

Ellen-Marie Ribe Kristiansen

"Det ligner på åttegangen"

En designstudie av fire elevers bruk av programmering som instrument i arbeid med tallfølger med multiplikativ struktur

Masteroppgave i Grunnskolelærerutdanning 1-7

Veileder: Oda Tingstad Burheim, Yvonne Grimeland, Benedikte Grimeland og Torunn Klemp

November 2023

Ellen-Marie Ribe Kristiansen

"Det ligner på åttegangen"

En designstudie av fire elevers bruk av programmering som instrument i arbeid med tallfølger med multiplikativ struktur

Masteroppgave i Grunnskolelærerutdanning 1-7
Veileder: Oda Tingstad Burheim, Yvonne Grimeland, Benedikte Grimeland og Torunn Klemp
November 2023

Norges teknisk-naturvitenskapelige universitet
Fakultet for samfunns- og utdanningsvitenskap
Institutt for lærerutdanning



Kunnskap for en bedre verden

Sammendrag

Dette masterprosjektet er en kvalitativ designstudie som undersøker problemstillingen: *Hvordan kan programmering som instrument bidra til elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur for en gruppe elever på 4. trinn?* For å besvare denne problemstillingen laget jeg et undervisningsopplegg hvor elevene skulle jobbe med tallfølger med multiplikativ struktur som la opp til bruk av programmering som verktøy. I utformingen av oppgaver valgte jeg å bruke programmeringsverktøyet *micro:bit*, med det tilhørende blokkbaserte programmeringsspråket *MakeCode*. Undervisningsopplegget ble gjennomført med en gruppe på fire elever i løpet av to undervisningsøkter. Disse undervisningsøktene ble tatt video- og lydopptak av. Elevene jobbet individuelt med oppgavene. De fleste oppgavene ble løst på ark, med programmeringen som et verktøy i løsningen, mens noen oppgaver skulle løses ved hjelp av blokkprogrammering på en Chromebook.

Datamaterialet består i hovedsak av videoopptak fra elevenes arbeid med undervisningsopplegget, som senere ble transkribert. Egne observasjonsnotater er også en del av datamaterialet, men disse er mindre brukt i analysen. Jeg var deltakende observatør i gjennomføringen av undervisningsopplegget, i begge undervisningsøktene. Analysen i denne studien baserer seg på en induktiv tilnærming, hvor jeg har gjennomført en åpen koding av det innsamlede datamaterialet. Funnene fra analysen blir drøftet ut fra sentrale teorier innenfor matematikdidaktikk. Mest sentral er Vygotskys (1978) sosiokulturelle læringssyn med fokus på medierende artefakter og Rabardels (1995; 2001) teori om skapelsen av et instrument. Drøftingen benytter seg også av teori om multiplikasjon, tidlig algebra og programmering.

Funn fra analysen viser at programmeringen i liten grad bidro som et instrument i elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur. I drøftingen kommer det frem flere måter hvor programmeringen ser ut til å være et hinder for elevenes utvikling av kunnskap. Elevene brukte i stor grad gjentatt addisjon som strategi for løsningen av oppgavene. Gjentatt addisjon skaper utfordringer for elevene når de skal løse multiplikasjon mellom tall som er under 1 (Anghileri, 2006). Elevenes bruk av gjentatt addisjon er altså et potensielt hinder i deres videre arbeid med multiplikasjon.

Dette forskningsprosjektet er gjennomført som en del av forskningsprosjektet LAB-Ted. Det har gjort at jeg har vært en del av samme praksisgruppe, med samme veiledere/oppfølgingslærere siden starten av tredje studieår. I tillegg har jeg hatt praksis ved samme skole siden starten av tredje studieår. Jeg hadde praksis i samme klasse flere ganger, og jeg kjente derfor medstudenter, elevene og veiledere bedre enn det som er vanlig i et masterprosjekt. Deltakelse i LAB-Ted betyr også at fokuset for masteroppgaven er valgt i samarbeid med både NTNU og praksisskolen.

Abstract

This master's project is a qualitative design study that examines the issue: *How can programming as an instrument contribute to students' development of knowledge about number sequences with a multiplicative structure for a group of students in the 4th grade?* To answer this problem, I created a teaching plan where the students had to work with number sequences with a multiplicative structure that facilitate the use of programming as a tool. In the design of the tasks, I chose to use the programming tool *micro:bit*, with the corresponding block-based programming language *MakeCode*. The teaching program was carried out with a group of four students during two teaching sessions. These teaching sessions were video- and audio-recorded. The students worked individually on the tasks. Most of the tasks were solved on paper, with programming as a tool in the solution, while other tasks were to be solved using block programming on a Chromebook.

