var jo et håp om at den nye læreplanen skulle være en forbedring å det, men da har de lagt programmering på, ..
- 67 M: Sånn jeg ser det gjør programmering at du må bruke de mer generelle måtene å løse ting på
- 68 L<sub>2</sub>: Ja, ja
- 69 M: At det tvinger deg til å måtte gjøre det.
- 70 L<sub>2</sub>: Ja, det er jo det som er positivt, men det er for deg som har hat det er det en naturlig ting å si, men for meg som ikke har hatt programmering er det en utfordring, enn så lenge
- 71 M: Og du føler språket er det som er utfordrende?
- 72 L<sub>2</sub>: I hvertfall språket, men det er jo kommandoene du skal gi.. det er jo litt som i CAS at gir du en feil kommando så står du der.
- 73 M: Ja.
- 74 L<sub>2</sub>: Som då er en utfordring, da i CAS også, når det stopper opp, og det er litt av det jeg syns med programmering også, at det er vanskelig å stå der og skulle undervise. Når jeg ikke selv vet hvordan jeg skal komme meg fra en linje til den neste.
- 75 M: Ja, husker jeg unngikk CAS da jeg var vikar, litt med vilje, hehe..
- 76 L<sub>2</sub>: Ja, hehe  
[Uinteressant prat om min privatundervisningserfaring ca. 15 sek]
- 77 L<sub>2</sub>: Og det er, i lt har du på en måte kunnet unngått de vanskeligste CAS-oppgavene, men så på eksamen har de plutselig hatt en sekseroppgave CAS-greie. Så jeg har også kjørt litt på siden, for elevene kan jo løse oppgaven, men kanskje ikke i CAS. Og der er jo jeg litt uenig i at om du kan løse den grafisk men ikke i CAS, .. sant også
- 78 M: Ja, fokuset er på verktøyet
- 79 L<sub>2</sub>: Ja.
- 80 M: For der, jeg har jo sett på eksempeloppgaven til eksamensoppgave i dette temaet.. programmering i Python. Jeg syns det var litt.. det krevde mer verktøyskompetanse en jeg tenkte først.
- 81 L<sub>2</sub>: Ja sant. Dette her er jo noe jeg ikke kan si noe om,.. jeg har jo reagert på noen av oppgavene, der du ikke får riktig selv om du har løst den, men du får ikke riktig fordi du ikke har løst den i CAS. Og da tenker jeg jo at det er rart. Og det kan godt hende at det blir sånn med programmering og, uten at jeg...
- 82 M: Ja, spørsmål,...., har dere gått gjennom digitale hjelpemidler før i denne klassen, ikke bare programmering.
- 83 L<sub>2</sub>: Ingen ting.
- 84 M: Ikke CAS ikke GeoGebra?
- 85 L<sub>2</sub>: Nei, ingen ting egentlig. Hadde noe denne uken.. Da begynte vi helt basic med CAS og GeoGebra på helt enkle ligninger.. og andregradsligninger. Og det tar tid. Jeg brukte.. på gjennomgangen så brukte jeg lang tid for du skal få med alle og du ja..
- 86 M: Ja, det så jo jeg også, det at probleme jo var språket, at elevene slet med det. Det med stor p i stede for liten p i print
- 87 L<sub>2</sub>: For eksempel. Og en annen ting som jeg har på en måte er at noen kommer fra ungdomsskolen og har hatt, har hatt noke GeoGebra og noen har hatt ingen ting. Sant. Så det og er vanskelig, så tror jeg at på en måte,.. i mitt hode så tenker jeg at de har hatt litt mer GeoGebra enn de har hatt.
- 88 M: Ja.
- 89 L<sub>2</sub>: På ungdomstrinnet. Men jeg så jo da når jeg gikk gjennom emm, løse ligninger i CAS og grafisk at de kunne lite.
- 90 M: Ja
- 91 L<sub>2</sub>: Og så i tillegg sant så prøver du å lære de å sette inn uttrykk sant og få litt godt utklipp og sånt
- 92 M: Du kan jo løse mange av disse tingene med programmering da, andregradsligning da for eksempel.

- 93 L<sub>2</sub>: Ja, det har jeg sjønt, sett et eksemple i Sinus-boken. Blandt annet
- 94 M: Så hva tenker du om å lære elevene i CAS og i programmering da?
- 95 L<sub>2</sub>: Nei, hehe, siden jeg syns programmering er en utfordring så tenker jeg at det er smør på flesk på en måte, å kunne det på, som du sikker sjønner så tenker jeg at det så lenge du har en metode for å lære det på så har du så begynner å få forståelsen på en måte
- 96 M: Ja
- 97 L<sub>2</sub>: kanskje fått den allerede, men jeg tenker flere verktøy jo bedre er det. Om du løser i CAS eller i programmering så kommer du frem til samme svaret. Men sånn algoritmisk så vil du måtte tenke anderledes i programmering da
- 98 L<sub>2</sub>: så altså, nei jeg vet faktisk ikke hva som er svaret på spørsmålet ditt altså, om det er nødvendig eller ikke. Når jeg ikke kan programmering så tenker jeg at det er unødvendig. Men hvis jeg hadde på en måte kommet mer inn i programmeringsspråket hadde jeg sett mer nytten av det.
- 99 M: Ja.. Da har jeg komt til slutten av det jeg ville spørre om.

[Intervju avsluttes]

---

## A.3 Intervju E<sub>1</sub>

- 
- 1 M: Det første jeg er interessert i å vite er hvor interessert du er i matematikkfaget.
- 2 E<sub>1</sub>: Mmm, det er jo et veldig viktig fag, så jeg er ganske motivert, og ... jeg tenker det kanskje det ... det viktigste faget for liksom grunnlag for videre utdanning. Så det er jo... så er det jo et avgangsfag også, så det er ganske så viktig.
- 3 M: Ja, så det er både karakterer og fremtid?
- 4 E<sub>1</sub>: Ja, begge deler.
- 5 M: Ja, vurderer du noe utdanning som krever matematikk da, eller er det bare at du ser at dette er nyttig?
- 6 E<sub>1</sub>: Ja, jeg vurderer ting innen ingeniørfaget, så kanskje noe med sånn, ææ, sånn bærekraftig utvikling og sånn.
- 7 M: Ja, har du noe tidligere erfaring med programmering?
- 8 E<sub>1</sub>: Ææ, stefaren min jobber som programmerer i IT-bransjen, men jeg har ikke lært så masse av ham.
- 9 M: Så du har ikke hatt noe i skolesammenheng tidligere?
- 10 E<sub>1</sub>: Mmm, nei, eller ikke på veldig lenge.
- 11 M: Hva husker du fra opplegget jeg hadde med dere?
- 12 E<sub>1</sub>: Jeg husker vi lærte hvordan vi programmerte inn formler ... eller vi liksom vi brukte programmering til å liksom løse oppgaver å sånn.
- 13 M: Husker du noe av det dere skrev i programmet? Kommandoer og sånt?
- 14 E<sub>1</sub>: eg husker ikke helt hva det var men, ææ ... jeg husker cirka hvordan det så ut da. Det var ord liksom, så brukte vi parenteser å sånn og innen for der skrev vi liksom hva oppgaven var og sånn da.
- 15 M: Ja, mmm, tror du programmering er et nyttig verktøy i matematikken?
- 16 E<sub>1</sub>: Ja det tror jeg. Og sånn generelt er det nyttig å kunne liksom.
- 17 M: Ja, har du noen formening om du kommer til å bruke programmering i matematikk, utenom det som blir undervist? Som et verktøy?
- 18 E<sub>1</sub>: Mmm, ææh. Det er jeg litt usikker på, jeg har ikke så masse kunnskaper om det, men det kan jo godt være.
- 19 M: Ja, akkurat som med GeoGebra eller kalkulator?
- 20 E<sub>1</sub>: Det tror jeg.
- 21 M: Ja. Tror du programmering kan gjøre noen oppgaver enklere?
- 22 E<sub>1</sub>: Kanskje oppgaver med større tall, for der får man liksom hjelp til å regne ut.
- 23 M: Mhm, var det noe du opplevde som vanskelig med programmering, fra denne timen?
- 24 E<sub>1</sub>: Æææ, ja, noe som var vanskelig det var jo at hvis man skriver en liten feil, for eksempel hvis man har en lang oppgave, så hvis man skriver en liten feil så blir jo hele svaret ødelagt, også ja, det kan være litt utfordrende hvis man har en liten skrivefeil.
- 25 M: Ja, og da var det vanskelig å finne hvor feilen var eller?
- 26 E<sub>1</sub>: Ja.
- 27 M: Nå skal jeg gi deg to oppgaver. Og jeg gjentar at dette ikke er en test, prøv å forklar muntlig hva du tenker, og du kan også skrive på oppgavearket. [Gir eleven oppgaveark og en penn]
- 28 E<sub>1</sub>: Jepp.
- 29 M: Også får du hjelp fra meg og jeg kommer til å være støtte her, også kan du lese oppgaven først. Du kan lese inni deg. [Elev begynner å lese oppgave 1]  
[Ca. 12 sek med lesing]
- 30 E<sub>1</sub>: Ææ, lage et program?
- 31 M: Du skal lage et program som skal finne, du får oppgitt to linjer, og så skal du lage et program som finner punktet der disse to rette linjene krysser.. I x-y koordinater.
- 32 E<sub>1</sub>: Ja, mm.



- 33 M: Hva tenker du først om det?
- 34  $E_1$ : Æ, ja, æ, jeg er litt usikker på hva det menes med å lage et program.
- 35 M: Du skal få en datamaskin til å løse det for deg.
- 36  $E_1$ : Åja, så ved hjelp av programmering da liksom?
- 37 M: Ja, ved hjelp av programmering.
- 38  $E_1$ : Åja ja. Ja det er vel noen formler å sånn man må bruke da. Litt usikker. Jeg tenker umiddelbart på det der GeoGebra, så bruker man jo sånn funksjon da, der man velger begge linjer så kommer skjæringspunktet fram. Men jeg vet ikke helt hvordan jeg skal programmere for å finne det.
- 39 M: Mhm, husker du den oppgaven jeg gjennomførte i timen helt på slutten? Der jeg viste at vi hadde to lineære ligningsett.
- 40  $E_1$ : Ja, det husker jeg.
- 41 M: Kanskje vi kan bruke samme metode på dette problemet?
- 42  $E_1$ : Ja kanskje det.
- 43 M: Husker du hva jeg gjorde?
- 44  $E_1$ : Litt usikker tror jeg.
- 45 M: Det jeg gjorde, var at jeg satte opp slik at vi fikk at de er like ...
- 46  $E_1$ : Ja, no husker jeg. Du satte du ikke inn, begge satt liksom sammen de to ... også regnet ut.
- 47 M: Mhm.
- 48  $E_1$ : Husker ikke helt utregningen, var litt vanskelig husker jeg, men ja.
- 49 M: Ja. Kjenner du igjen disse formlene her da? [Peker på eksempelet med to lineære linjer]
- 50  $E_1$ : Ja de husker jeg.
- 51 M: Hva står de ulike tallene for i denne ligningen? [Peker på en av ligningene]
- 52  $E_1$ : Æhm, ... litt usikker. Det er vel, det er vel konstantleddet tror jeg. Det er hvor masse det stiger med hver gang. [Elev peker på de ulike delene av ligningen]
- 53 M: Så hvis du ikke skulle brukt tall da ... bare fikk fortalt at du får to linjer og du skal lage program for å løse det.
- 54  $E_1$ : Hvordan jeg skal få til å? ... jeg er litt usikker.
- 55 M: I hvertfall en måte å løse det på er jo å sette opp, i stede for å bruke tall her da ...
- 56  $E_1$ : Ja.
- 57 M: Tre og ti, som er tilfellet i denne linjen så kan du si at  $y = ax + b$ .
- 58  $E_1$ : Åja sånn ja.
- 59 M: Sant.
- 60  $E_1$ : Ja.
- 61 M: Og då er disse to kjente.
- 62  $E_1$ : Og så bruke denne formelen til å finne.
- 63 M: Dette er jo, y og x endrer seg langs denne linjen.
- 64  $E_1$ : Ja.
- 65 M: Når x endrer seg endrer y seg også. Og omvendt. Det vi er ute etter er når y-en på denne linjen er lik y-en til denne linjen og x-en på denne linjen er lik x-en på denne linjen.
- 66  $E_1$ : Ja.
- 67 M: Så da må vi også beskrive denne linjen uten tall. Vi bruker  $cx + d$ .
- 68  $E_1$ : Ja, så det er de to .. de to linjene.
- 69 M: Ja.
- 70  $E_1$ : Også skal vi finne krysningspunktet ved å bruke de?
- 71 M: Ja. Når denne y-en er ...

- 72 E<sub>1</sub>: Ja det husker jeg fra timen, det var jo at du ganget inn med de bokstavuttrykkene der, men det husker ikke helt hvordan du gjorde det.
- 73 M: Du kan jo prøve å vise hva du husker på arket.
- 74 E<sub>1</sub>: Ja, jeg kan jo prøve ...[elev skriver på arket]. ... Skal man gange denne med denne da, eller skal man liksom gange denne både med begge to da? Når man ganger de sånn liksom.. ganger de inn.
- 75 M: Dette ser litt mer inviklet ut en det jeg tenkte... hva vis du setter denne y-verdien lik denne da?
- 76 E<sub>1</sub>: Ja.
- 77 M: For det skjer jo i dette punktet.
- 78 E<sub>1</sub>: Mhm.
- 79 M: At y-en til denne linjen er lik y-en til denne linjen. Og da må jo denne funksjonen være lik denne funksjonen.
- 80 E<sub>1</sub>: Åja, sånn ja. Er litt usikker på hvordan jeg skal gå frem videre. ... Tror du på hjelpe litt.
- 81 M: Ja, du setter denne lik denne [skriver ned  $ax + b = cx + d$  i notatboken min].
- 82 E<sub>1</sub>: Åja, sånn,  $ax$  ... pluss  $b$  er lik  $cx$  pluss  $d$  ... sånn.
- 83 M: Ja.
- 84 E<sub>1</sub>: Ja. Også ... kan man for eksempel ... kan man dele på  $x$ , er det lov å dele på  $x$  her, på begge sider?
- 85 M: Hva skjer da?
- 86 E<sub>1</sub>: Man kommer nærmere, liksom finne det andre.
- 87 M: Mhm, det du begynner med kan jo være bytte rundt på ting før du ganger eller deler.
- 88 E<sub>1</sub>: Flytte over da?
- 89 M: Ja, flytte-bytte eller hva dere kaller det ...
- 90 E<sub>1</sub>: Ja, så hvis man skal flytte over ... æhh ... hva skal vi flytte over liksom?
- 91 M: Altså, det vi er ute etter er å få ukjente på en side av likhetstegnet og kjente på den andre siden.
- 92 E<sub>1</sub>: Ja. Da må jeg dele på  $x$ . Må jeg liksom flytte over... Hvis jeg skal flytte  $x$  over, må jeg dele på  $x$  her då liksom? Hvis jeg fjerner det her liksom.
- 93 M: Hvis du fjerner  $x$  her må du trekke det fra på denne siden.
- 94 E<sub>1</sub>: Trekke fra. Ja, oki. Tenkte for at det var  $ax$ ,  $a$  ganger  $x$ . Så hvis jeg flytter over  $x$  da... Bli det pluss der ... Minus  $x$  sånn?
- 95 M: Nei, ikke helt sånn.
- 96 E<sub>1</sub>: Ikke helt.
- 97 M: Fordi du må flytte hele denne.
- 98 E<sub>1</sub>: Åja, flytte hele. Hele  $ax$ .
- 99 M: Ja.
- 100 E<sub>1</sub>: ... Bare sånn liksom?
- 101 M: Ja. Helt rett. Så gjør du det samme med  $d$ -en.
- 102 E<sub>1</sub>: Ja sånn ha, oki. Sånn?
- 103 M: Helt rett.
- 104 E<sub>1</sub>: Så skal jeg ... hmm. Må jeg prøve å flytte over  $e$ -en og  $a$ -en. Hvordan gjør jeg det?
- 105 M: Husker du dette [viser eleven at  $ex - ax = x(e - a)$ ]. At du kan gjøre dette? Faktorisere.
- 106 E<sub>1</sub>: Skal jeg faktorisere slik at  $x(e - a)$
- 107 M: Ja.
- 108 E<sub>1</sub>: Ja.
- 109 M: [Venter ca. 10 sek] Så kan du dele dette vekk.
- 110 E<sub>1</sub>: Dele dette vekk? Så må jeg gjøre det samme på denne siden?

- 111 M: Ja.
- 112 E<sub>1</sub>: Ja, det var det jeg tenkte. Å så da blir det ... Eh, delt på e minus a her borte.
- 113 M: Ja.
- 114 E<sub>1</sub>: Så da stryker jeg ut her.
- 115 M: Da har du fått at x er lik.
- 116 E<sub>1</sub>: Da har jeg fått, .. ja sånn ja. [Skriver i ca. 10 sek] Sånn?
- 117 M: Ja.
- 118 E<sub>1</sub>: Det minus det delt på e minus a er lik x.
- 119 M: Ja.
- 120 E<sub>1</sub>: Ja.
- 121 M: Dette kan du kanskje også se at stemmer med disse linjene.
- 122 E<sub>1</sub>: Ja, bruke formelen til å løse denne?
- 123 M: Ja.
- 124 E<sub>1</sub>: Ja. Med den formelen her da?
- 125 M: Ja, så må du vite hva de ulike står for.
- 126 E<sub>1</sub>: [ca. 8 sek der eleven skriver/tenker] Å ja, jeg har gjort en liten feil her, jeg har begynt å skrive e i stedet for c her.. Det var litt forvirrende, men der sånn. Då då blir det b-en blir den så det blir minus ti minus .. jeg skal ta med minusen her også sant?
- 127 M: Ja.
- 128 E<sub>1</sub>: minus d, og d er 7.5, delt på, ehm c som er minus 0.5 også minus a som er 3, 3x, eller 3?
- 129 M: Ja. Tenkte du kunne gjøre utregningen her [Har tatt frem Python Shell på pc-en]. For vi får jo
- 130 E<sub>1</sub>: minus 17.5 delt på minus 3.5, ... då er x lik 5.
- 131 M: Ja, og det stemmer jo også med bildet, ikke sant? Hvis vi teller her.
- 132 E<sub>1</sub>: Mhm, ja det stemmer.
- 133 M: Også må vi finne y-en også.
- 134 E<sub>1</sub>: Ehm, [stille i ca. 10 sek] det står jo at y er lik, jeg kan jo bruke flere formler for å finne y ikke sant? Jeg kan bruke hvilken som helst av disse formlene?
- 135 M: De to øverste ja, men ikke disse.
- 136 E<sub>1</sub>: Ja, ok. y er lik ... 3 x ... æh, også pluss b, blir det minus da, siden det står minus 10?
- 137 M: Ja.
- 138 E<sub>1</sub>: 3 ganger 5 minus 10, ..., det blir jo fem, men det så jeg jo allerede. Men her brukte jeg jo formler og sånn.
- 139 M: Så hvis du skal beskrive et program, hvis du skal lage et program som finner dette punktet. Du kan skrive programmet inn på pc-en om du vil.
- 140 E<sub>1</sub>: Da skal jeg bruke formelen og sånn?
- 141 M: Ja.  
[eleven jobber på pc-en i ca. 1 min]
- 142 E<sub>1</sub>: Æh, så skal jeg bare skrive inn x og y også der? For x og y er jo det vi skal finne ut. Så er dette greit sånn?
- 143 M: Ja, men Python bruker punktum i stedet for komma så
- 144 E<sub>1</sub>: Ja. Ja, det gjør jo GeoGebra også tror jeg.
- 145 M: Ja, komma blir brukt til å separere ulike tall.
- 146 E<sub>1</sub>: Ja, talllinje.
- 147 M: Ja, og hva er neste steget for å få et program til å finne punktet?
- 148 E<sub>1</sub>: Er ikke det å skrive inn formlene?