The data material mainly consists of video recordings from the pupils' work with the teaching programme, that was later transcribed. My own observation notes are also a part of the data material, but these are less used in the analysis. I was a participant observer in the implementation of the teaching plan, in both teaching sessions. The analysis in this study is based on an inductive approach, where I have carried out an open coding of the collected data. The findings from the analysis are discussed based on central theories within mathematics didactics. Most central is Vygotsky's (1978) sociocultural learning view with a focus on mediating artefacts and Rabardel's (1995; 2001) theory on the creation of an instrument. The discussion also makes use of the theory of multiplication, early algebra, and programming.

Findings from the analysis show that the programming contributed to a small extent as an instrument in the students' development of knowledge about number sequences with a multiplicative structure. In the discussion, several ways emerge in which the programming appears to be an obstacle to the pupils' development of knowledge. The students largely used repeated addition as a strategy for solving the tasks. Repeated addition creates challenges for students when they have to solve multiplication between numbers that are less than 1 (Anghileri, 2006). The students' use of repeated addition is therefore a potential obstacle in their further work with multiplication.

This research project has been carried out as part of the research project LAB-Ted. This has meant that I have been part of the same practice group, with the same supervisors/follow-up teachers since the start of the third year of study. In addition, I have had practice at the same school since the start of the third year of study. I had practice in the same class several times, and I therefore knew my fellow students of the practice group, the students in the school, and the supervisors better than is usual in a master's project. Participation in LAB-Ted also means that the focus for the master's thesis has been chosen in collaboration with both NTNU and the practice school.

Forord

Denne masteroppgaven markerer slutten på fem og et halvt år på lærerstudiet. Tenk at det er over! Selv om det tok litt lengere tid på grunn av sykdom, er jeg endelig ferdig med oppgaven. Det har vært veldig lærerikt og spennende å fordype seg i programmering. Oppgaven følte lenge kjempestor og helt u håndgripelig, men etter mye prøving og feiling på veien, har jeg med mye støtte kommet i mål på et vis.

Det å være deltaker i forskningsprosjektet LAB-Ted har kommet med mange fordeler. Den viktigste av dem er de fire veilederne mine, Oda Tingstad Burheim, Yvonne Grimeland, Benedikte Grimeland og Torunn Klemp. Takk for gode konstruktive tilbakemeldinger gjennom hele prosessen, og god støtte når oppgaven så helt uoverkommelig ut. Det hadde ikke blitt noen ferdig oppgave uten dere. Jeg setter også stor pris på det samarbeidet vi har hatt gjennom prosessen med både veiledere, praksislærer og ikke minst medstudenter. Takk til elevene som har deltatt i studien.

Den største takken går til nærmeste familie og kjæresten min, som har vært viktig støtte i tunge stunder. Spesielt takk til mamma, som har lest gjennom oppgaven, og gitt tips til korrektur.

Alta, november 2023
Ellen-Marie Kristiansen

Innhold

Sammendrag	v
Abstract	vi
Figurer	x
Tabeller	x
1 Innledning	1
1.1 Bakgrunn	1
1.2 Forskningsprosjektet LAB-Ted	2
1.3 Min studie	2
1.4 Disposisjon	3
2 Teori	4
2.1 Læring gjennom deltakelse i sosiokulturell læringsteori.....	4
2.1.1 Samhandling med digitale verktøy i den proksimale utviklingszone.....	4
2.2 Artefakter og mediering	5
2.2.1 Instrumentell skapelse.....	8
2.3 Programmering.....	9
2.3.1 MakeCode og micro:bit	11
2.3.2 Scratch	15
2.3.3 <i>Unplugged</i> programmeringsaktiviteter	16
2.3.4 Tidligere forskning på programmering i matematikkundervisning	17
2.4 Tidlig algebra	17
2.4.1 Multiplikasjon	18
2.4.2 Mønster i matematikk	19
2.4.3 Mønster i multiplikasjon	20
2.4.4 Mønster i tallfølger	21
2.4.5 Multiplikative tallfakta.....	21
3 Metode	22
3.1 Forskningsmetode.....	22
3.1.1 Designstudie.....	22
3.2 Kontekst	24
3.2.1 Deltakelse i LAB-Ted	25
3.2.2 Utvalg.....	25
3.3 Metode for datainnsamling	25
3.3.1 Oversikt over innsamlet datamateriale	26
3.3.2 Observasjon	26
3.3.3 Åpen observasjon.....	27
3.3.4 Deltakende observasjon	27
3.3.5 Observasjonsdata som video	28

3.4 Utforming av oppgavene til datainnsamlingen.....	28
3.4.1 Beskrivelse av oppgavene som ble brukt.....	28
3.5 Analysemetode.....	32
3.5.1 Transkripsjon.....	33
3.5.2 Inndeling i sekvenser.....	34
3.5.3 Åpen koding.....	34
3.6 Etske betraktninger.....	36
3.6.1 Troverdighet og transparens.....	38
4 Resultater.....	39
4.1 Bruk av programmering.....	39
4.1.1 Bruker programmeringen til å utforske store tall.....	39
4.1.2 Bruker egenlagde koder som ikke var hensiktsmessig i løsning av oppgavene.....	40
4.1.3 Bruker programmeringen til å sjekke om det de tenkte var riktig.....	41
4.1.4 Utfordringer med å bruke programmet i arbeid med store tall.....	42
4.1.5 Utfordringer med forståelsen av programmeringen som hinder for løsningen av oppgavene.....	43
4.2 Arbeid med multiplikasjon.....	45
4.2.1 Bruk av gjentatt addisjon.....	45
4.2.2 Elevene bruker oppgavearket eller programmeringen til å løse oppgavene på en annen måte enn multiplikasjon.....	46
4.2.3 Elevene knytter ikke arbeidet med oppgavene til multiplikasjon.....	48
5 Drøfting.....	49
5.1 Er programmering som artefakt til hjelp eller hindring i arbeidet med tallfølger med multiplikativ struktur?.....	49
5.2 Elevenes valg av løsningsstrategi.....	51
5.3 Metodediskusjon.....	52
6 Konklusjon.....	54
Referanser.....	55
Vedlegg.....	58

Figurer

Figur 1: Stimulus-mediering-respons	5
Figur 2: Prosessen med å skape et instrument	9
Figur 3: Eksempel på blokker som passer sammen i MakeCode	10
Figur 4: Framsiden (til venstre) og baksiden (til høyre) av en fysisk micro:bit	11
Figur 5: Den offisielle editoren til micro:bit	12
Figur 6: Animasjonen av micro:bit med knappene SHAKE og A+B	13
Figur 7: Startblokk, handlingsblokk og verdiblokk	13
Figur 8: Noen basisblokker	14
Figur 9: Noen inndatablokker	14
Figur 10: Variabelblokkene	15
Figur 11: Scratch.....	16
Figur 12: programmet i oppgave 1.....	29
Figur 13: Koden i oppgave 10	32
Figur 14: Bjørns kode	41

Tabeller

Tabell 1: Oversikt over innsamlet datamateriale	26
Tabell 2: Kategorier for funn i analysen	36

1 Innledning

1.1 Bakgrunn

Programmering i skolen er relativt nytt, og forskningen innenfor emnet er derfor begrenset (Forsström & Kaufman, 2018, s. 29). Framkomsten av digitale verktøy innenfor utdanningspraksiser endrer måten vi lærer og innhenter kunnskap om nye ting (Säljö, 2010, hentet Hundeland et al., 2014, s. 209). Det interessante innenfor den digitale verdenen er evnen vi har til å bruke digitale verktøy på produktive og innsiktsfulle måter og på måter som passer i gitte situasjoner. Læring handler om å mestre bruken av redskapene og bruke dem på passende måter når man samhandler med IKT (informasjons- og kommunikasjonsteknologi)- applikasjoner. På denne måten er det å bli kjent med digitale verktøy viktig for elevene ut fra et utdanningsperspektiv. For å bli kompetente deltakere i samfunnet, som i økende grad er sofistikert og spesialisert, trenger de kommende generasjonene ferdigheter innenfor digitale redskaper og de mulighetene og begrensningene som finnes i disse redskapene (Hundeland et al., 2014, s. 209). Også Forsström og Kaufman (2018, s.18) understreker at programmeringsferdigheter blir mer og mer viktig for elever å lære i det teknologiske samfunnet vi lever i. Undervisningssektoren må gi elever kompetansen til å mestre og lage egne digitale teknologier og å forberede elevene på fremtiden. Derfor er programmering og koding både i formell og uformell setting essensielt i skolen (Forsström & Kaufman, 2018, s. 19).