- 149 M: Ja, prøv det.
- 150 E<sub>1</sub>: Ja, så da går jeg ned hit.
- 151 M: Husk at du må ha gangetegn mellom to tall
- 152 E<sub>1</sub>: Ja, det var den stjernen.
- 153 M: Ja.
- 154 E<sub>1</sub>: Stjernen, det var der. Ja. Også, også må jeg ha gangetegn. Sånn.
- 155 M: Tror du dataprogrammet skjønner dette?
- 156 E<sub>1</sub>: Ja, tror det. Ikke?
- 157 M: Det du ser er jo at x-en er ukjent for programmet.
- 158 E<sub>1</sub>: Mhm.
- 159 M: Så når du skriver at y er lik noe, så må det til høyre være kjent.
- 160 E<sub>1</sub>: Ja. Ja, må jeg skrive inn formelen der?
- 161 M: Ja.
- 162 E<sub>1</sub>: Ja, oki. Skal jeg la dette stå her og skrive en ny en inn her?
- 163 M: Ja.
- 164 E<sub>1</sub>: Ok. Så delt på det, bare sånn?
- 165 M: Ja. Hvordan får du programmet til å vise hva som er funnet?
- 166 E<sub>1</sub>: Var ikke det bare man skulle trykke på run?
- 167 M: Prøv det.  
[Programmet kjører]
- 168 M: Oj, en feilmelding.
- 169 E<sub>1</sub>: Ja, skal jeg inn og sjekke da. Det står name e is not defined. Og da på grunn av jeg skrev jo e i stedet for c i formelen min. Så bare endre den til c.  
[Programmet kjører på nytt]
- 170 M: Hvorfor viser ikke programmet noe svar?
- 171 E<sub>1</sub>: Det vet jeg ikke.
- 172 M: Husker du print kommandoen.
- 173 E<sub>1</sub>: Ja print ja. ... Hvordan var det igjen?
- 174 M: print parentes x gjør at x vises. Også det samme for y.
- 175 E<sub>1</sub>: Print parentes x .. også print parentes y. Sånn?
- 176 M: Ja. Kjør programmet.
- 177 E<sub>1</sub>: Ja da skal jeg bare kjøre det da?
- 178 M: Ja.  
[Programmet kjører]
- 179 E<sub>1</sub>:Der.
- 180 M: Ja.
- 181 E<sub>1</sub>: Da har vi funnet de.
- 182 M: Ja. Da kan vi gå gjennom siste oppgaven. Denne oppgaven handler om temaet dere jobber med idag i timen. Så jeg tror ikke dere har jobbet med dette enda. Hakke vanskeligere enn forrige oppgave kanskje.
- 183 E<sub>1</sub>: Mhm. [Elev leser oppgave] Ja, det her var litt mer vanskelig.
- 184 M: Har dere gått gjennom dette alt?
- 185 E<sub>1</sub>: Nei.
- 186 M: Men jeg tror fortsatt at du skal være i stand til å klare oppgaven. Bare spør når du ikke forstår ting.

- 187 E<sub>1</sub>: Ja. Vi gikk litt gjennom det.
- 188 M: Du vet hva nullpunkt er?
- 189 E<sub>1</sub>: Det her er de to nullpunktene [peker på nullpunktene].
- 190 M: Ja. Og de to finner du ..
- 191 E<sub>1</sub>: Med de to formlene.
- 192 M: Ja.
- 193 E<sub>1</sub>: Mhm.
- 194 M: Du skal jo skrive ett program som forteller deg x-verdiene til nullpunktene.
- 195 E<sub>1</sub>: Mhm. Ja ... mmh ... når vi har jobbet med disse here, det kan være både pluss og minus så først må man på en måte finne verdier også må man sette de inn. Det er sånn, det er det jeg husker.
- 196 M: Ja. Jeg har jo noen funksjoner med som programmet skal løse. Sånn at du kan se disse. Disse funksjonene ser du hva er?
- 197 E<sub>1</sub>: Dette er sånne andregradsformler ikke sant?
- 198 M: Ja. Og her oppe ser du også en sånn, men uten tall.
- 199 E<sub>1</sub>: Ja, det er vel, b i andre må jo være den.
- 200 M: Ikke helt, hvis du ser den ..
- 201 E<sub>1</sub>: Oja, der ja. Det må jo være den, for den er i andre.
- 202 M: Ja.
- 203 E<sub>1</sub>: Så da bli a 1, b-en 3
- 204 M: Ja
- 205 E<sub>1</sub>: Også c-en 1
- 206 M: Ja, også skal du skrive ett program.
- 207 E<sub>1</sub>: Må jeg ikke sånn som forrige gang skrive inn verdiene.
- 208 M: Ja.
- 209 E<sub>1</sub>: a var lik 1, b lik 3. Nå tar jeg den første da. Og c er lik 1. Sånn. Også, neste som var må jo skrive inn funksjonen da, æhh, formelen, disse to. Ja.
- 210 M: Siden vi har to ulike x-verdier kan vi kalle den ene x1 og den andre x2. Og så kan kvadratroten oppgis på denne måten (skrive på ark  $**(0.5)$ )
- 211 E<sub>1</sub>: Ja... Skal jeg skrive inn den nå, kvadratroten?
- 212 M: Ja, om det løser oppgaven. Bare lurte litt på hvorfor du skrev inn minus 3 der?
- 213 E<sub>1</sub>: Åja ja, trenger ikke sette inn verdiene forskjellige der. Sånn ja. Også, hvordan var det med kvadratroten.
- 214 M: Det som er under kvadratroten i parentes også stjerne stjerne 0.5.
- 215 E<sub>1</sub>: Ja. Også
- 216 M: Husk gange tegn mellom
- 217 E<sub>1</sub>: Ja. Sånn, også parentes. Jepp. Skal jeg skrive inn den andre no då?
- 218 M: Ja.
- 219 E<sub>1</sub>: [Elev tenker i ca. 20 sek]Blir ikke det riktig da?
- 220 M: Du kan jo prøve å kjøre programmet.
- 221 E<sub>1</sub>: Men da må vi printe også. x1 ... x2. Æh ja. Var det nullpunktet?
- 222 M: Det var nullpunktet ja.
- 223 E<sub>1</sub>: Står det ikke at nullpunktet er litt sånn ca. -2.8 og 2.8?
- 224 M: Ja. Kan du prøve å løse disse to også? [Peker på de to andre funksjonene]
- 225 E<sub>1</sub>: Jepp, ehmm. Ja her er det samme, a er en, også er det to og to. Også er det run module.  
[Intervjuet avsluttes etter dette]
-

## A.4 Intervju E<sub>2</sub>

- 
- 1 M: Det første jeg lurer på er hvor motivert du er i matematikkfaget?
- 2 E<sub>2</sub>: Mmm, jeg har jo lyst å få den beste karakteren jeg klarer å få.
- 3 M: Ja, det er motivasjonen da?
- 4 E<sub>2</sub>: Ehh, det er jo få god utdanning og matte er jo viktig i veldig mange utdanninger.
- 5 M: Har du tenkt hva du vil gjøre?
- 6 E<sub>2</sub>: Jeg har jo lyst å bli psykolog.
- 7 M: Har du noen tidligere erfaring med programmering.
- 8 E<sub>2</sub>: Emm, jeg programerte i niende og tiende.
- 9 M: Ja, hvilke program brukte dere da, husker du det?
- 10 E<sub>2</sub>: Da brukte vi Python.
- 11 M: Python ja, ikke scratch?
- 12 E<sub>2</sub>: Jo, vi brukte scratch litt også ja, og micro:bit.
- 13 M: Hva husker du fra det opplegget jeg underviste i?
- 14 E<sub>2</sub>: Ehm, jeg husker at vi lærte om det mest 'basic'e som, ehm print, ehm så var det en variabler og hvordan man bruker ulike tegn, som pluss og minus og det var et prosenttegn.
- 15 M: Ja. Ehm, tror du programmering er et nyttig verktøy i matematikk?
- 16 E<sub>2</sub>: Ja, ja.
- 17 M: Har du noen tanker om det?
- 18 E<sub>2</sub>: Hehe, ehm, æh det vet jeg ikke helt. Jeg føler det er mer sånn fremtidig, noe vi kan bruke mer i fremtiden.
- 19 M: Ja. Tror du du vil bruke programmering som et verktøy i matematikk?
- 20 E<sub>2</sub>: Ehm, hvis oppgaven spør om det spesifikt.
- 21 M: Ja. Er det noe du føler blir enklere når du programmerer, oppgaver eller noe sånt?
- 22 E<sub>2</sub>: Ehm, ja det kan vel det kanskje. Eller liksom sette det inn i mer realistisk eller på en måte andre typer settinger kan man si.
- 23 M: Ja. Er det noe du synes er vanskelig med programmering.
- 24 E<sub>2</sub>: Det er vel å huske alle de ulike kommandoene og hva ting betyr egentlig
- 25 M: Ja. Da har vi komnt til oppgavene. Jeg kan gjenta at dette ikke er en test eller prøve. Det er også litt nye ting. Om du kan snakke om hva du tenker eller skrive på ark. [Elev leser på ark]
- 26 E<sub>2</sub>: Ehm okei. Ehh.
- 27 M: Kan bare si at figuren bare er ett eksempel.
- 28 E<sub>2</sub>: Jajaja! To linjer skal finne hverandre. Der de krysser. Altså to punkter då må de jo være sånn, ..., eller to linjer. Er de to linjer som ikke stopper på en måte?
- 29 M: Mhm.
- 30 E<sub>2</sub>: Da må de jo. Ehm då må de jo være to formler ikke sant, på en måte.
- 31 M: Hvordan tror du formlene ser ut?
- 32 E<sub>2</sub>: Ehm. Eh ja, for dette er bare et eksempel.
- 33 M: Ja.
- 34 E<sub>2</sub>: Jeg er litt usikker men.
- 35 M: Bare prøv. Du kan skrive ned det du tenker også. ... Hva er du usikker på?
- 36 E<sub>2</sub>: Ehm. Jeg tenkte to linjer. Liksom at man vet disse to forskjellige linjene. Å da kan man jo si at det blir liksom a x minus b.
- 37 M: Ja. Har du vært borti denne formelen før?

- 38 E<sub>2</sub>: Ehm, jeg har vel det, men jeg husker ikke.
- 39 M: Om du skrev  $ax$  pluss  $b$  da?
- 40 E<sub>2</sub>: Ja, jajaja! Ja oki. Det at den har negativ verdi betyr jo at den begynner under. [Indikerer under  $x$ -aksen i eksempelet]
- 41 M: Mhm.
- 42 E<sub>2</sub>: Ja. Så er det enten pluss eller minus.
- 43 M: Og du har to linjer da.
- 44 E<sub>2</sub>: Så blir det  $ax$  pluss eller minus  $b$  igjen. Ehm ... Så skal vi regne ut punktet der de møtes.
- 45 M: Ja.
- 46 E<sub>2</sub>: Ehm. ... Jeg må tenke litt.
- 47 M: Hva er det du tenker på nå da?
- 48 E<sub>2</sub>: Ehm, jeg vet ikke helt. Jeg prøver å finne på hva det kan være på en måte.
- 49 M: For må  $a$  ene og  $b$  ene være like?
- 50 E<sub>2</sub>: Hem nei. Nei. Blir det sånn  $a_1$  og  $b_2$  da også.. nei  $b_1$  mener jeg også blir det  $a_2$  og  $b_2$ .
- 51 M: Ja. Det kan du for eksempel skrive. ... [tenker i ca. 15 sek]. Hva er det de linjene beskriver. Hva er det  $a$ -ene og  $b$ -ene beskriver?
- 52 E<sub>2</sub>:  $a$  beskriver jo hvor mye den går opp
- 53 M: Stigningstallet?
- 54 E<sub>2</sub>: Stigningstallet ja. Også  $b$  hvor linjen begynner, eller hvor den begynner på  $y$  liksom.  
[Elev tenker i ca. 30 sek]
- 55 E<sub>2</sub>: Åhh, då videre på en måte sånn, jeg føler det skal bli sånn  $y_1$  også  $y_2$ . Eller  $y_1$  er lik  $y_2$  når  $y_1$  er lik  $y_2$ .
- 56 M: Ok. Hva mener du med det?
- 57 E<sub>2</sub>: Jeg vet ikke.
- 58 M: Hva er  $y_1$  da?
- 59 E<sub>2</sub>: Ja, at  $y_1$  er den ene formelen og  $y_2$  den andre formelen også når de har lik verdi. ... Blir det da  $\text{print}(y_1)$ ?
- 60 M: Hvordan finner programmet ditt  $y_1$  da?
- 61 E<sub>2</sub>: Ehh, hehe jeg holdt på å si jeg gir opp jeg men!
- 62 M: Husk at jeg prøver å få deg til å si mest mulig om hva du kan.
- 63 E<sub>2</sub>: Jeg vet ikke helt hva jeg gjør, kan du gi men noe hint liksom?
- 64 M: For du er jo på rett vei..
- 65 E<sub>2</sub>: Asså jeg føler det noe sånn when eller if  $y_2$  så leser den liksom opp den verdien. Men hvordan programmet vet hvilke verdi det er.
- 66 M: Husker du det siste jeg gjorde på tavlen. Oppgaven med det lineære ligningsettet.
- 67 M: Det jeg gjorde var jo flytte bytte med formlene slik at vi fikk kjente på ene siden og ukjente på andre. Beskrive ukjente ved hjelp av kjente.
- 68 E<sub>2</sub>: Mhhh, ja. Åja så istede for  $a_1$  og  $b_1$  så blir det  $a$   $b$   $c$  og  $d$ ?
- 69 M: Jeg synes det ser helt greit ut. Vi bruker ulike måter å skrive på og jeg likte liksom  $a_1$  og  $d$ . Dette er en grei notasjon i matematikk. Men du velger selv hva du vil.
- 70 E<sub>2</sub>: Ja. Okei. Mhhm så det blir på en måte  $a_1x$  pluss minus  $b_1$  og  $a_2x$  pluss minus  $b_2$ .
- 71 M: Ja, du trenger ikke skrive pluss minus. For fortegnet kan endre seg,  $b$  kan liksom være positiv eller negativ.
- 72 E<sub>2</sub>: Ja. Oki, ... Så blir det som en en linje da eller?
- 73 M: Hva mener du?
- 74 E<sub>2</sub>: Som  $y$  er lik .. eller  $y$  i annen eller ... mmmm.

- 75 M: For vi snakket om flytte bytte ..
- 76 E<sub>2</sub>: Ja oki, det blir sånn  $a_1x$  minus  $a_2x$  også er det lik  $b_2$  minus  $b_1$ .
- 77 M: Husker du faktorisering? Altså at vi kan gjøre dette [viser på ark at  $ax + bx = x(a + b)$ ]
- 78 E<sub>2</sub>: Så det blir liksom  $x a_1$  minus  $a_2$  ?
- 79 M: Ja.
- 80 E<sub>2</sub>: Ja [Elev skriver på ark ca. 12 sek]. Jeg tror jeg husker at du gjorde det samme.
- 81 M: Ja. Hva tenker du du skal gjøre videre?
- 82 E<sub>2</sub>: Jeg vet ikke.
- 83 M: Hva er målet med det du gjør nå?
- 84 E<sub>2</sub>: Ehhm at programmet skulle finne på en måte hvilke verdi de to linjene har når de krysser hverandre.
- 85 M: Rett, .. men nå tenkte jeg på akkurat dette [peker på arket] Altså at du skal finne verdi for  $x$
- 86 E<sub>2</sub>: Jajaja. Ehm ... Så da må jeg flytte denne på denne siden.
- 87 M: Ja, hvordan husker du?
- 88 E<sub>2</sub>: Er det ikke delt?
- 89 M: Ja.
- 90 E<sub>2</sub>: Mhh.
- 91 M: Nå har du jo alt du trenger for å skrive programmet, så kanskje du bare skal begynne. Her har du en pc. [Tar frem pc]. Hvordan vil du starte på programmet?
- 92 E<sub>2</sub>: Ehhh, hehe, no kom jo de variablene inn. Gjorde de ikke?
- 93 M: Hvilke variabler?
- 94 E<sub>2</sub>: Ehh, nei nå husker jeg ikke mer ... hvis jeg skriver .. nei det går ikke.
- 95 M: Hva er det som ikke går?
- 96 E<sub>2</sub>: Nei jeg tente at hvis jeg skriver  $ax$  pluss  $b$ .
- 97 M: Hvorfor går ikke det?
- 98 E<sub>2</sub>: Nei fordi det skjønner jo ikke programmet.
- 99 M: Ja. Helt rett. Hvordan kan du få programmet til å skjønne?
- 100 E<sub>2</sub>: Ja, ja, jeg kan ja, jeg kan skrive  $a$  er lik også kan liksom den personen som holdt på med programmet skrive inn sin egen verdi liksom.
- 101 M: Då kan jeg si at en av kommandoene vi brukte var [jeg skriver ned  $a = \text{eval}(\text{input}("a = "))$ ].
- 102 E<sub>2</sub>: Ja. Jo jeg skjønner input.
- 103 M: Ja den ber om input.
- 104 E<sub>2</sub>: Ja, men eval, når jeg har pleid å skrive dette så bruker jeg int eller str.
- 105 M: Ja, du kan bruke int str eller float også
- 106 E<sub>2</sub>: Ahh, jeg har ikke hørt om float.
- 107 M: Ja, og det er derfor du bør bruke eval, fordi inputen er tekst-streng, men når vi vil ha tall så kan du bruke desimaltall eller heltall. Int vil da fjerne desimalene, mens eval vil se om det er et heltall eller desimaltall og velge for deg. Eval står for evaluer liksom, den evaluerer tekst og oversetter til tall.
- 108 E<sub>2</sub>: Ahh, oki. Oki jaja.
- 109 M: Men om du vet at det kun er heltall kan du bruke int.
- 110 E<sub>2</sub>: Nå skriver jeg det vi spurte den som skal skrive ned noe .. og da bare kopierer jeg hele denne også [eleven kopierer  $a = \text{eval}(\text{input}("a = "))$  slik at det står seks ganger under hverandre], da tror jeg vi trenger seks variabler.
- 111 M: Hvorfor tror du du trenger seks slike?
- 112 E<sub>2</sub>: Vel, det var jo på den første linjen, da trenkte vi  $a$  og  $b$  der linjen startet på  $y$ -aksen, også  $y$ -en blir c.. trenger jeg å legge inn  $x$  og  $y$ , det er svaret jeg skal finne liksom? Det er svaret, det det skal bli på en måte.