Plowman og Stephen (2003) mener at bruken av IKT i utdanningspraksiser er et verdifullt innslag, eller et ufarlig tillegg til eksisterende ressurser (Hundeland, 2014, s. 209). De mener at bruken av disse verktøyene ikke endrer praksisen. Andre forskere har funnet at digitale verktøy er effektive til å bedre elevens læring innenfor matematikk og problemløsning i alderen 3-6 år (Lieberman et al., 2009, funnet i Hundeland, 2014, s. 209).

Programmering er ofte sett på som nødvendig kompetanse for å lære, arbeide og leve i dagens og morgendagens samfunn (Senter for IKT i utdanningen, 2016, s. 10). Samfunnet vi lever i blir stadig mer digitalisert, noe som fører til et stort behov for yrker der programmering står sentralt (Senter for IKT i utdanningen, 2016, s. 11). Det trengs kunnskap og ferdigheter i programmering for at vi som samfunn kan utvikle teknologi videre. I tillegg til at programmeringsferdigheter og kunnskap om programmering er nødvendig i yrker hvor det skjer teknologiutvikling, trenger også resten av samfunnet en viss forståelse for programmering for å kunne bruke den teknologien som utvikles (Senter for IKT i utdanningen, 2016, s. 11).

Det har i de siste årene utviklet seg en internasjonal bevegelse for å fremme programmering i skolen (Senter for IKT i utdanningen, 2016, s. 6). Noen land har innført programmering som et eget fag i skolen, mens andre land har gjort programmering til et tverrfaglig tema på tvers av allerede eksisterende fag (Kunnskapsdepartementet, 2019). I den norske læreplanen ble programmering innført som et tverrfaglig tema ved skiftet til *læreplanverket for kunnskapsløftet 2020* (Kunnskapsdepartementet, 2019). Programmering er vektlagt i kompetansemål og kjerneelementer i noen fag, blant annet i matematikkfaget (Kunnskapsdepartementet, 2019).

Programmering er en del av læreplanen, og lærere er derfor lovpålagt å undervise programmering til elevene. Læreplanen sier imidlertid lite om på hvordan man skal undervise programmeringen, og det er derfor opp til hver enkelt lærer. Det kan være utfordrende å vite hvordan programmeringen henger sammen med de andre læreplanmålene. Forsström og Kaufmann utførte en litteraturstudie innenfor temaet programmering i matematikkundervisning i 2018 som blant annet konkluderte med at det trengs mer forskning om hvordan programmering kan brukes i sammenheng med annen matematikk. Forskning på hvordan programmering kan brukes i sammenheng med annen matematikk er nødvendig for at programmeringen ikke kun blir et isolert emne, men noe som henger sammen med resten av matematikkundervisningen. Det finnes en del forskning om sammenhengen mellom programmering og geometriundervisningen, men det er lite forskning som ser på sammenhengen mellom programmering og andre matematiske tema (Forsström & Kaufmann, 2018, s. 30). Også Sarama og Clements (2004) så et behov for å forske for å identifisere rollen til digitale redskaper, og deres bidrag til matematikklæring (referert i Hundeland et al., 2014, s. 207).

1.2 Forskningsprosjektet LAB-Ted

Denne studien er gjennomført som en del av forskningsprosjektet *Learning, Assessment and Boundry crossing in Teacher education (LAB-Ted)*. Et fokus for prosjektet er å ha et trepartssamarbeid for oppgaveskriving i lærerutdanningen (Norges teknisk-naturvitenskapelige universitet, u.å.). Dette trepartssamarbeidet er mellom studentene, veiledere fra universitetet og praksisskolen, representert ved rektor og praksislærer. Dette samarbeidet gjør at fokuset på studentenes masteroppgaver er valgt i samarbeid med veiledere fra universitetet og praksislærer og rektor i praksisskolen. På den måten er målet at masteroppgavene skal være relevante for lærerutdanningen på universitetet, for praksis i skolene, og for studentene selv.

Jeg har gjennom deltakelse i LAB-Ted vært del av samme praksisgruppe de tre siste årene av studiet mitt. Praksisperiodene har i disse årene foregått ved samme skole, med de samme oppfølgingslærerne. Både oppfølgingslærerne fra NTNU og skolen, representert ved rektor og praksislærer, var deltaker i LAB-Ted. På denne måten har vi fått veiledning av både veiledere fra universitetet og praksislærer. I praksisperioden og ved innsamling av datamateriale fikk vi veiledning av praksislærer, som også hadde innsikt i masterprosjektene våre. I selve skrivearbeidet hadde vi gruppeveiledning med alle fire matematikkstudenter og alle fire veiledere. Alle fire masteroppgavene som er skrevet av praksisgruppa har felles overordnet tema innen matematikkdiraktikk.