- 113 M: Ja, så du trenger altså ikke at brukeren blir bedt om disse da?
- 114 E<sub>2</sub>: [Elev tenker en stund]. Oki, ja, fordi vi ikke har noe y. Oki... Så blir det mer som ... [tenkepause] ... jeg er usikker på om det blir denne jeg skal skrive ned eller denne.
- 115 M: Den nederste.
- 116 E<sub>2</sub>: [Tenkepause] Tror [mumling] a [Tenkepause/elev skriver] Jeg tenkte på å skrive et punkt.
- 117 M: Det du kanskje husker når vi gikk gjennom i timen fant vi x først og satte dette inn i formelen for y.
- 118 E<sub>2</sub>: Ja, så jeg kunne ta y er lik så blir det a minus b?
- 119 M: Differansen mellom a og b?
- 120 E<sub>2</sub>: Jeg tenker på at det at jeg kanskje må tenke på de øverste først før jeg tenker på de nederste [refererer til utregningen der eleven regnet fram  $x =$  på arket]
- 121 M: Men de ulike verdiene endres ikke, så er det ikke lettere å velge der x beskrives kun ved hjelp kjente verdier?
- 122 E<sub>2</sub>: Jaja, men kan jeg ikke gjøre slik at y er lik b og z for eksempel er lik em c minus, Neinei åja jaa..
- 123 M: Du tenkte å separerer formelen mer?
- 124 E<sub>2</sub>: Jeg tenkte liksom at jeg skulle finne at disse ble liksom en variabel til sammen der så disse ble en variabel så jeg kunne skrive at de to der.
- 125 M: Ja, men ikke bruk samme navn ... y bruker du alt. Men du kan også bare skrive inn i samme regnestykke å bruke parantes
- 126 E<sub>2</sub>: Men då kan jeg ikke bruke a b vel, då må jeg ta de som kommer etter hverandre eller spiller det ikke noen rolle egentlig?
- 127 M: Det spiller ikke noen rolle, bare du ikke bruker samme navn på ulike verdier.  
[Elev skriver inn på PC, ca 2 min.]
- 128 E<sub>2</sub>: ...Også bare print svar, men må vi her må vi skrive? Hvordan skriver vi print? Jeg tenkte siden det er tall på en måte. Blir det print eval?
- 129 M: Nei det trenger du ikke. Siden du ikke blander str int eller flyt går det greit. ... Tror du kan prøve å kjøre programmet.
- 130 E<sub>2</sub>: Oki, nå må vi oppgi verdiene.
- 131 M: Kan bare bruke fra eksempelet.
- 132 E<sub>2</sub>: Em, så a2 blir minus null.
- 133 M: Punktum ikke komma. Og minus ti, ikke ti.
- 134 E<sub>2</sub>: Ja, ja så klart.
- 135 M: Og nå har du kun funnet x-verdien.
- 136 E<sub>2</sub>: Mmm jeg har jo delt de to på hverandre for å finne x. mmm. Siden her blir det liksom er lik y. Kan jeg liksom sette y på siden og skrive minus y.
- 137 M: Hva tenker du på?
- 138 E<sub>2</sub>: Nå tenkte jeg bare... Jeg tenkte bare på det jeg holdt på med i timen nå. [ca. 20 sek tenking]
- 139 M: Hva er det du tenker på nå?
- 140 E<sub>2</sub>: Jeg tenker på hva om jeg flytter y-en inn her hva det egentlig gjør. Emm på hvordan jeg kunne finne y verdien her... Altså nå prøver jeg bare å finne y-verdien med i stykket på en måte.
- 141 M: Så du tenker å bruke dette. Kan jeg si at x ikke lengre er ukjent for programmet etter dette. ... Du skal bare finne y verdien. ... Så hvis du fikk oppgitt denne linjen her liksom, dette eksempelet, og viste at x er lik 5, og du ble bedt om å fortelle hva y er lik.
- 142 E<sub>2</sub>: Er det minus ti eller
- 143 M: Hvordan fant du det?
- 144 E<sub>2</sub>: Nei jeg bare vet at møter på y-aksen.
- 145 M: Så hvis du fikk en oppgave, denne linjen, og du vet at denne formelen, ..., og at x er lik fem hvordan ville du funnet y?
- 146 E<sub>2</sub>: At x er lik fem?

- 147 M: Ja.
- 148 E<sub>2</sub>: Altså da tar jeg fem inn i stykket så blir det tre ganget fem femten minus ti, så da blir det fem.
- 149 M: Ja.
- 150 E<sub>2</sub>: Ja. Aaa, oki, da blir det jo, da tar jeg jo svaret jeg får her.. åja oki.
- 151 M: Ja. Så nå kan du prøve å kjøre programmet på nytt.
- 152 E<sub>2</sub>: Då tar jeg, y er lik emm, er lik emm, ... a e emmm, blir det gange, emmm, gange svaret e plus emm b... Også print... Print x, emm er det riktig skrevet.
- 153 M: Jeg er usikker på om komma funker [elev skrev print(x, y)], men prøv det så finner vi ut.
- 154 E<sub>2</sub>: Ja oki.
- 155 M: Jeg har aldri prøvd det før, men vi kan jo bare teste ut.
- 156 E<sub>2</sub>: Så minus ti denne gangen ... ja.
- 157 M: Ja der visten den svaret, som likt det vi ser.
- 158 E<sub>2</sub>: Ja.
- 159 M: Nå går vi over på oppgave to, dette jobbet dere med i denne timen. Så les gjennom oppgaven og se ... [Elev leser oppgaven]. Du har kanskje sett slike funksjoner i dag.
- 160 E<sub>2</sub>: Jojo, her har vi jo a, b og c, og så putter vi inn de ulike verdiene her. Men er det liksom viktig liksom at det står liksom x opphøyd i annen liksom.
- 161 M: Ja, om det ikke er x i annen så er det ikke en annengradsfunksjon. Da funker ikke disse to formlene [peker på løsningene for abc-formelen].
- 162 E<sub>2</sub>: Ehhm, kan jeg skrive sånn i annen i programmet?
- 163 M: Ja, da bruker du to stjerner og ett to-tall.
- 164 E<sub>2</sub>: Ja, oki.
- 165 M: Stjernene betyr at du opphøyer i. Om du tar kvadratrot kan du skrive opphøyd i 0.5. Er en måte å skrive kvadratrot.
- 166 E<sub>2</sub>: Altså først hadde jeg jo spurt om, om at den som skal løse stykket sitt skriver ned verdiene sine. Altså at jeg tar a er lik også gjør jeg det samme som i sted. [Elev skriver på pc]. Da blir det jo skriv tallet a.
- 167 M: Ja. [Elev skriver i ca. 30 sek]
- 168 E<sub>2</sub>: Så langt har jeg komt der.
- 169 M: Det du må huske når du skriver tekst i Python er at du må bruke hermetegn.
- 170 E<sub>2</sub>: Åja, det glemte jeg. ... Ja.
- 171 M: Hvoran tenker du at du skal løse selve oppgaven da? Kan jeg spørre om først om du vet hvor x1 og x2 er plassert i eksempelet?
- 172 E<sub>2</sub>: I denne her?
- 173 M: Ja, på denne figuren.
- 174 E<sub>2</sub>: Vanligvis vil det være her, jeg vet ikke?
- 175 M: Vet du hva nullpunkt er?
- 176 E<sub>2</sub>: Ehh, vi hadde jo nettopp om nullpunkt.
- 177 M: Det jeg kan si er at nullpunkt til en funksjon er punktet som krysser x-aksen. Altså y er lik null.
- 178 E<sub>2</sub>: Jaja.
- 179 M: Programmet vil jo bare vite hvor de er plassert, hvilke x-verdier altså. Hvordan tenker du at du kan løse denne oppgaven.
- 180 E<sub>2</sub>: Mmm, jeg tenker ... jeg tenker på når jeg prøver å løse ett sånt stykke.
- 181 M: Og hva trenger datamaskinen for å løse problemet?
- 182 E<sub>2</sub>: Åsså no har vi jo allerede, noe har vi jo allerede a b og c.
- 183 M: Ja.
- 184 E<sub>2</sub>: så no må vi jo liksom finne forskjellen mellom disse to. Og det er jo at det står pluss her og minus her. Også må jeg ikke skrive inn denne i programmet da? Skrive inn abc-formellen inn i programmet?

- 185 M: Du kan jo prøve og se hva som skjer. [Elev skriver ca. 28 sek]
- 186 E<sub>2</sub>: Jeg jo, jeg kan jo gjøre slik at mmmm.. at i hvertfall denne blir en egen variabel.
- 187 M: Det kan hende.
- 188 E<sub>2</sub>: Ja. [Elev skriver/tenker i ca 22 sek]
- 189 M: Hva er det du... Jeg kan komme rundt for å se skjermen [går rundt bordet].
- 190 E<sub>2</sub>: Men no er jo spørsmålet om den oppfatter det riktig eller på en måte.
- 191 M: Hva mener du.
- 192 E<sub>2</sub>: Eller at liksom, eller jo denne er jo, bare glem det, det går sikkert. .. Sånn. ... x1 er lik ... Eller kanskje vi må ta det som er oppe her først også liksom.
- 193 M: Ser at timen deres er ferdig nå, dette er det siste, så håper det går fint at vi fullfører det.
- 194 E<sub>2</sub>: Ja, ja. [Elev jobber på pc, ca. 15 sek]. Oki, da blir det... [elev jobber ca 5 sek]
- 195 M: Når du gjør det på denne måten må du ha parentes rundt dette stykket, eller så deler den kun på denne med denne.
- 196 E<sub>2</sub>: Ja, selvfølgelig, oki.
- 197 M: Og gangetegn mellom tall. Nå har du skrevet for x1, siden x2 bare har en minus i stedet kan du bare copy-paste.
- 198 E<sub>2</sub>: Ja oki, ..., skal jeg printe det ut også?
- 199 M: Ja.
- 200 E<sub>2</sub>: Og her går det med komma?
- 201 M: Ja, det fungerer. Her har i hvertfall jeg lært noe nytt. ... Så kan du prøve med disse funksjonene. Ja!
- 202 E<sub>2</sub>: Ja, det funket.
- 203 M: Vet du omtrent hvor funksjonen er plassert i koordinatsystemet med disse x-verdiene. Kan vis med å flytte koordinatsystemet [overhead-med x-y akser på] over denne figuren. Trenger ikke helt nøyaktig.
- 204 E<sub>2</sub>: Den første er i hverfall mellom null og 1 og den andre er ehmm, sånn.
- 205 M: Med unntak av feil fortegn på den så ja. Kan prøve den andre funksjonen. ... Ja hva skjer her. Har du en ide om hvorfor du ikke kan forstå svaret?
- 206 E<sub>2</sub>: Kan det ha noe med at b og c er helt like?. Eller.
- 207 M: Du kan jo prøve å regne ut for hånd og se hva som skjer.
- 208 E<sub>2</sub>: Ja, oki [Elev regner i ca. 35 sek]. Her får vi kvadratrot av negativt tall. Da blir det jo ingen svar.
- 209 M: Kan prøve dem siste også.
- 210 E<sub>2</sub>: Ja,...
- 211 M: Ja, hva skjedde her da?
- 212 E<sub>2</sub>: Her ser vi jo at b og c er like igjen, hmm.
- 213 M: Kan du plassere denne på figuren.
- 214 E<sub>2</sub>: Ja, for her er jo begge like, men jeg vet ikke ... [jeg viser elev hvordan den er plassert], åja sånn ja.
- 
- [Intervju avsluttes]

## A.5 Intervju E<sub>3</sub>

- 
- 1 M: Første spørsmålet er hvor motivert er du i matematikkfaget?
- 2 E<sub>3</sub>: Eh, jeg er ganske motivert til å så, ja jeg er ganske motivert.
- 3 M: Ja, hvorfor, på grunn av karakter?
- 4 E<sub>3</sub>: Jeg har et veldig stort ønske om å bli kirurg så det er en ganske stor motivasjon til å få gode karakterer, på grunn av da oppnår jeg mine mål senere i tiden.
- 5 M: Ja, har du hatt noen tidligere erfaringer med programmering i skolesammenheng?
- 6 E<sub>3</sub>: Ja det har jeg hatt på ungdomsskolen.
- 7 M: Husker du noe om det?
- 8 E<sub>3</sub>: Tja, det våknet da litt når vi hadde deg.
- 9 M: Og det var programmering i Python eller Scratc eller?
- 10 E<sub>3</sub>: Det var Python.
- 11 M: Hva husker du fra det opplegget jeg hadde da?
- 12 E<sub>3</sub>: Då husker jeg at vi hadde to forskjellige felt, den ene gav kommandoer og den andre kunne vi skrive rett i. Ja.
- 13 M: Ja.
- 14 E<sub>3</sub>: Og så lærte vi for eksempel a, b, c osv. osv. og så kunne vi programmere då til at pcen skulle regne for oss.
- 15 M: Ja, mhm. Husker du noen av kommandonavnene da?
- 16 E<sub>3</sub>: Jaaah.
- 17 M: For du sier a b c nedover og sånn, hva var det du gjorde med de?
- 18 E<sub>3</sub>: Det husker jeg faktisk ikke.
- 19 M: Tror du programmering kan være ett nyttig verktøy i matematikk?
- 20 E<sub>3</sub>: Ja det tror jeg.
- 21 M: Har du noen tanke om hvordan det kan være?
- 22 E<sub>3</sub>: Altså sånn no fremover skal vi jo prøve å bruke mindre ark og sånne ting og då er jo det en kjempe tjeneste å bare kunne skrive rett inn der.
- 23 M: Ja. Tror du du kommer til å bruke programmering i matematikk, utenom når du blir bedt om det?
- 24 E<sub>3</sub>: Mmm, nei, i og med den retningslinjen jeg skal ta etter skolen så tviler jeg.
- 25 M: Er det noen ting du tror kan bli enklere når du bruker programmering?
- 26 E<sub>3</sub>: Ja, sånn asså merket jeg at regnestykker så kunne du bare skrive inn hva du ville skulle være lik så fikk du bare svaret servert på en måte.
- 27 M: Når du programmerte, var det noe vanskelig syns du?
- 28 E<sub>3</sub>: Ja, det å huske alle de forskjellige tegnene på en måte. Og hva er lik er lik er for eksempel det var litt sånn.
- 29 M: Ja. Nå har vi kommet til oppgavene. Du kan skrive på arket, eller forklare hva du tenker til meg.
- 30 E<sub>3</sub>: Ja. [Elev leser i ca. 19 sek]. Hvordan jeg vil lage et program i Pyth..
- 31 M: Hvordan du vil lage et program i Python ja, hvordan du vil løse dette problemet, finne krysningspunktet. Dette viser et eksempel, programmet skal ikke bare finne løsning på denne oppgaven, men alle linjepar.
- 32 E<sub>3</sub>: Ja, nei men vi har jo fått oppgitt to sånne koder, eller koordinater eller, ja så jeg vil jo skrive inn koordinatene tror jeg. Mm, ja jeg vil skrive inn koordinatene for å finne inn P punktet.
- 33 M: Hvordan?
- 34 E<sub>3</sub>: Jeg ville skrevet parentes, og så heter det kolon det oppe?
- 35 M: Hermetegn?