1.3 Min studie

I dette masterprosjektet vil jeg undersøke hvordan programmering kan undervises i sammenheng med andre matematiske tema, slik som Forsström og Kaufmann (2018) foreslår. Sarama og Clemets (2004) fant i sin studie ut at det dukker opp muligheter for matematikklæring når elever er engasjert i datamaskinprogramvare (referert i Hundeland, 2014, s. 209). Disse mulighetene for matematikklæring som ligger i elevenes bruk av datamaskinprogramvare vil jeg se nærmere på i dette masterprosjektet. Forskningen i denne masteren ble gjort på et 4. trinn. Derfor så jeg på kompetansemålene for 4. trinn i læreplanen for matematikk (Kunnskapsdepartementet, 2019) for å finne hvilke matematiske tema som kunne være relevant å se på i sammenheng med programmering. Et av kompetansemålene etter 4. trinn er «utforske

og beskrive strukturer og mønstre i lek og spill». Nesten all matematikk er ifølge Mulligan og Mitchelmore (2013, s.30) basert på mønster og struktur. Jeg valgte derfor å fokusere på mønster i matematikkfaget. Mønster i matematikk er et vidt felt, siden nesten all matematikk er basert på mønster og struktur. Mønster i matematikk er et vidt område, så jeg måtte snevre temaet inn. Jeg valgte å fokusere på mønster i multiplikasjon. For å se på mønster i multiplikasjon har jeg valgt å se på multiplikasjon som tallfølger. En av grunnene til at jeg valgte tallfølger i multiplikasjon var at arbeid med tallfølger er noe som er gjennomførbart i sammenheng med programmering.

I denne masterstudien har jeg valgt å bruke micro:bit med det blokkbaserte programmeringsspråket *MakeCode* i matematikkundervisning på 4. trinn. Forskningsspørsmålet mitt er: *Hvordan kan programmering som instrument bidra til elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur for en gruppe elever på 4. trinn?*

Denne masterstudien er en designstudie hvor jeg har utformet et oppgavesett som en gruppe med elever skulle løse. Det viktigste argumentet for å utføre en designstudie er å øke forskningens relevans for både praksisfeltet og forskningsfeltet (Akker et al., 2006, s. 2). Utdanningsforskning har lenge blitt kritisert for at forskningen ofte har liten sammenheng med praksis (Akker et al., 2006, s. 2). Designstudier kan bidra til større praksisrelevans. Jeg ville at min masteroppgave skulle ha mest mulig relevans for praksisfeltet, og at min masteroppgave skulle henge mest mulig sammen med mitt framtidige arbeid som lærer. Derfor valgte jeg å gjennomføre en designstudie. Oppgavene jeg ga til elevene omhandler tallfølger med multiplikativ struktur og programmering i *MakeCode*. På denne måten vil jeg se på hvordan elevene brukte programmeringen som en instrument i arbeid med tallfølger med multiplikativ struktur. For å kunne diskutere på hvilken måte programmering som instrument kan bidra til elevenes kunnskap i multiplikasjon skal jeg bruke Rabardels teori om *instrumentell skapelse (instrumental genesis)*. I tillegg vil jeg bruke teori om tidlig algebra og multiplikasjon.

1.4 Disposisjon

Teorikapittelet inneholder en beskrivelse av læring som deltakelse i sosiokulturell læringsteori. Så blir artefakter og mediering forklart, med fokus på Rabardels teori om instrumentell skapelse. Videre inneholder teorikapittelet teori om programmering og tidlig algebra. Metodekapittelet inneholder en beskrivelse av kontekst for studien, forskningsmetode, metode for datainnsamling, utforming av oppgavene til datainnsamlingen, analysemetode og etiske betraktninger.

I resultatkapittelet presenteres funn fra analysen. Funnene tar utgangspunkt i forskningsspørsmålet og er delt inn i hovedkategoriene *Bruk av programmering* og *Arbeid med multiplikasjon*. I drøftingen blir programmering som artefakt i arbeid med tallfølger med multiplikativ struktur drøftet. I tillegg blir elevenes valg av løsningsstrategi drøftet, før refleksjoner rundt studiens begrensninger blir presentert i *Metodediskusjon*. Til slutt oppsummeres de viktigste funnene i dette masterprosjektet i *Konklusjon*.