- 36 E<sub>3</sub>: Ja, jeg ville skrive sånn. ... nei det var uten den, sånn.
- 37 M: Hva kommer du frem til med dette?
- 38 E<sub>3</sub>: Jeg ville skrevet dette og så fylt på inni.
- 39 M: Ja, du kan jo prøve å skrive et program på dataen her da.
- 40 E<sub>3</sub>: Ja.  
[Elev jobber ca. 1 min]
- 41 E<sub>3</sub>: Ja nå fikk jeg jo ingen ting.
- 42 M: Hva som skjedde? ... Ja, hvorfor skjer dette?
- 43 E<sub>3</sub>: Mhm, [mumling].. nei vet ikke.
- 44 M: Husker den siste oppgaven jeg gikk gjennom i timen? Den med lineære ligningsett.
- 45 E<sub>3</sub>: Nei.
- 46 M: For det jeg gjorde der da var at jeg omformet regnestykket slik at jeg fikk kjente på en side og ukjente..
- 47 E<sub>3</sub>: Å ja, ja du flyttet jo over, flytte-bytte regelen.
- 48 M: Ja.
- 49 E<sub>3</sub>: Då ville jeg flytte over ehh, den første.
- 50 M: Du kan skrive ned, kanskje enklere å skrive ned på papiret.
- 51 E<sub>3</sub>: Då ville jeg tatt over 0.5x for grunn av da blir den positiv.
- 52 M: Mhm
- 53 E<sub>3</sub>: Å då måtte vi hatt med ... [elev tenker/skrive]
- 54 M: Kan kanskje hjelpe litt på vei.
- 55 E<sub>3</sub>: Ja.
- 56 M: Vi har jo to linjer, disse to med to ulike formler. Denne linjen med denne formelen og denne med denne. Og det vi vil finne er jo når denne y'en er lik denne, og denne x'en er lik denne.  
[Elev tenker/skriver ca. 15 sek]
- 57 E<sub>3</sub>: Sånn?
- 58 M: Så hvis vi bare prøver å løse denne her da uten å bruke asså det at vi bruker spesifikt disse to linjene då, sånn vi begynner med å sette denne formelen er lik denne formelen og finner.
- 59 E<sub>3</sub>: Å ja, sånn at du regner ut x på en måte på begge.
- 60 M: Ja.
- 61 E<sub>3</sub>: Ja.
- 62 M: Sånn at vi får en verdi på x, den ene i dette punktet.
- 63 E<sub>3</sub>: Så då ville du fått x'en på en side og tallene over på en annen side. [Elev regner på ark] Då må vi flytte å bytte.  
[Elev skriver i ca. 1 min og 40 sek]
- 64 M: Kan jeg bare få se på linjene nå. Det ser ikke ut som at det er rett. Du ser du har minus der.
- 65 E<sub>3</sub>: Åå, ja då blir jo det pluss da.
- 66 M: Ja, ja litt fortegnsfeil, men ellers rett.
- 67 E<sub>3</sub>: Men jeg må fortsatt flytte den over.
- 68 M: Ja.
- 69 E<sub>3</sub>: Så den over der. ... Men blir de, siden de blir liksom ... Blir den pluss da.
- 70 M: Nei, ikke med denne, minus og pluss i addisjon blir minus.
- 71 E<sub>3</sub>: Så må vi dele det.
- 72 M: Her kan du bruke Python, litt lettere, bare husk punktum i stede for komma.

- 73 E<sub>3</sub>: Det var slash som var deletegn?
- 74 M: Ja.
- 75 E<sub>3</sub>: Trykke enter?
- 76 M: Enter når du vil at den skal regne ut ja. ... Problemet nå er at dataen ikke vet hva x er, den er ukjent i dette regnestykket.
- 77 E<sub>3</sub>: Må bare sette inn et tall da for x på en måte.
- 78 M: Ja, men det jeg tenker da er hvis du deler denne med minus tre komma fem
- 79 E<sub>3</sub>: Ja da hadde jeg fått x alene.
- 80 M: Ja
- 81 E<sub>3</sub>: Ja. Så da trenger jeg bare å dele, 17.5 delt på ... sånn, og da kan jeg trykke enter. Og da er x lik fem.
- 82 M: Og hvis du skal finne y verdien?
- 83 E<sub>3</sub>: Da ville jeg satt inn fem der.
- 84 M: Hvor?
- 85 E<sub>3</sub>: x står.
- 86 M: Ja, gjør det.
- 87 E<sub>3</sub>: Føler det blir feil. ...  
[Elev regner ut i ca. 30 sek]
- 88 E<sub>3</sub>: Det blir minus ... eller det blir pluss kanskje.
- 89 M: Minus så rett ut.  
[Elev skriver og mumler i ca. 15 sek]
- 90 E<sub>3</sub>: Fem er lik fem.
- 91 M: Ja, det må jo stemme her
- 92 E<sub>3</sub>: Ja for det jeg gjorde var jo å sette prøve på svaret.
- 93 M: Ja, for begge y-ene er de samme i dette punktet, helt rett.
- 94 E<sub>3</sub>: Mhm, ja. Ja.
- 95 M: Hvordan vil du lage et program som gjør denne utregningen for deg?
- 96 E<sub>3</sub>: Hmm, ja..
- 97 M: Vet du hvilke formel du kan bruke for å beskrive en linje, der du ikke vet disse verdiene?
- 98 E<sub>3</sub>: Da var det a ... ganger b?
- 99 M: Ikke helt, ax pluss b.
- 100 E<sub>3</sub>: a x pluss b, ja så da må jeg ... y er lik ax pluss b
- 101 M: Ja. Og du vil finne kryssning mellom to ukjente linjer som ser sånn ut.
- 102 E<sub>3</sub>: Jo, ja.
- 103 M: ... Så det du ser da, du har a og b i denne formelen sant. De står for...
- 104 E<sub>3</sub>: Det vet vi, null koma fem og syv komma fem.
- 105 M: Ja det var denne linjen. ... så kanskje vi kan bruke andre tall for en ny linje, men ha x og y med der..
- 106 E<sub>3</sub>: Hæ?
- 107 M: Altså at en av linjene er y er lik ax pluss b, så da kan vi bruke y er lik cx pluss d kanskje?
- 108 E<sub>3</sub>: Ja, åja det var jo den delt på ...
- 109 M: Du husker litt av den oppgaven fra timen altså?
- 110 E<sub>3</sub>: Ja du brukte ax pluss b delt på ... da gikk den helt til f da?
- 111 M: Ja, ligningene var litt anderledes. Vi ganget y med tall også.

- 112 E<sub>3</sub>: Ja. ...  
[Elev tenker ca 10 sek]
- 113 E<sub>3</sub>: Ja også kan jeg sette inn tallene der og der.
- 114 M: Ja, så kan du se på hvordan du løste oppgaven på denne siden. [peker på eksempeloppgaven som ble løst tidligere]. Hvordan gjorde du dette..
- 115 E<sub>3</sub>: Satte x på en side.
- 116 M: Du satte x på en side altså, kanskje du skal prøve det. Du setter denne y-en er lik denne.
- 117 E<sub>3</sub>: Ja, men det blir jo, ..., var det ikke sånn?
- 118 M: Det er sånn som dette ja, men nå bruker du a,b, c, d i stede...
- 119 E<sub>3</sub>: I stede for å bruke tall ja. .. Åsså skal jeg gjøre det samme som jeg gjorde her, og då flytter jeg ...  
[mumler + jobber ca. 50 sek]
- 120 M: Hva tenker du å gjøre no då?
- 121 E<sub>3</sub>: Ehhm, vi ganget inn a i hvert ledd, gjorde vi... nei det var en annen oppgave kanskje.
- 122 M: Ja det gjorde vi i timen. I denne oppgaven kan du bruke,.. skal vise [skriver ned  $ex + fx = x(e + f)$ ] kan sette x utenfor en parentes når du har et slikt uttrykk.
- 123 E<sub>3</sub>: Ja, sånn at det og blir x, ..., nei sånn at, ... sånn at du på en måte får like, ..., skjønner ikke helt hvordan jeg skal forklare det, ehm ... ja.
- 124 M: ..... Altså målet er jo at du her skal kunne sette ukjente verdier på en side og kjente på en annen side. Så har du jo x i begge disse to leddene så du kan sette uten for en parentes sånn ..
- 125 E<sub>3</sub>: Sånn at det blir ganget inn i parentesen.
- 126 M: Da får du det vi startet med. Se. Vi prøver å få x alene på en side. Om vi kjenner til verdiene til disse blir de jo et eget tall, hva ville vi gjort da? .... [stille i ca. 10 sek], så det du gjør er at du deler på dette.
- 127 E<sub>3</sub>: Så har du fått x på en eget... ja.  
[Stille ca 11 sek.]
- 128 E<sub>3</sub>: Men hva er.. hva er jeg skal frem til liksom.
- 129 M: Få x på ene siden og alt det andre på denne siden
- 130 E<sub>3</sub>: Da.. ville jeg kanskje delt .. det på det, for da kommer det over forde jeg må gjøre det på begge sider.  
[Elev skriver ned]
- 131 E<sub>3</sub>: Og side jeg gjør det på den siden så vil det gå vekk og da står x alene som d minus b .. c pluss a.
- 132 M: Kan du prøve å bruke dette til å skrive et program. Husker du.. hvordan vil du begynne
- 133 E<sub>3</sub>: Kommando liksom.. men det eneste jeg husker er å skrive a er lik også ett tall.
- 134 M: Ja.
- 135 E<sub>3</sub>: Er det bare det jeg skal gjøre liksom.
- 136 M: Ja. Men du kan bruke disse tallene, da vet vi svaret også, så kan vi se at programmet er rett eller om noe er feil.
- 137 E<sub>3</sub>: Ja, a er lik. Da kan jeg ta den som er tre .. b er lik minus ti ..... minus fem.
- 138 M: Husk punktum i stede for komma med desimal.
- 139 E<sub>3</sub>: Åå ja. og det er lik 7 punktum 5. Sånn. Og då kan jeg skrive der den koden jeg har funnet liksom.
- 140 M: Ja. Her kan det være smart, når du skriver i Python, at det er parantes rundt deler for å få ..
- 141 E<sub>3</sub>: Åja ja for grunn av altså ja.. Da var det det minus det ... [elev skriver ferdig] også trykker jeg enter.
- 142 M: Men hvordan.. men enter nå så kommer ny linje .. så her i dette trykker du der. [Viser til Run Module]
- 143 E<sub>3</sub>: Åja, ja.
- 144 M: Men har du fortalt at datamaskinen skal gi deg en verdi?
- 145 E<sub>3</sub>: Hmm.
- 146 M: Husker du print-kommandoen.

- 147 E<sub>3</sub>: Nei... Må jeg skrive, må jeg ikke skrive det foran her da ..
- 148 M: Ja her, eller der.
- 149 E<sub>3</sub>: Og då var det print ...
- 150 M: Print og så parantes med det du vil skrive ut inn. For eksempel x-verdien.
- 151 E<sub>3</sub>: Okei, så da er det print parantes x, .. print skal utenfor parantesen.
- 152 M: Og hvis du har gjort det rett ser du at print får en annen farge en teksten.
- 153 E<sub>3</sub>: Ja, den ble oransje.
- 154 M: Mhm. Og hvis du nå kjører programmet..
- 155 E<sub>3</sub>: Fem.
- 156 M: Hva skal du gjøre nå. Oppgaven ville jo også at du finner y-verdi,.., at programmet skal finne den for deg også.
- 157 E<sub>3</sub>: Dååå, ....
- 158 M: Du fant jo den på denne siden, .. da du skrev ned
- 159 E<sub>3</sub>: Har jeg?
- 160 M: Ja. y er lik ax pluss b, og y er lik cx pluss d
- 161 E<sub>3</sub>: Ja....
- 162 M: Begge disse må jo stemme
- 163 E<sub>3</sub>: Ja
- 164 M: For at det skal være ett kryssningspunkt.
- 165 E<sub>3</sub>: Jeg er ... ....
- 166 M: For det du kan gjøre, at du skriver inn dette for eksempel.
- 167 E<sub>3</sub>: y er lik ax pluss b.
- 168 M: Ja. For i dette er jo x kjent.
- 169 E<sub>3</sub>: Ja så da setter jeg inn den koden på en måte, .. også etterpå skriver jeg print y. Og det skriver jeg inn her?
- 170 M: Ja.
- 171 E<sub>3</sub>: y er lik ... pluss b ... print y, parantes y. Sånn, og så kan jeg kjøre programmet.
- 172 M: Ja.
- 173 E<sub>3</sub>: Der. x not a [usikker på hva som blir sagt her!]
- 174 M: Åja ja du må ha gangetegn mellom a og x.
- 175 E<sub>3</sub>: Åja ja, og det var den stjernen
- 176 M: Ja det var den stjernen.
- 177 E<sub>3</sub>: x is not defined.
- 178 M: Å du brukte ... åja du har tatt vekk den formelen der du fant x.
- 179 E<sub>3</sub>: Den skrev jeg ikke inn her, eller jo? Nei.
- 180 M: Åja du skrev den ikke inn der inne. For det må du ha før du kan bruke den.
- 181 E<sub>3</sub>: Så da må jeg skrive den. Parantes også er det d minus b delt på ....Sånn no.
- 182 M: Så får vi håpe det fungerer.
- 183 E<sub>3</sub>: Sånn
- 184 M: Ja, der ser vi fem og fem spm stemmer med svaret.
- 185 E<sub>3</sub>: Ja.
- 186 M: Den andre oppgaven handler om ting dere jobbet med da jeg tok deg ut av timen, men er litt usikker på om dere har komt så langt som disse formlene. Men du kan bare lese oppgaven først og så ser vi.



- 187 E<sub>3</sub>: Ja, okei.
- 188 E<sub>3</sub>: Okei så her skal jeg skrive et program som forteller hva x1 og x2 er?
- 189 M: Ja.
- 190 E<sub>3</sub>: Ja.
- 191 M: Vet du hvor x1 og x2 er plassert, .. hvis vi bruker denne figuren og plasserer dette koordinatsystemet over [overhead (gjennomsiktig ark) med koordinatsystem på], hvor plasseres x1 og x2?
- 192 E<sub>3</sub>: Ehhh.
- 193 M: Hvor nullpunktene er plassert.
- 194 E<sub>3</sub>: Nullpunktene til funksjonen.
- 195 M: Ja.
- 196 E<sub>3</sub>: No må jeg nesten gjette for å være ærlig.
- 197 M: Ok.
- 198 E<sub>3</sub>: Mmm, der.
- 199 M: Ja, det er jo rett, der den krysser x-aksen. Y er null. Vi ser jo at en linje har ett kryssningspunkt. Denne kan ha to, og derfor har vi x1 og x2.
- 200 E<sub>3</sub>: Har det noe å si med det vi gjorde i sted.
- 201 M: Tingene du brukte for å få programme til å funke skal være likt, men matematikken er ikke.  
[Elev gir uttrykk for å ikke vite hva h\*n skal gjøre - mumlende]
- 202 M: Kan bare fokusere på disse to funksjonene. Her har vi en funksjon med a, b og c, mens disse tre er med tall plassert inn for disse tre.
- 203 E<sub>3</sub>: Det der x over på en side?
- 204 M: Hva?
- 205 E<sub>3</sub>: Jeg ville jo hatt, siden vi skal finne verdiene for de forskjellige funksjonene så ville jeg flyttet de over på .... en side.
- 206 M: Her har du jo to funksjoner [peker på  $x1 = (-b + \sqrt{b^2 - 4ac})/2a$  og tilsvarende for x2], så er det nødvendig?
- 207 E<sub>3</sub>: Nei. De er allerede gjort der. Oki men då må vi sette inn tall då. .. Siden de allerede er funnet. Så må jeg sette inn tall for at de skal få ..
- 208 M: Verdi?
- 209 E<sub>3</sub>: Ja.
- 210 M: Så om du vil at datamaskinen skal gjøre dette. Python.
- 211 E<sub>3</sub>: Eh, då er jeg nødt til å gjøre det samme som jeg gjorde i sted. Også må jeg skrive inn de forskjellige verdiene til a b og c.
- 212 M: For å finne kvadratroten i Python skriver du sqrt også parentes.
- 213 E<sub>3</sub>: Oki.
- 214 M: Det står for square root.
- 215 E<sub>3</sub>: Ja, så det er kvadratroten på engelsk altså.
- 216 M: Ja.
- 217 E<sub>3</sub>: Ja, men da kan jeg bare sette inn hvilke som helst tall.
- 218 M: Prøve det.
- 219 E<sub>3</sub>: Da bruker jeg de fra denne [peker mot en av eksempel-funksjonene.]
- 220 M: Ja.
- 221 E<sub>3</sub>: For her har vi jo komt ganske langt i regneprosessen da. For vi har jo komt til at vi setter inn i ...
- 222 E<sub>3</sub>: Oki, a det er en, fordi det er x der .. og b er tre .. en også då vil jo jeg skrevet den formelen
- 223 M: Ja

- 224 E<sub>3</sub>: Så då ville jeg skrevet den formelen der, og så vil den løse det der
- 225 M: Ja.
- 226 E<sub>3</sub>: Tror jeg, eg ikkje sikker. Så da skriver jeg x1 på den første. Jeg ville jo bare skrevet den koden sånn som den står, for vi har jo verdier.  
[Elev skriver inn på pc i ca. 15 sek]
- 227 E<sub>3</sub>: Og då var det sqrt..
- 228 M: Og opphøyd i trenger to stjerner.
- 229 E<sub>3</sub>: Det var tre, og i annen ...
- 230 M: Kan jeg komme med et spørsmål?
- 231 E<sub>3</sub>: Ja
- 232 M: Hvorfor skriver du tretallet der?
- 233 E<sub>3</sub>: Oj..
- 234 M: Jeg bare..
- 235 E<sub>3</sub>: Jeg satte bare inn tallene med en gang jeg [elev bytter tall med bosktaver] Vis jeg ikke tar helt feil nå så tror jeg at jeg skal skrive print x1.
- 236 M: Ja, da forteller Python verdien til x1.
- 237 E<sub>3</sub>: En x verdi. Men jeg har ikke fullført hele tingen endå
- 238 M: Ja, så må du ha gangetegn mellom ulike ting sånn der.
- 239 E<sub>3</sub>: Sånn
- 240 M: Ja.. og det samme når du skriver delt på to a, to ganger a.
- 241 E<sub>3</sub>: Ja.. Sånn
- 242 M: Ja.
- 243 E<sub>3</sub>: Men så bruker jeg parantes?
- 244 M: Ja
- 245 E<sub>3</sub>: Så kan jeg skrive print parantes x1
- 246 M: Ja
- 247 E<sub>3</sub>: Sånn
- 248 M: Ja.
- 249 E<sub>3</sub>: Og så må jeg ta ...
- 250 M: Den kom med to varsler .. Det ser ut som at dette er min feil  
[Her var det feil i < from math import sqrt> linjen jeg hadde skrevet]
- 251 M: Da prøver vi dette nå ..
- 252 E<sub>3</sub>: Ja.
- 253 M: Håper dette fungerer.
- 254 E<sub>3</sub>: Ja. Hmm, der kom det et svar.
- 255 M: Så om du kan flytte på dette koordinatsystemet, hvor ville du plassert x1?
- 256 E<sub>3</sub>: Då blir jo den null .. minus null, så då ville jeg plassert den på minussiden. ...
- 257 M: Ja, sånn omtrent.
- 258 E<sub>3</sub>: Det var jo ganske nærmt nå.
- 259 M: Men nå har du bare ett punkt
- 260 E<sub>3</sub>: Så då gjør jeg det samme bare med minus da.
- 261 M: Ja gjør det samme.
- 262 E<sub>3</sub>: Heh, jeg føler det er feil når du sier det på den måten.

- 263 M: Nei det er ikke feil... ikke feil å bare copy-paste og bytte den der [pluss blir minus]
- 264  $E_3$ : Sånn og så er det print x ... Sånn .. også kjøre programmet. Og da fikk jeg minus to komma seks.
- 265 M: Så da er koordinatsystemet rundt sånn da. Neste funksjon da.
- 266  $E_3$ : Samme som i sted?
- 267 M: Ja, gjør det da.
- 268  $E_3$ : Da må jeg gjøre om på verdiene her da.
- 269 M: Ja.
- 270  $E_3$ : Må jeg bare ta vekk alt?
- 271 M: Hvorfor vil du det?
- 272  $E_3$ : Jeg må jo skrive inn verdiene... åå det kan stå sånn, så kan jeg bare skrive nye verdier der sånn da.  
[Elev skrive inn nye verdier]
- 273  $E_3$ : Så kan jeg kjøre programmet
- 274 M: Ja. Sjønner du svaret.
- 275  $E_3$ : Nei.
- 276 M: Denne krysser ikke x-aksen.
- 277  $E_3$ : Derfor har den [vanskelig å forstå], på grunn av at den ikke krysser
- 278 M: Ja. dette er et komplekst svar, kvadratrot av negativt
- 279  $E_3$ : Så det betyr altså at den ikke går innom x-aksen. Skal vi ta siste da
- 280 M: Ja.
- 281  $E_3$ : Så det var rett gjort.
- 282 M: Ja. helt rett.
- 283  $E_3$ : Skjønnte liksom litt mer pointet no når jeg satt her egentlig. Minus to minus to. Det vil si at sånn.
- 284 M: Hvorfor går den gjennom minus to og to? Åja du tenkte minus to på y-aksen.
- 285  $E_3$ : Jha.
- 286 M: Nullpunkt er når y er null, så hvordan .. der ja
- 287  $E_3$ : Er det rett
- 288 M: Ja, det er rett. Da har vi funnet svarene.
- [Intervju avsluttes]
-

## A.6 Intervju E<sub>4</sub>

- 
- 1 M: Hvor motivert er du i matematikkfaget?
- 2 E<sub>4</sub>: Det er mitt favorittfag. Så jeg er ganske motivert.
- 3 M: Ja. Så du liker matte altså.
- 4 E<sub>4</sub>: Ja, jeg liker å regne.
- 5 M: Har du noen tidligere erfaring med programmering?
- 6 E<sub>4</sub>: Programmerte tre år på ungdomsskolen.
- 7 M: Så alle årene.
- 8 E<sub>4</sub>: Som valgfag.
- 9 M: Som valgfag ja.
- 10 E<sub>4</sub>: Ja.
- 11 M: Hvilke språk brukte du då? Python.
- 12 E<sub>4</sub>: Pythhon, Idle, javascript, Scracth, micro:bit
- 13 M: Ja. Du var innom en del da.
- 14 E<sub>4</sub>: Ja.
- 15 M: Hva husker du fra det opplegget jeg underviste i?
- 16 E<sub>4</sub>: Jeg husker det at å skjønne feilmeldingene.. og gjøre endringer slik at jeg kunne få programmet til å virke.
- 17 M: Ja.
- 18 E<sub>4</sub>: Åsså de første oppgavene de var mer logiske.
- 19 M: Mm, ja når du jobber i matematikk tror du programmering kan være et nyttig verktøy?
- 20 E<sub>4</sub>: Ja, altså alt er jo på en måte programmering.
- 21 M: Hva mener du.
- 22 E<sub>4</sub>: Alle formler.. at for å lage et programmeringsprogram så bruker du matte for eksempel så alt vi gjør handler om programmering.
- 23 M: Tror du at du kommer til å bruke programmering i matematikken som verktøy, annet en når du blir bedt om det?
- 24 E<sub>4</sub>: Det tror jeg nok.
- 25 M: Ja.. Er det noe du føler blir enklere når du programmerer?
- 26 E<sub>4</sub>: Har ikke tenkt helt over det egentlig. Men det blir vel lettere etter hvert når vi sjønner litt mer av det.
- 27 M: Av programmering?
- 28 E<sub>4</sub>: Ja.
- 29 M: Ja
- 30 E<sub>4</sub>: Da bruker man det mer uten å tenke kanskje.
- 31 M: Er det noe du føler er vanskelig når du programmerer?
- 32 E<sub>4</sub>: Det er jo å finne ut hvordan du skal starte å begynne et program. Før du kommer deg i mål. Og så er det ikke alle feilmeldingene som er så lette å skjønne.
- 33 M: Ja. Da er vi kommet til oppgavene. Jeg sier igjen at dette ikke er for å teste deg, men for å se hvordan en elev løser oppgavene. Du kan forklare muntlig eller skrive ned hva du tenker. Ehh, så da kan du bare lese gjennom.
- 34 E<sub>4</sub>: Hvordan vi ville løste de i programmering?
- 35 M: Ja.
- 36 E<sub>4</sub>: Eller er det vanlig.

- 37 M: Hvordan du ville brukt programmering for å løse den oppgaven.  
[ca. 20 sek der eleven fortsetter med lesing]
- 38 E<sub>4</sub>: Jeg vet faktisk ikke.
- 39 M: Hva vil du.. hva har du problem med?
- 40 E<sub>4</sub>: Jeg vet ikke hvordan jeg vil løse den ved programering.
- 41 M: Skjønner du hva oppgaven er ute etter?
- 42 E<sub>4</sub>: Lage et program som finner punktene der linjene møter.
- 43 M: Ja. Så har du fått oppgitt to linjer da. Men det trenger ikke være de to.. Vet du hvordan du skal løse dette uten bruk av programmering?
- 44 E<sub>4</sub>: Ikke sånn rett på matte, trenger mer tid til å tenke. Få hode på plass liksom
- 45 M: Ja. Eh, .. vet du hvordan du beskriver dette punktet.
- 46 E<sub>4</sub>: Jeg ville sagt hvilke plasser den har på x-aksen og hvilke plass den har på y-aksen.
- 47 M: Ja. Og når du da har formelen til en linje, hva er det den sier da?
- 48 E<sub>4</sub>: Den sier det siste tallet sier hvor den krysser y-aksen .. og de andre talle sier noe om stigningen
- 49 M: Ja. Og hvis du endrer x-verdien, om du bestemmer deg for en x verdi.
- 50 E<sub>4</sub>: Ja det bestemmes av fortegnet, siden at minus, da vil den gå mot venstre, og dersom det er pluss da går den mot høyre.
- 51 M: Ok, ...
- 52 E<sub>4</sub>: Altså ehh symbolet før ax verdien sier noe om linjen vil være negativ eller positiv.
- 53 M: Ok, men det jeg mente var om du velger en verdi, for eksempel syv.. Hva skjer da
- 54 E<sub>4</sub>: da vil den øke med syv...
- 55 M: Hva mener du
- 56 E<sub>4</sub>: Altså begge vil jo gå opp på syv, men så vil den ene krysse den veien og den andre den veien.
- 57 M: Hva mener du.
- 58 E<sub>4</sub>: I stede for tre på denne tre x minus ti og vi bytter det ut med syv så vil det punktet være en halv under den som allerede er markert på denne.
- 59 M: Ja .. det jeg tenkte på var vis vi bytter x-verdien med ett tall, hva skjer.
- 60 E<sub>4</sub>: Den vil flytte på seg.
- 61 M: Om du har tre ganger syv minus ti.. da får du en y verdi.
- 62 E<sub>4</sub>: jo.
- 63 M: Som du ser, x-verdien her, .. 7 på linjen gir denne y verdien. Skjønner du hva jeg er ute etter?
- 64 E<sub>4</sub>: Nei.
- 65 M: For oppgaven er ute etter punktet der begge linjene har punkt i samme sted. Når x er lik syv, så er y lik 4 og 11 på disse linjene. Vi er ute etter dette punktet, der xene og yene er like.
- 66 M: Husker du den oppgaven jeg gikk gjennom i timen. Med lineære ligningssett?
- 67 E<sub>4</sub>: Ja.
- 68 M: Dette er egentlig en lik oppgave. Husker du hva vi gjorde?
- 69 E<sub>4</sub>: Vi hadde jo den vanlige måten å skrive funksjonsuttrykk på, med a, b, c, d, e og f
- 70 M: Ja
- 71 E<sub>4</sub>: og da byttet vi plass på de ulike bokstavene for å komme frem til
- 72 M: ja, hva ville du fram til
- 73 E<sub>4</sub>: Hvordan linjen gikk.
- 74 M: For i denne oppgaven er det ferre bokstaver. .. og i den andre oppgaven fant vi x er lik noe og y er lik noe

- 75 E<sub>4</sub>: ja
- 76 M: Så du har to ligninger der y er lik noe, og du kan sette disse to lik hverandre.
- 77 E<sub>4</sub>: Altså jeg må jo gjøre noe for at tre ganget med xen minus ti skal være det samme som det.
- 78 M: Ja så hva må du finne ut?
- 79 E<sub>4</sub>: Da må jeg først finne en x som kan .. altså tre ganger det tallet jeg velger, jeg må bruker samme xen på begge
- 80 M: Ja. .. Hvordan kan du finne den x-en da.
- 81 E<sub>4</sub>: Det første jeg tenker er å prøve meg frem, men det tar jo lang tid.
- 82 M: Det tar lang tid ja. .. Så du kan regne ut matematisk ved å sette denne ligningen lik denne å se hvilke x-verdi.
- 83 E<sub>4</sub>: Ja, men jeg får jo samtidig likhetstegn
- 84 M: Hva mener du?  
[Elev viser en ide om å sette  $(y = ax + b) = (y = cx + d)$ , der det er problem med hva som skal gjøres med alle likhetstegnene]
- 85 E<sub>4</sub>: Altså hvis jeg setter parantes rundt sånn, så får vi ...
- 86 M: Altså jeg tenkte å bytte denne y-en med ligningen her.
- 87 E<sub>4</sub>: Ja .... Så....  
[Elev regner ut på ark, ca. 1 min]
- 88 E<sub>4</sub>: Men jeg kan ikke det i hode da.
- 89 M: Det blir fem, søtten koma fem delt på tre komma fem blir fem.
- 90 E<sub>4</sub>: Oki.
- 91 M: Da har du funnet x-verdien, hvordan vil du finne y-verdien
- 92 E<sub>4</sub>: Da vil jeg bytte ut x-en med y.
- 93 M: Ja. ... Ja, nå har du løst for disse to linjene. Oppgaven vil at du skal kunne løse for hvilke som helst linjer.
- 94 E<sub>4</sub>: Så jeg skal bare lage meg to forskjellige linjer?
- 95 M: Eh, ja. Men du vet ikke hva talle er når du skriver programmet.
- 96 E<sub>4</sub>: Mm. Men vi skal velge.. programme skal velge hvilke linjer det skal være?
- 97 M: Du.. vi velger linjen så skal programme fortelle hvor de krysser.
- 98 E<sub>4</sub>: Oki.  
[Elev tenker/skriver ca 20 sek]
- 99 M: Om du har lyst, om du tror det er lettere å bare begynne å skrive programmet, teste ut så kan du bare gjøre det.
- 100 E<sub>4</sub>: Tror ikke det er lettere.
- 101 M: Du tror ikke det er lettere nei...
- 102 E<sub>4</sub>: Jeg har ikke så mye kunnskaper om dette..
- 103 M: Men det gjør ikke noe.
- 104 E<sub>4</sub>: Men er det sånn at vi skal bruke if og else her
- 105 M: Nei, ikke i denne nei. Her trenger du kun det jeg gikk gjennom med dere.
- 106 M: Husker du den med de lineære ligningsettene?
- 107 E<sub>4</sub>: Altså vi lagde jo at det var ax b også c, ax i andre pluss b.
- 108 M: Ikke i dette tilfelle. Har du sett ligningen y er lik ax pluss b før sant?
- 109 E<sub>4</sub>: Ja.
- 110 M: Ja, men passer det for begge linjene? samme tall for a og b

- 111 E<sub>4</sub>: Nei, for da vil det være samme tallene, så man må bruke andre tall,.. , nei andre bokstaver.
- 112 M: Prøv å gjør det da.  
[elev prøver ca. 10 sek]
- 113 M: Ja, når du løste dette her så gjorde du det samme som du skal nå, bare med tall, ikke bokstaver.
- 114 E<sub>4</sub>: ax pluss b er lik cx pluss d .. men da skal det jo være minustegn foran siden den er negativ.
- 115 M: Er den negativ?
- 116 E<sub>4</sub>: Nei... men så må vi liste opp tall .. hvilke bokstaver som hører til hvilket tall.
- 117 M: Ja, ok. Her har du datamaskin så du kan starte med det om du vil. Det blir ofte lettere når man har skrevet det ned.  
[Elev skriver på pc, ca. 40 sek]
- 118 M: Husk at desimaler trenger punktum i stede for komma.
- 119 E<sub>4</sub>: Ja
- 120 M: Desimalseperator heter det vel.
- 121 E<sub>4</sub>: Ja  
[Elev fortsetter å skrive, ca. 33 sek]
- 122 M: Ja så no har du tatt de ulike verdiene her er lik
- 123 E<sub>4</sub>: ja
- 124 M: og så har du her at den ax pluss b er lik cx pluss d.
- 125 E<sub>4</sub>: Nei, det må jeg også skrive ned .. ja
- 126 M: Hvordan gjorde vi det for at programmet vet hva den skal gjøre..  
[Elev ser på oppgaven ca. 15 sek]
- 127 M: .. så du må gjøre noen utregninger først, før du kan skrive det inn i python.
- 128 E<sub>4</sub>: Ja... men jeg kan ikke bare sette inn hvilke som helst x for jeg har ikke definert x.
- 129 M: Så du må finne en x? Det du gjorde i denne oppgaven, her fant du en x
- 130 E<sub>4</sub>: Jeg brukte x og flyttet disse sammen..
- 131 M: Prøv det
- 132 E<sub>4</sub>: men jeg har fortsatt ikke definert x-en her.
- 133 M: Men det gjorde du ikke i denne oppgaven heller. Du flyttet disse dit og så var x alene. Så kom du til at x er lik fem.
- 134 E<sub>4</sub>: Men man trenger ikke det når man skriver det på ark. Men for at programmet skal forstå det må man definere en variabel.
- 135 M: Ja. mhm så da må du regne litt ut først kanskje. For her er x den ukjente, så du kan flytte litt rundt. Så får du at x er noe kjent, men ikke med tanke på tall men bokstaver..
- 136 E<sub>4</sub>: Ja ..  
[Elev skriver inn i Python og prøver å kjøre program]
- 137 E<sub>4</sub>: Dette fungerte ikke.
- 138 M: For python klarer å gjøre det.. utregning på begge sider av erlikhetstegn?
- 139 E<sub>4</sub>: Nei... men jeg mangler noe.
- 140 M: Ja! Mitt råd er å regne ut forhånd, som du gjorde her, nedover på arket her og se hva du får. Finne noe python kan klare.  
[Elev skriver inn i Python]
- 141 M: Husk gangetegn mellom to ting som her.
- 142 E<sub>4</sub>: Og så må jeg snu om på formelen, sånn at jeg får x alene.
- 143 M: Du kan jo prøve å kopiere det du gjorde her men bytte ut tallene med bokstaver.

- 144 E<sub>4</sub>: Men vi må jo definere x-en.
- 145 M: Men her definerte du ikke x-en når du gjorde dette.
- 146 E<sub>4</sub>: Men det er anderledes på papir enn det er på python.
- 147 M: Det er derfor det kan være en ide å skrive det på papir først.  
[Elev begynner å skrive utregning på papir]
- 148 M: Ja her, her trenger du kanskje å faktorisere.
- 149 E<sub>4</sub>: Ja.
- 150 M: Åja, ja du klarer det faktisk. Bra. Ta ut sånn ja.  
[Elev skriver viderer]
- 151 M: Ja, der er x.
- 152 E<sub>4</sub>: Jo
- 153 M: beskrevet ved hjelp av kun kjente verdier.
- 154 E<sub>4</sub>: Jo  
[Elev skriver inn i Python]
- 155 M: Det siste er du vil i programmet er jo å vite hva det er.
- 156 E<sub>4</sub>: Jo  
[Elev skriver inn med print]
- 157 M: Jeg husker jo at du begynte formatering, at jeg hjalp deg med det i timen
- 158 E<sub>4</sub>: Jo
- 159 M: Litt mer avansert. Så ser at du kan skrive slik med disse tegnene sånn. Da er det .format() også inni
- 160 E<sub>4</sub>: Det var det du viste meg  
[Elev skriver inn print('x = , y = ',format(x, y))]  
[Mer mumling da programmet ikke fungerte, fra begge, om å finne/rette feil]
- 161 E<sub>4</sub>: Er det komma eller punktum?
- 162 M: Du velger det selv,.. sånn inni tekst der i hvertfall. Dette er jo bare for at det skal se fint ut.
- 163 M: Så kan du kjøre programmet så ser vi igjen
- 164 M: Åja, må være punktum format.
- 165 E<sub>4</sub>: Oj  
[skrivning ca. 15 sek]
- 166 M: Enda en feilmelding.
- 167 E<sub>4</sub>: Hmm.  
[Lesing av kode ca 20 sek]
- 168 M: Problemet er der, de hermetegnet ... Kanskje fjerne mellomrommet inni den der ..  
[Kodesnutten print('x = , y = ',format(x, y)) er nå rettet til print('x = , y = '.format(x, y))]
- 169 M: Nå funknet det.
- 170 E<sub>4</sub>: Fem, og fem som er det vi kom frem til der.
- 171 M: Ja. Då går vi over på oppgave to. Dette jobbet jo dere med idag så jeg. Du kan jo lese..  
[Elev leser ca. 30 sek]
- 172 M: Ja, det var vel den formelen dere lærte no, \*\*\*\* gikk gjennom da jeg kom idag.
- 173 E<sub>4</sub>: Ja, og disse er samme,.. nei det er pluss og minus.
- 174 M: Ja. Og jeg har tatt med en transparant med koordinatsystem, og figuren sånn, da kan vi se at figuren kan være plassert hvor som helst i koordinatsystemet.



- 175 E<sub>4</sub>: Ja.
- 176 M: Så for det første skjønner du hva oppgaven ber om?
- 177 E<sub>4</sub>: Altså, funksjonen den, så skal vi printe de to
- 178 M: Ja..
- 179 E<sub>4</sub>: Eller vise de.
- 180 M: Hva må du gjøre for å få det til
- 181 E<sub>4</sub>: Vi må først definere de.
- 182 M: Ja.
- 183 E<sub>4</sub>: Og da kan vi ikke bruke ukjente, som a b og c, men vi må endre sånn at vi sier bare de tre.
- 184 M: Ja. Jeg har med tre eksempelfunksjoner.. her. Eksempler på hva abc kan være.
- 185 E<sub>4</sub>: Ja. Da må vi definere a b og c, det samme som forrige.
- 186 M: Mhm, kan jo gjøre det.
- 187 E<sub>4</sub>: Skal jeg ta utgangspunkt i en av de tre formlene
- 188 M: Ja, kan begynne øverst.  
[Elev skriver på pc, ca. 20 sek]
- 189 M: Og hva er det neste etter dette  
[12 sek tenking]
- 190 E<sub>4</sub>: Mmm, må definere de.
- 191 M: Ja
- 192 E<sub>4</sub>: Så da kan jeg skrive, .. ,hvordan skriver man det i Python
- 193 M: Jeg ville brukt x1 for eksempel. Python kan ikke nedsenke tall sånn
- 194 E<sub>4</sub>: Ja, ok.  
[Skriver inn i ca. 25 sek]
- 195 E<sub>4</sub>: Hvordan skriver pluss minus? Nei du har jo separert de to
- 196 M: Ja, der oppe. Kan bruke sqrt [viser på ark sqrt(utrykk)]
- 197 E<sub>4</sub>: Men måkke det være inni en parantes, sånn hele den jeg skal ta inni en parantes.
- 198 M: Hva mener du
- 199 E<sub>4</sub>: Ehh, det, så tar vi dobbelparantes rundt det og rundt det så får vi..
- 200 M: Det kan fungere, altså du viser hvordan programmet skal regne ut, den tar det inni først når den regner.  
[Elev mumler/skriver inn ca. 45 sek]
- 201 E<sub>4</sub>: Men jeg må ha alt dette i en parantes ...
- 202 M: For å
- 203 E<sub>4</sub>: For å så ta delt på
- 204 M: Jah  
[Mer skriving, ca. 20 sek]
- 205 E<sub>4</sub>: Trenger jeg å gjøre det egentlig?
- 206 M: Usikker, men når du først har gjort det tror jeg ikke det har så mye å si
- 207 E<sub>4</sub>: Da har jeg de to
- 208 M: Ja
- 209 E<sub>4</sub>: Også skal jeg [hører ikke hva eleven sier, ca. 6 ord]
- 210 M: Printe ut x1 og x2 ja. Så du trenger ikke skrive på denne formen. Oppgaven ba jo om å få disse to
- 211 E<sub>4</sub>: Ja, jeg har jo regnet frem og tilbake mellom disse to, de er de samme [elev peker på at  $ax^2+bx+c=a(x-x_1)(x-x_2)$ ]

- 212 M: Ja disse er de samme
- 213 E<sub>4</sub>: Ja, bare faktorisert.
- 214 M: Ja, og vi er kun interresert i hva  $x_1$  og  $x_2$  er i denne oppgaven.  
[Elev skriver på pc, ca. 20 sek]
- 215 M: Ja, det du kan gjøre, om du ikke er interresert i å formatere kan du bare skrive print  $x_1$  komma  $x_2$
- 216 E<sub>4</sub>: Men det vil jeg ha med
- 217 M: Det vil du ha med ja
- 218 E<sub>4</sub>: Ja. Jeg vil jo det skal se fint ut.  
[Elev skriver på pc, programmet kjører med feilmelding]
- 219 E<sub>4</sub>: Er det denne første
- 220 M: Ja.
- 221 E<sub>4</sub>: Så bytte ut det som er inni parantesen med  $x_1$  og  $x_2$ , ja du har  $y$  der, den heter  $x_2$ .  
[Elever skriver på pc]
- 222 M: Så kan du kjøre programmet, ..., ja der ser du får to verdier. Kan du flytte på dette sånn at funksjonen er plassert rett på koordinatsystemet?
- 223 E<sub>4</sub>: Ja. Sånn at det stemmer med det?
- 224 M: Ja. Du trenger ikke være veldig nøyaktig, men du ser du har plassert sånn ca rett der, nullpunktene.
- 225 E<sub>4</sub>: Ja.
- 226 M: Mhm, kan jo prøve å finne på den funksjonen.
- 227 E<sub>4</sub>: Skal jeg bare bruke samme program?
- 228 M: Ja
- 229 E<sub>4</sub>: Skal det være samme da...
- 230 M: Prøv å flytt sånn at dette stemmer da? [Bruker figur og overhead]
- 231 E<sub>4</sub>: Blir det sånn
- 232 M: Ja. Kan prøve denne oppgaven nå. Se hva som skjer.
- 233 E<sub>4</sub>: Den er ikke lineær
- 234 M: Hva tenker du
- 235 E<sub>4</sub>: Det vil si at hvis du bruker abc-formelen, vi kan ikke ta kvadratroten av et negativt tall.
- 236 M: Ja. Kan jo se.. vet du hvor ville du plassert denne funksjonen på koordinatsystemet?
- 237 E<sub>4</sub>: Jeg vet ikke...
- 238 M: Ja, er jo over  $x$ -aksen, eller under. Men den krysser ikke.
- 239 E<sub>4</sub>: Ja.  
[Intervju avsluttes]
-

## A.7 Intervju E<sub>5</sub>

- 
- 1 M: Så begynner vi med spørsmålene. Hvor motivert er du i matematikkfaget?
- 2 E<sub>5</sub>: Ehh, det kommer veldig an på dagen, ehmm men jeg har jo selvfølgelig veldig lyst å få god karakter i matematikk så det kommer på vitnemålet.
- 3 M: Ja
- 4 E<sub>5</sub>: Sånn at jeg kan ha det lett i matte hvis jeg følger med ordentlig.
- 5 M: Mhm. Eh før det opplegget jeg underviste i, har du hatt noen erfaring med programmering?
- 6 E<sub>5</sub>: Nei.
- 7 M: Ingen erfaring fra skolesammenheng?
- 8 E<sub>5</sub>: Ingen ting
- 9 M: Ingen ting, Eh hsuker du noe fra det opplegget, .. timen jeg hadde.
- 10 E<sub>5</sub>: Mmm, vi gikk gjennom sånn kommando greier tror jeg. Kunne få den til å si ting, eller skrive ting
- 11 M: Mhm, det er alt du husker sånn her nå?
- 12 E<sub>5</sub>: Ja.
- 13 M: Tror du programmering kan være et nyttig verktøy i matematikk?
- 14 E<sub>5</sub>: Ja hvis man finner ut at, eller hvis man lærer å bruke det veldig sånn matte messig så tror jeg det kan være veldig lurt. For det var ikke vanskelig når man først skjønnte det så var det bare å skrive de tingene så gjorde den mye for deg.
- 15 M: Ja. Eh, tror du du kommer til å bruke programmering i matematikk, utenom når oppgave krever det?
- 16 E<sub>5</sub>: Mmm det kommer også an på hvor mye jeg lærer, hvor mye jeg forstår det.
- 17 M: Så det er avhenging av hvor mye du kan av programmering.
- 18 E<sub>5</sub>: Ja.
- 19 M: Em , er det noe du tror kan bli enklere med programmering?
- 20 E<sub>5</sub>: Noe jeg .. blir enklere ..
- 21 M: Når du jobber matematikk for eksempel.
- 22 E<sub>5</sub>: Ja, den dere lignings.. når det var sånn to ligningsoppsett, å ha den formelen bare klar i programmeringsoppsettet. Det var jo veldig..
- 23 M: Mmm, ja. Er det noe du følte ble vanskeligere med programmering? Altså var vanskelig med programmering.
- 24 E<sub>5</sub>: Å huske ting å skrive ned og sånne ting og sånt.
- 25 M: Då går vi over på oppgavene. Då vil jeg at du leser oppgaven. Og etterpå kan du si hva du tenker, eller skrive på dette arket.  
[Elev leser. Ca. 40 sek]
- 26 E<sub>5</sub>: Mmmm.
- 27 M: Skjønnte du oppgaven?
- 28 E<sub>5</sub>: Jeg vet ikke.
- 29 M: Du har jo dette eksempelet. Denne figuren.
- 30 E<sub>5</sub>: Ja, så skal jeg finne punktet.
- 31 M: Ja, men oppgaven vil at du finner for to linjer, trenger ikke være disse. Altså her har du to linjer som er gitt. Det trenger ikke være saken før du har skrevet programmet.  
[Elev tenker ca. 20 sek]
- 32 M: Så poenget med oppgaven skal forklare hvordan du går frem. Hvilke formel.. husker du formelen for å beskrive en rett linje?
- 33 E<sub>5</sub>: Nei..

- 34 M: Så du husker ikke det?
- 35 E<sub>5</sub>: Nei.
- 36 M: Kan skrive ned [skriver ned  $y = ax + b$ ]
- 37 E<sub>5</sub>: Har brukt GeoGebra på ungdomsskolen liksom.
- 38 M: Ja. På ungdomsskolen. Du har sett denne funksjonen før.
- 39 E<sub>5</sub>: Mmm, ja.
- 40 M: Ja. Og det er formelen til en linje der a er stigningen og b er der den krysser y-aksen.
- 41 E<sub>5</sub>: Ja.
- 42 M: Og da hvis du ser på disse formlene her  
[Viser sammenheng mellom a og b på eksempelet, ca. 1 min]
- 43 E<sub>5</sub>: Ja, ja, men jeg vet ikke hvordan jeg finner dette punktet.
- 44 M: Husker du i timen jeg hadde, det med ligningsettet?
- 45 E<sub>5</sub>: Ja, vi satte de opp sånn, over hverandre.
- 46 M: Ja, ja du kan jo gjøre det. Kan jo vise med disse to eksempellinjene. Finne en x og y verdi  
[Elev skriver på ark, ca. 20 sek]
- 47 E<sub>5</sub>: Er det sånn man gjør det?
- 48 M: Ja, y er lik dette ja. Så hvis du setter x er lik en, her på denne figuren, så ser du at dette, så går du opp på linjen og ser bort dit og har en y verdi og.
- 49 E<sub>5</sub>: Ja.
- 50 M: Så hvis x er lik en har de to linjene y er lik minus null komma fem og syv komma fem. Sant, og om du bytter x ut med 1 i ligningene får du det samme.  
[Forklarer litt videre om linjene og kryssningspunkt]
- 51 E<sub>5</sub>: Så skal jeg gjøre bort hit?
- 52 M: Vi vil finne der y-verdiene er like og x-verdiene er like.
- 53 E<sub>5</sub>: Ja
- 54 M: ...Altså der dette er lik dette.
- 55 E<sub>5</sub>: Ja.
- 56 M: Og det du kan gjøre er å sette dette lik dette. At høyre side her er lik denne høyre siden. Så bruker du dette til å finne x-verdien.
- 57 E<sub>5</sub>: Så det er som en vanlig ligning?
- 58 M: Ja.  
[Elev regner ut på ark ca. 50 sek]
- 59 M: Kan si at dette delestykket blir fem.
- 60 E<sub>5</sub>: Ja.
- 61 M: Og nå skal du ..
- 62 E<sub>5</sub>: Jeg setter x-en inn i ... der.
- 63 M: Prøv det ja.  
[Elev regner på ark ca. 15 sek]
- 64 E<sub>5</sub>: De er like.
- 65 M: Ja begge er fem. Og dette stemmer på figuren, vi ser en to tre fire fem og en to tre fire fem.
- 66 E<sub>5</sub>: Ja.
- 67 M: Nå har vi brukt disse eksemplene, det oppgaven spør om er
- 68 E<sub>5</sub>: Lage en..

- 69 M: At linjene ikke har oppgitt verdier der.
- 70 E<sub>5</sub>: Mmm. At den tre x på en måte blir a og y er lik ax minus b og y er lik cx minus d.
- 71 M: Ja. Kan si at jeg ville brukt pluss ikke minus, men det fungerer dette også.
- 72 E<sub>5</sub>: Sånn at det blir som en formel på en måte.  
[Elev regner på arket, ca. 40 sek]
- 73 M: Og hva vil du gjøre nå?
- 74 E<sub>5</sub>: Det samme som i sted.
- 75 M: Mhm.  
[Elev regner på ark, ca. 15 sek]
- 76 E<sub>5</sub>: Åsså samme sånn.
- 77 M: Ja. ... Men no har du ikke tall
- 78 E<sub>5</sub>: Nei, bare bokstaver sånn.  
[Elev regner på ark, ca. 30 sek]
- 79 E<sub>5</sub>: Er det ax minus cx er lik d minus b?
- 80 M: Ja, nesten i mål nå. Her kan vi faktorisere ut x
- 81 E<sub>5</sub>: Jaa. At jeg tar x a minus d er lik d minus b.
- 82 M: Ja, nå her du kommet til dette stedet, der du delte på tre komma fem.
- 83 E<sub>5</sub>: Så da putter jeg inn x-en i den formelen?
- 84 M: Men da har du hoppet over delingen.
- 85 E<sub>5</sub>: Åja dele steget, .. deler jeg på .. a minus d da, [mumling]. Også putter jeg det inn i den
- 86 M: Ja. Men tror du du trenger det når du programmerer?
- 87 E<sub>5</sub>: Nei, da skriver du bare x er lik i en egen, .. og da skjønner den det.
- 88 M: Her har du en pc du kan skrive programmet i. Jeg kan hjelpe med kommandoer og sånt dersom du er usikker.
- 89 E<sub>5</sub>: Ja, .. hvordan var det man skrev først, alle de tingene.
- 90 M: For å oppgi verdier? Skrive a er lik.
- 91 E<sub>5</sub>: Så du skriver ikke print først?
- 92 M: Det kan vi, det vil jo gjøre at du kan fortelle hva programmet gjør ja. Men vi trenger ikke..
- 93 E<sub>5</sub>: Oki, Ok. Så a er lik
- 94 M: Du kan bruke tallene fra dette eksempelet nå.  
[Elev mumler tallene og bokstavene som blir skrevet inn]
- 95 E<sub>5</sub>: Hmm, syv komma fem.
- 96 M: Husk at desimal markeres med puntum i Python.
- 97 E<sub>5</sub>: Og så skriver jeg inn den..?
- 98 M: Hva tror du må komme først?
- 99 E<sub>5</sub>: y er lik a x minus b..
- 100 M: For du har disse to ligningene..
- 101 E<sub>5</sub>: x-verdi først.
- 102 M: Ja.
- 103 E<sub>5</sub>: Og så y-verdi ..... Sånn
- 104 M: Du har glemt gangetegn..
- 105 E<sub>5</sub>: Mellom bokstaver

- 106 M: Ja.
- 107 E<sub>5</sub>: Sånn... Print parantes, sånn hermetegn?
- 108 M: Det trenger du ikke her, men vanligvis ja.
- 109 M: Nå kan du bare kjøre programmet og se.
- 110 E<sub>5</sub>: Fem
- 111 M: og fem
- 112 E<sub>5</sub>: fem
- 113 M: Som var forventet.
- 114 E<sub>5</sub>: Ja x og y er begge fem.
- 115 M: Da var vi komt til oppgave to. Kan lese gjennom..  
[Elev leser, ca. 40 sek]
- 116 M: Forstår du oppgaven?
- 117 E<sub>5</sub>: Ja, det er abc-formelen, er det ikke?
- 118 M: Ja. Det første jeg kan vise, det at det er et koordinatsystem, har jeg med en overhead slik at vi kan flytte x og y aksene i figuren, her.
- 119 E<sub>5</sub>: Ja.
- 120 M: Ja, hvordan vil du lage dette programmet, et program som tar inn a b og c og printer ut to x-verdier, x1 og x2?
- 121 E<sub>5</sub>: Mmmm, jeg ville, mmm, skrevet inn først funksjonen. Eller
- 122 M: Det blir litt,..., hva trenger du for å regne ut x1 og x2
- 123 E<sub>5</sub>: a b og c
- 124 M: Ja, så trenger ikke skrive inn hele funksjonen
- 125 E<sub>5</sub>: Må bare ha formelen. Disse to
- 126 M: Ja. Du kan jo bare skrive inn formelen. For å få kvadratrot skriver du sqrt parantes og så det som skal ta kvadratrot av.
- 127 E<sub>5</sub>: Ja, så jeg skriver inn først x1 og så formelen...  
[Elev skriver på Pc ca. 1 min]
- 128 E<sub>5</sub>: Trenger jeg å skrive inn x2 også?
- 129 M: Det er to, .. om du ser på aksene her hvor ville du plassert x1 og x2?
- 130 M: Ja. Det er to ulike verdier
- 131 E<sub>5</sub>: Mhm
- 132 M: Og for å regne de ut må du bruke to ulike formler. Men du kan kopiere denne og bare endre det som er ulikt. Trenger ikke skrive opp igjen alt.  
[Elev skriver på pc, ca. 35 sek]
- 133 E<sub>5</sub>: Da her jeg skrevet inn begge formlene
- 134 M: Mhm. Her har jeg tre ulike funksjoner. Jeg vil at programmet du lager skal finne x1 og x2 for alle disse.
- 135 E<sub>5</sub>: Mm, da må jeg bare legge inn a b og c verdien kanskje
- 136 M: Ja.
- 137 E<sub>5</sub>: Er lik, i den første så er det lik en b er lik tre og c er lik en. Også er det print x1 og print x2
- 138 M: Mhm.  
[Ca 20 sek skriving på pc]
- 139 E<sub>5</sub>: Og så er det kjør program?
- 140 M: Ja, hvis programmet er ferdig så
- 141 E<sub>5</sub>: Eller skal jeg skrive inn alle.

- 142 M: Nei, det trenger du ikke. Kan ta en og en om gangen.
- 143 E<sub>5</sub>: Er det ferdig
- 144 M: Bare prøv å kjør
- 145 E<sub>5</sub>: Nei...
- 146 M: Problemet nå er at det er viktig å huske at programmet leser fra første linje og nedover.
- 147 E<sub>5</sub>: Så jeg bør ha verdiene før jeg bruker de?
- 148 M: Ja.
- 149 E<sub>5</sub>: Så det kommer på oppsiden og så skriver jeg a og b c og d verdiene
- 150 M: Mhm.
- 151 E<sub>5</sub>: Så kjører jeg program .. Nei ikke nå heller
- 152 M: Ja, kan du lese feilmeldingen?
- 153 E<sub>5</sub>: Mmm fire ac ... må ha gange i mellom ... sånn
- 154 E<sub>5</sub>: Og tallet ja.
- 155 M: Den fikk det ikke til nå heller.
- 156 E<sub>5</sub>: Nei. Er det fortsatt .. feil?
- 157 M: Skal bare se for den sier at er noe ... [ser etter feil, tar ca. 50 sek]
- 158 M: Det er litt merkelig for jeg ser.. programmet ser rett ut, det skjer ikke rette ting .. åå vent litt, du har bruk den vanlige måte å skrive opphøyd i [ $\wedge$  tegnet], python bruker to stjerner .. gange gange
- 159 E<sub>5</sub>: Åja, to stjerner. .. Ja der fikk vi i hvertfall to verdier, men de var veldig..
- 160 M: Ikke fine? Du kan jo vise hvor du tror figuren er plassert på koordinatene her.
- 161 E<sub>5</sub>: Der.. der .. hm sånn.
- 162 M: Fortegnet er feil, men tallet rett på denne.
- 163 E<sub>5</sub>: Åj
- 164 M: Kan prøve neste funksjon da.
- 165 E<sub>5</sub>: Da bare bytter jeg inn verdiene.
- 166 M: Mhm.
- 167 E<sub>5</sub>: Nei, gjør sånn...
- 168 M: Programmet fungerer, har du noen ide om hvorfor det ikke gir svar?
- 169 E<sub>5</sub>: Jeg endret jo bare på a b og c
- 170 M: Ja
- 171 E<sub>5</sub>: Og det funket i sted
- 172 M: Sp det kan ha med disse verdiene
- 173 E<sub>5</sub>: De er like, for eksempel b og c er lik
- 174 M: Da kan du prøve denne funksjonen, se om det stemmer. Disse er like og, hva skjer da?
- 175 E<sub>5</sub>: Oj, der gikk det.
- 176 M: Ja, hvor plasserer du denne da?
- 177 E<sub>5</sub>: Er begge minus to?
- 178 M: Ja rett. ..
- 179 E<sub>5</sub>: Sånn
- 180 M: Ja for du trenger bare at den er ani x-aksen
- 181 E<sub>5</sub>: Ååå, men hvorfor funket det ikke når det var to.
- 182 M: To og to, det funket ikke fordi funksjonen, denne er plassert over .. sånn .. og nullpunkt på ha y er lik null

183 E<sub>5</sub>: Ja. Og det liker ikke programmet?

184 M: Nei, for om du ser her så får du fire minus åtte, kvadratroten av dette..

185 E<sub>5</sub>: Det går ikke, det blir ingen løsning.

186 M: Da er det ingen løsning. Da krysser ikke funksjonen x-aksen

187 E<sub>5</sub>: Ja

[Intervju avsluttes]

---



Tillegg B

Svar fra NSD



Det innsendte meldeskjemaet med referansekode 525425 er nå vurdert av NSD.

Følgende vurdering er gitt:

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 10.09.2020, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

#### DEL PROSJEKTET MED PROSJEKTANSVARLIG

Det er obligatorisk for studenter å dele meldeskjemaet med prosjektansvarlig (veileder). Det gjøres ved å trykke på "Del prosjekt" i meldeskjemaet.

#### MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde:

[nsd.no/personvernombud/meld\\_prosjekt/meld\\_endringer.html](https://nsd.no/personvernombud/meld_prosjekt/meld_endringer.html)

Du må vente på svar fra NSD før endringen gjennomføres.

#### TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 31.12.2020.

#### LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake. Lovlig grunnlag for behandlingen vil dermed være den

registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

#### LÆRERES TAUSHETSPLIKT

Informantene i prosjektet er lærere, og har taushetsplikt. Det er viktig at intervjuene gjennomføres slik at det ikke registreres taushetsbelagte opplysninger. Vi anbefaler at dere minner informantene om dette i forbindelse med intervjuene.

#### PERSONVERNPRINSIPPER

NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke behandles til nye, uforenlige formål
- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

#### DE REGISTRERTES RETTIGHETER

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), underretning (art. 19), dataportabilitet (art. 20).

NSD vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

#### FØLG DIN INSTITUSJONS RETNINGSLINJER

NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og/eller rådføre dere med behandlingsansvarlig institusjon.

#### OPPFØLGING AV PROSJEKTET

NSD vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Lykke til med prosjektet!



Tillegg C

Brev til skole





Hei!

Jeg er en masterstudent i matematikdidaktikk ved Universitet i Bergen. Gjennom masteren skal jeg se nærmere på hvordan programmering på en god måte kan innlemmes i matematikkundervisningen. I forbindelse med dette ønsker jeg derfor å gjennomføre ett opplegg for en eller flere klasser som tar 1T på studiespesialisering.

Opplegget tar i bruk programmering i Python som hjelpemiddel for å utforske et matematisk tema. Opplegget utarbeides gjerne i samarbeid med lærer, og er ikke avhengig av et bestemt matematisk tema. Målet med opplegget vil være å se hvordan programmering passer inn i undervisningen i matematikkfaget, og det vil blant annet passe inn med følgende kompetansemål fra den nye læreplanen:

- formulere og løse problem ved hjelp av algoritmisk tenkning, ulike problemløsningstrategier, digitale verktøy og programmering (Kilde: <https://www.udir.no/lk20/mat09-01/kompetansemaal-og-vurdering/kv42>)

All data jeg samler inn vil bli anonymisert. Dersom du har en 1T klasse og dette er noe du kunne vært interessert i, eller du kan dele dette med lærere du tror kunne vært interessert, hadde det vært til stor hjelp!

Dersom du er interessert eller har noen spørsmål kan jeg kontaktes på [egil.dahle@student.uib.no](mailto:egil.dahle@student.uib.no)

På forhånd takk!

Med vennlig hilsen Egil Johnsen Dahle



Tillegg D

Intervju informasjonskriv



## Vil du delta i forskningsprosjektet

### "Hvordan kan programmering innlemmes i matematikkundervisningen?"

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å se på programmering og algoritmisk tenking i matematikkundervisning. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

### Formål

Dette er en kvalitativ studie som ser på hvordan programmering og algoritmisk tenking kan innlemmes i matematikkundervisningen. Prosjektet vil se på og analysere undervisningsøkter som har brukt programmering og algoritmisk tenking som del av matematikkundervisningen. Dette er en mastergradsoppgave innen matematikdidaktikk.

### Hvem er ansvarlig for forskningsprosjektet?

Matematisk institutt ved Universitetet i Bergen er ansvarlig for prosjektet. Prosjektleder er førsteamanuensis Johan Lie og student Egil Dahle.

### Hvorfor får du spørsmål om å delta

Du har deltatt i undervisningsopplegg som har brukt programmering og algoritmisk tenking som del av matematikkundervisningen, ledet av student.

### Hva innebærer det for deg å delta?

Hvis du velger å delta i prosjektet innebærer det at du bli intervjuet der det blir tatt lydopptak. Det vil ta deg ca.30-45 min. Svarene dine vil bli lagret elektronisk på lydopptaksenhet og transkribert (skrevet ned) til digitalt tekstformat.

## Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Intervjuet vil gjennomføres i skoletiden, tidspunkt avtales mellom deltaker og student eller gjennom lærer. Om deltaker er elev vil eventuelle lærere/skole bli rådført dersom dette skulle gå ut over undervisningstid.

## Ditt personvern - hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

Det er bare student og veileder som får tilgang til lydopptak. Navn og kontaktopplysninger vi får tilgang til vil erstattes med kode/alias og lagres separat fra lydopptak, notater og transkripsjon.

I publikasjon vil skole, klasse og elev anonymiseres.

## Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

I oppgaven vil alle opplysninger være anonymisert, og når oppgaven blir godkjent vil alle navn, kontaktopplysninger, og lydopptak slettes/destrueres. Dette vil etter planen skje i slutten av desember 2020.

## Dine rettigheter

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke personopplysninger som er registrert om deg, og å få utlevert en kopi av opplysningene,
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og

- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

## Hva gir oss rett til å behandle opplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Universitetet i Bergen har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

## Hvor kan jeg finne ut mer?

Hvis du har spørsmål til studien, eller ønsker å benytte deg av dine rettigheter, ta kontakt med:

Universitetet i Bergen ved

- Egil Johnsen Dahle, student, på e-post [egil.dahle@student.uib.no](mailto:egil.dahle@student.uib.no) eller telefon: 47 89 06 09
- Johan Lie, veileder, på e-post [johan.lie@uib.no](mailto:johan.lie@uib.no)

Hvis du har spørsmål knyttet til NSD sin vurdering av prosjektet, kan du ta kontakt med:

- NSD – Norsk senter for forskningsdata AS på epost ([personverntjenester@nsd.no](mailto:personverntjenester@nsd.no)) eller på telefon: 55 58 21 17.

Med vennlig hilsen

Prosjektansvarlig  
(Veileder)

Johan Lie

Student

Egil Johnsen Dahle

## Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet *Hvordan kan programmering innlemmes i matematikkundervisningen?* og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i intervju
- at det blir tatt lydopptak av intervjuet

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

---

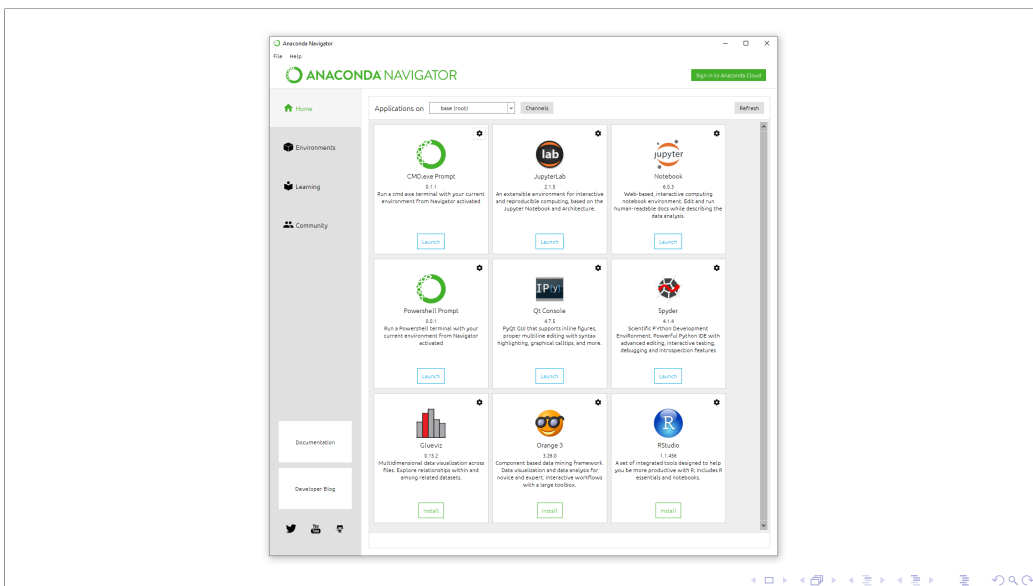
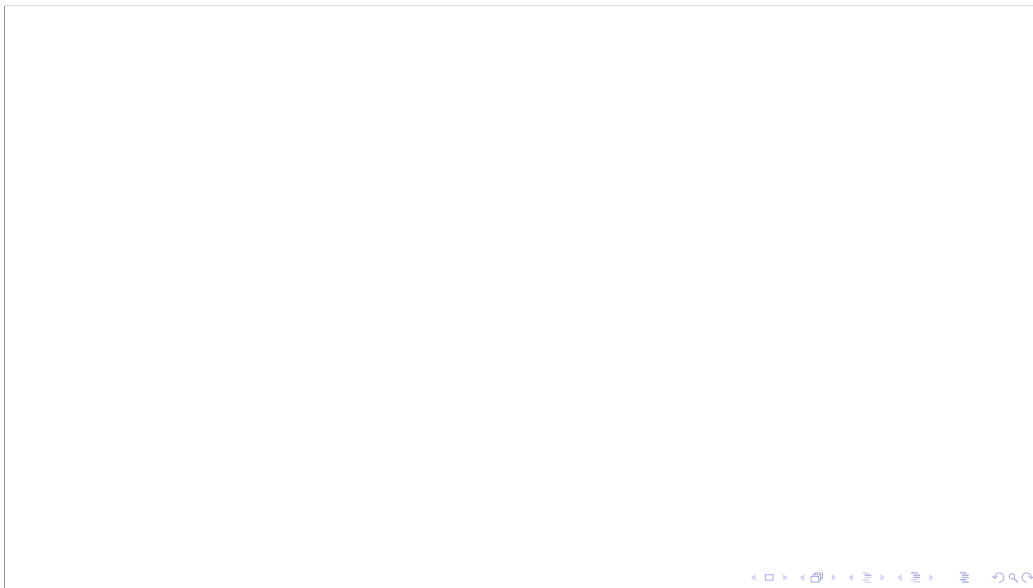
(Signert av prosjektdeltaker, dato)

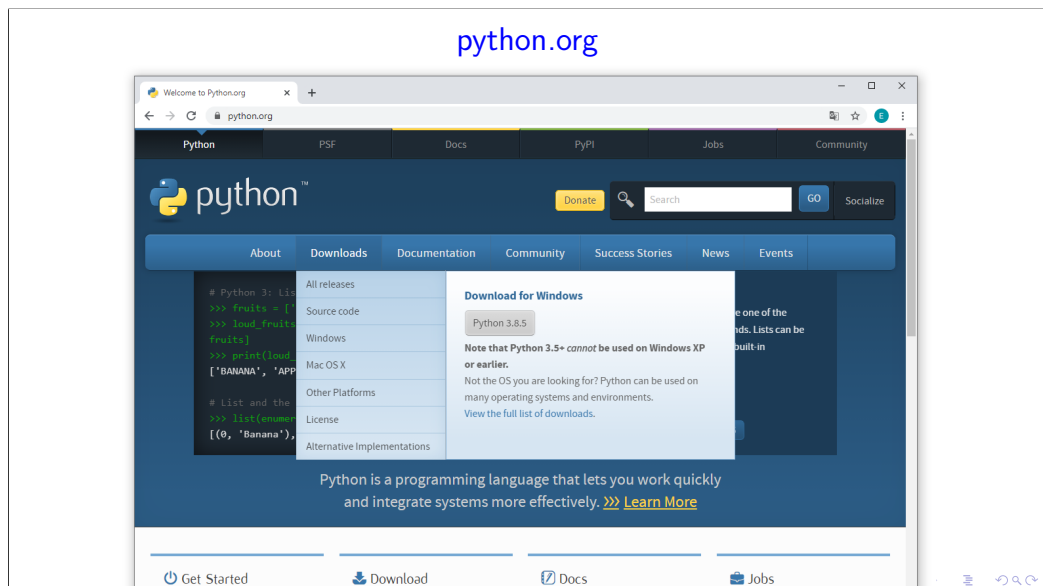
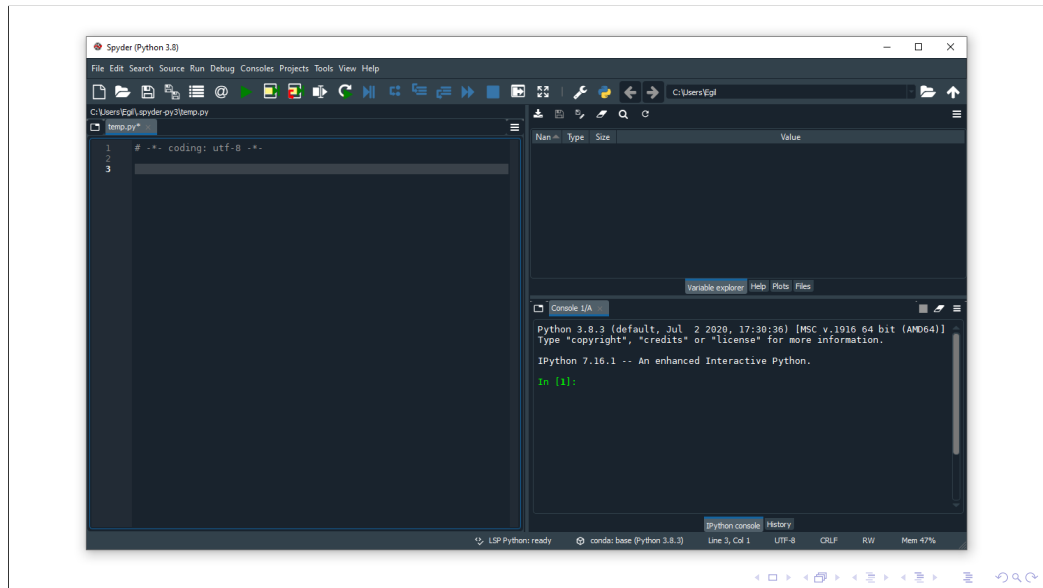


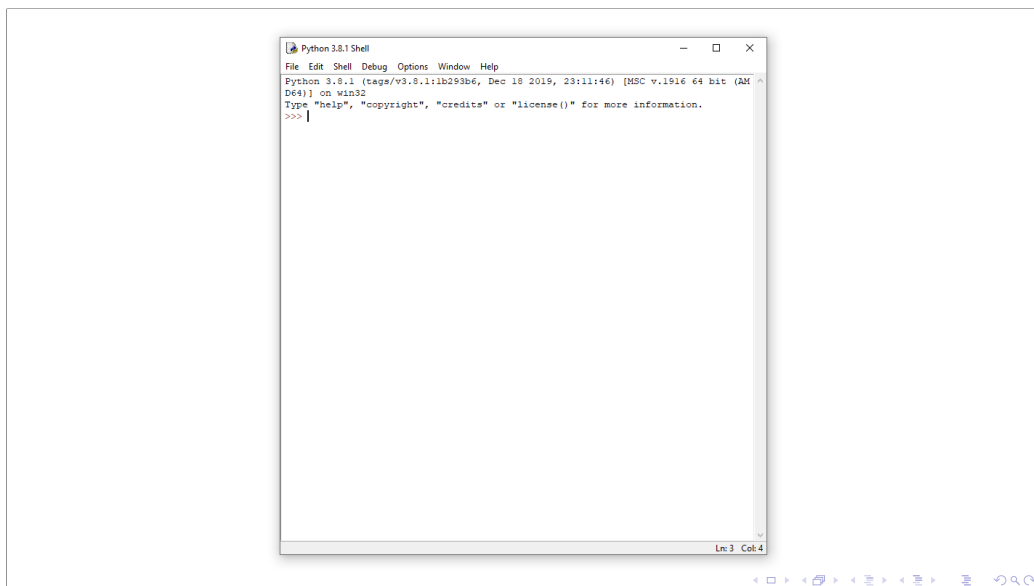
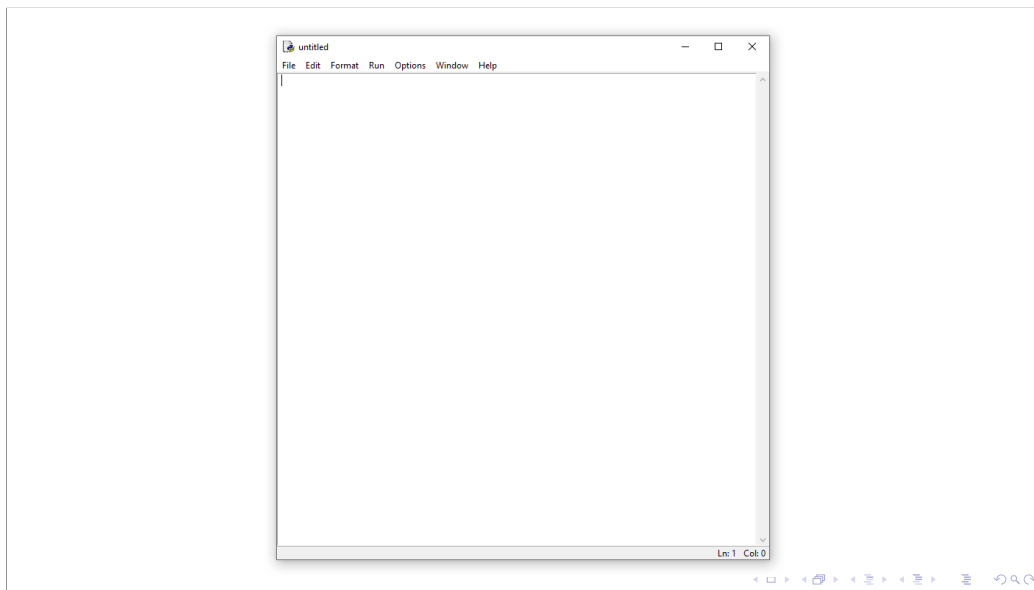
## Tillegg E

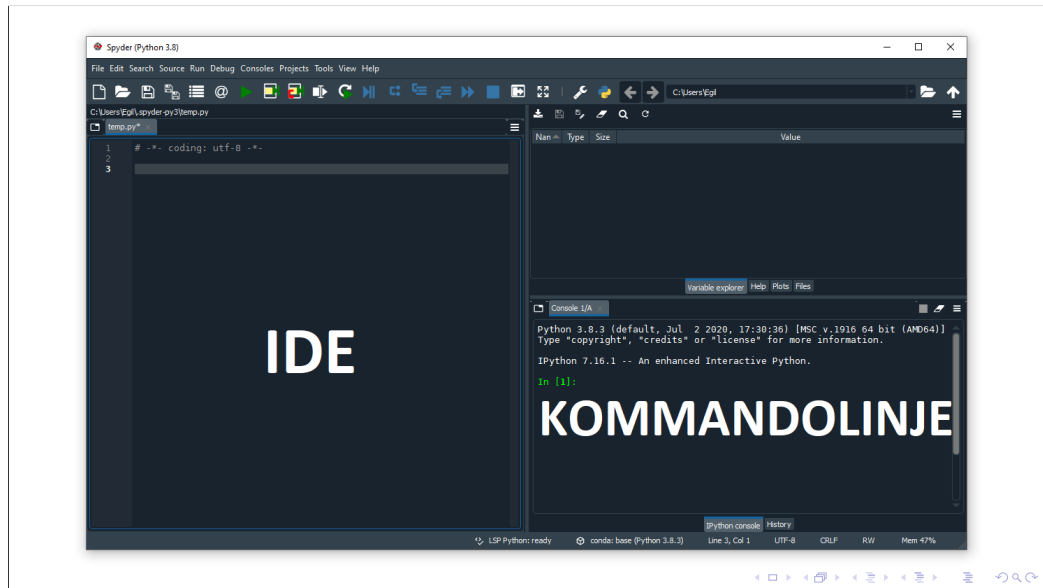
### Lysbilder brukt i opplegget











	Navn	Symbol	Python syntaks
1	Addisjon	+	+
2	Subtraksjon	-	-
3	Multiplikasjon	*	*
4	Divisjon flyttall	:	/
5	Divisjon heltall	:	//
6	Rest	:	%
7	Eksponent	:	**
8	Mindre enn	<	<
9	Mindre enn eller lik	≤	<=
10	Er lik	=	==
11	Større enn eller lik	≥	>=
12	Større enn	>	>
14	Ulik	≠	!=

## Vårt første program

1. Vi skriver i IDE-vinduet:
2. `print("Hello, World!")`
3. Lagre og kjør programmet



1. Skriv et program som printer differansen mellom to tall en bruker har oppgitt.
2. Skriv et program som printer summen av to tall en bruker har oppgitt.
3. Et partall er et heltall som har null i rest når det deles med 2; hva gjør programmet under? Skriv programmet inn og test det ut.

```
1 a = eval(input("Oppgi et heltall: "))  
2 print(a%2==0)
```

4. Skriv et program som finner ut om  $a - b$  er større enn 0.  $a$  og  $b$  skal være bruker-input.
5. Skriv en kode der en bruker oppgir høyde og bredde til en trekant; programmet skal printe ut arealet til trekanten.



$$\begin{aligned}2x + 11y &= 4 \\3x + 14y &= 1\end{aligned}$$



$$\begin{aligned}2x + 11y &= 4 \\3x + 14y &= 1\end{aligned}$$

$$\begin{aligned}ax + by &= c \\dx + ey &= f\end{aligned}$$





Tillegg F

Intervjuguide lærer



## Intervjuguide for intervju med lærer

### Rammesetting

- Ønsk velkommen
- Informasjon om intervjuet/samtalen
  - Formål med samtalen
  - Forklar hva intervjuet skal brukes til, informasjon om rettigheter (anonymitet, taushetsplikt, mulighet til å trekke seg)
  - Informere om og få samtykke for lydopptak
  - Spør om respondenten har spørsmål og om noe er uklart
- Start opptak

### Intervjuet

Formålet med intervjuet er at det skal være et supplement til egne observasjoner fra undervisningsøkten. Disse observasjonene vil fokusere på hvordan programmering og algoritmisk tenking passer inn i matematikkundervisning. Observasjoner, kommentarer eller lignende som retter seg inn mot dette temaet kan derfor bli brakt inn i intervjudelen.

- Masteroppgaven jeg skriver ser på hvordan programmering kan innlemmes i matematikkundervisningen, har du noen tanker om dette?
- Har du noen observasjoner fra undervisningsøkten relatert til tema som du vil dele?
- Har du noen observasjoner fra andre undervisningsøkter relatert til tema som du vil dele?
- Med ny læreplan blir det et sterkere fokus rettet på digital kompetanse;
  - Hva legger du i begrepene *digital kompetanse*, *algoritmisk tenking* og *programmering*?
  - Hvordan har du som lærer forberedt deg på å undervise den nye læreplanen? Eks. kurs, lest seg opp, lært programmering osv.
  - Hvilke tanker har du gjort deg om fokuset på digital kompetanse i læreplanen?
- Hvilke effekt kan programmering og algoritmisk tenking ha på matematikkundervisningen? (positiv/negativ/nøytral? Hva mener du elevene står igjen med?)
- Har du noe mer du vil kommentere om fra undervisningsøkten?

**Oppsummering**

- Oppsummer de viktigste funnene/poengene fra samtalen
- Spør om jeg har forstått korrekt
- Respondenten får kommentere dersom de mener det er noe mer som er relevant
- Takk for deltagelse
- Stopp opptak

Tillegg G

Intervjuguide elev



## Intervjuguide for intervju med elev

### Rammesetting

- Ønsk velkommen
- Informasjon om intervjuet/samtalen
- Formål med samtalen - Husk at dette ikke er en test
- Forklar hva intervjuet skal brukes til, informasjon om rettigheter (anonymitet, taushetsplikt, mulighet til å trekke seg)
- Informere om og få samtykke for lydopptak
- Spør om respondenten har spørsmål og om noe er uklart
- Start opptak

### Intervjuet

- Hvor motivert er du i matematikkfaget?
- Spør eleven om tidligere erfaring med programmering. Har du hatt programmering i skolesammenheng tidligere?
- Finn ut hva eleven husker fra opplegget.
- Spør eleven om h\*n tror programmering er nyttig som verktøy i matematikk? (på linje med kalkulator og/eller GeoGebra)
- Spør eleven om h\*n tror at h\*n vil bruke programmering som verktøy i matematikk, i tillegg til når det undervises direkte?
- I så fall hvorfor/hvorfor ikke?
- Er det noe eleven føler at blir enklere ved å bruke programmering?
- Er det noe eleven synes var/er vanskelig med programmering?
- Gi eleven en programmeringsoppgave (beskrevet senere).
- Gi eleven en programmeringsoppgave (beskrevet senere).
- I oppgavene, Eleven skal forklare muntlig, men kan også tegne/skrive ned hva h\*n tenker. Disse notatene tar jeg vare på.

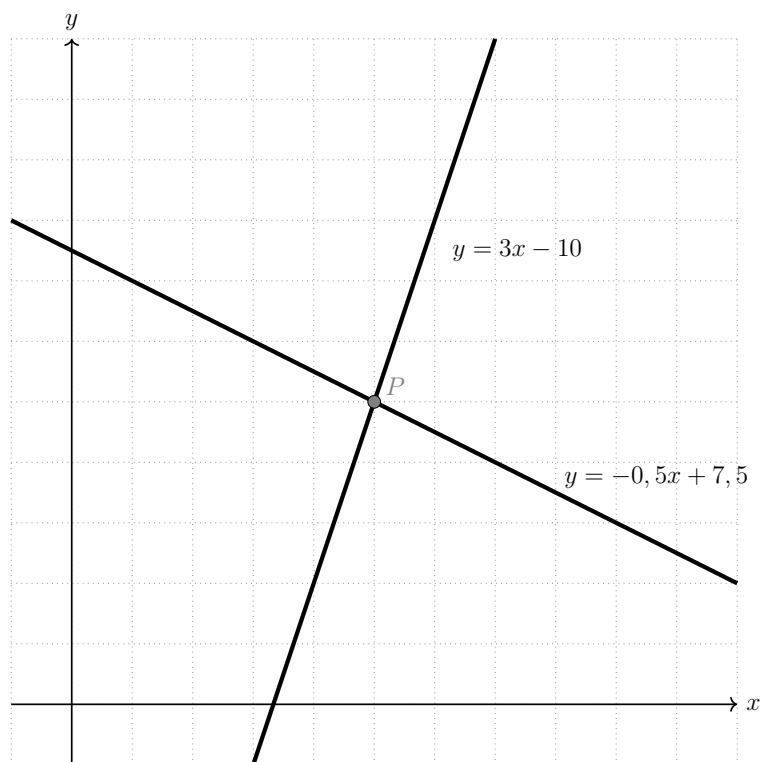
**Oppsummering**

- Oppsummer de viktigste funnene/poengene fra samtalen
- Spør om jeg har forstått korrekt
- Respondenten får kommentere dersom de mener det er noe mer som er relevant
- Takk for deltagelse
- Stopp opptak



## Oppgave 1

Forklar hvordan du vil gå frem for å lage et program som finner punktet der to linjer krysser hverandre. I eksempelet under vises to linjer som krysser hverandre, programmet du beskriver skal klare å finne punktet  $P$  i dette eksempelet.

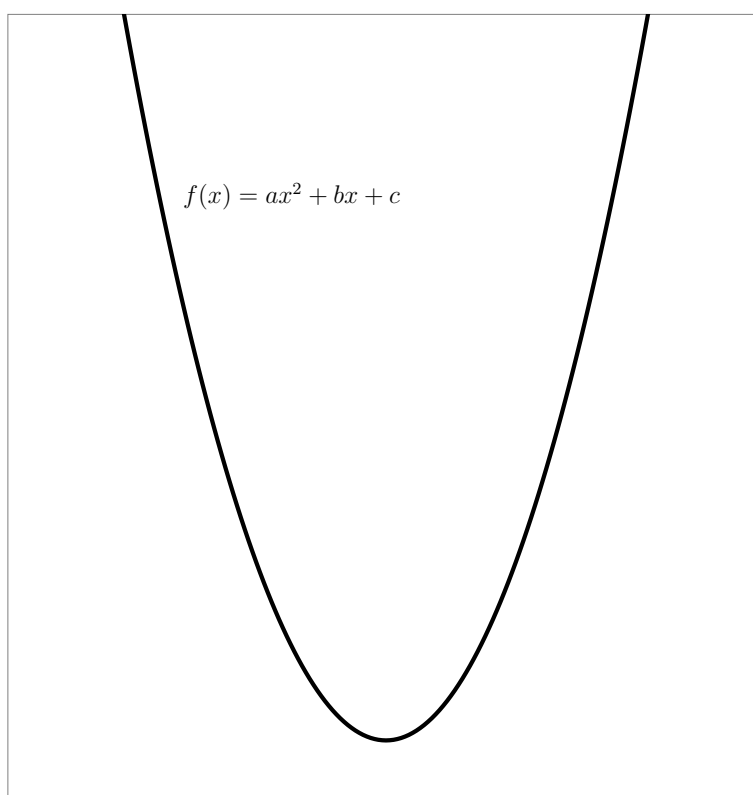


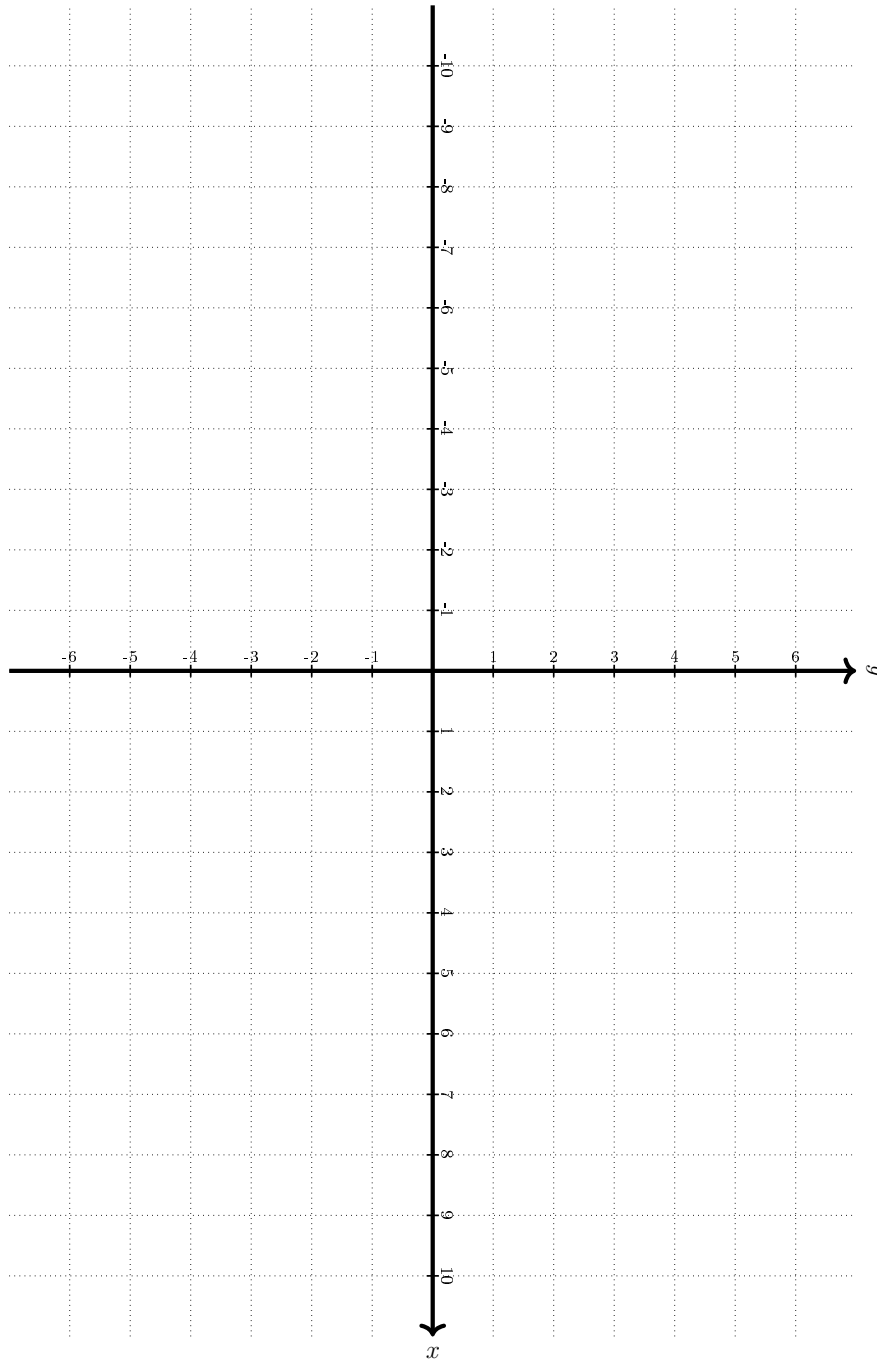
## Oppgave 2

Vi kan finne ut når funksjonen  $f(x) = ax^2 + bx + c$  har nullpunkt ved å bruke formlene

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{og} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Skriv et program som tar inn  $f(x)$  og som regner ut og printer  $x_1$  og  $x_2$ .





$$f(x) = x^2 + 3x + 1$$

$$f(x) = x^2 + 2x + 2$$

$$f(x) = x^2 + 4x + 4$$