2 Teori

Først i teorien presenteres læring gjennom deltakelse i sosiokulturell læringsteori. Sosiokulturell læringsteori er viktig for min oppgave fordi det danner det teoretiske grunnlaget som jeg bruker i mitt masterprosjekt. Videre presenteres artefakter og mediering, med fokus på instrumentell skapelse. Dette er den viktigste teorien for mitt prosjekt, da det er utgangspunktet for drøftingen av om elevene har brukt programmeringen som et instrument i arbeidet med oppgavene.

Programmering, mer spesifikt blokkprogrammering, blir skrevet om i delkapittel 2.3. Blokkprogrammeringsspråket MakeCode blir forklart, sammen med den fysiske programmerbare databrikken micro:bit. MakeCode og micro:bit blir brukt i oppgavene som elevene blir gitt i undervisningsopplegget.

Til slutt i teorikapittelet blir tidlig algebra forklart. Spesifikt forklares multiplikasjon, mønster i matematikk, mønster i multiplikasjon, mønster i tallfølger og multiplikative tallfakta. Denne delen er med for å kunne brukes til å forklare i hvilken grad elevene utvikler sin kunnskap om tallfølger med multiplikativ struktur.

2.1 Læring gjennom deltakelse i sosiokulturell læringsteori

I sosiokulturelt perspektiv ses læring som en situert og sosial prosess hvor individer tilegner seg konsepter, redskaper og handlinger gjennom samhandling og kommunikasjon (Hundeland et al., 2014, s.208). Sosiokulturelt perspektiv gir en linse som man kan bruke til å analysere læringsaktiviteter som innebærer digitale redskaper.

Tilegnelse (*Appropriation*) som en sosiokulturell metafor for læring handler om prosessen å ta noe som andre har, og gjøre det til sitt eget (Wertsch 1998, funnet i Hundeland et al., 2014 s. 208). For at denne prosessen skal kunne skje må individet delta i sosiale interaksjoner med andre. Sentralt i dette ligger kommunikasjon og bidrag knyttet til ideer og argumentasjon som essensielle elementer (Hundeland, 2014, s.208). Ideen om tilegnelse handler om at folk gjennom deltakelse endrer seg og i denne prosessen blir med forberedt til å delta i liknende aktiviteter. Gjennom å delta i en aktivitet, og å delta i dens meninger, er det uunngåelig at man bidrar kontinuerlig (Rogoff, 1995, s. 150). Bidragene kan være gjennom konkrete handlinger eller gjennom det å strekke seg for å forstå handlingene og ideene til andre. Deltakelse er på denne måten en tilegnelsesprosess. Tilegnelse er brukt som en kontrast til internalisering. I stedet for å se på prosessen som en internaliseringsprosess hvor noe statisk er tatt fra det eksterne til det interne, ser Rogoff (1995, s.151) på barnets aktive deltakelse i seg selv som en prosess hvor det tilpasser seg i møtet med en aktivitet. Som Wertsch og Stone (1979, s. 21, hentet Rogoff, 1995, s. 151) skrev det: «Proessen er produktet».

2.1.1 Samhandling med digitale verktøy i den proksimale utviklingszone

I sosiokulturelt syn på læring er den proksimale utviklingssonen sammenhengende mellom læring som tilegnelse og redskapers medierende rolle. Vygostky (1978) argumenterer at dersom et barn skal lære noe, må interaksjonen med mer dyktige jevnaldrende skje innenfor barnets proksimale utviklingszone (Hundeland et al., 2014, s.211). Zonen for proksimal utvikling er forskjellen mellom det et barn klarer å gjøre alene uten hjelp, og det det kan gjøre i samhandling med voksne eller mer kompetente medelever.

Ideen om den proksimale utviklingssonen kan være et nyttig verktøy i analysen av barns samhandling med digitale redskaper. Den proksimale utviklingssonen som ide kan også

brukes til å analysere hvordan barn samhandler med digitale redskaper, og hvordan barn samhandler med mer kompetente andre rundt bruken av digitale redskaper (Hundeland et al., 2014, s. 211).

I samarbeid med den voksne blir barnet utsatt for resonnering og handlinger som de gradvis tilegner seg gjennom å bli kyndig til å bruke disse resonneringene og utføre disse handlingene selv. Ifølge Chaiklin (2003, funnet i Hundeland et al., 2014, s. 211) kan man analysere barns engasjement med IKT-redskaper ved å se på kvaliteten på samhandlingene innenfor den proksimale utviklingssonen som skjer mellom barnet og mer kompetente andre.

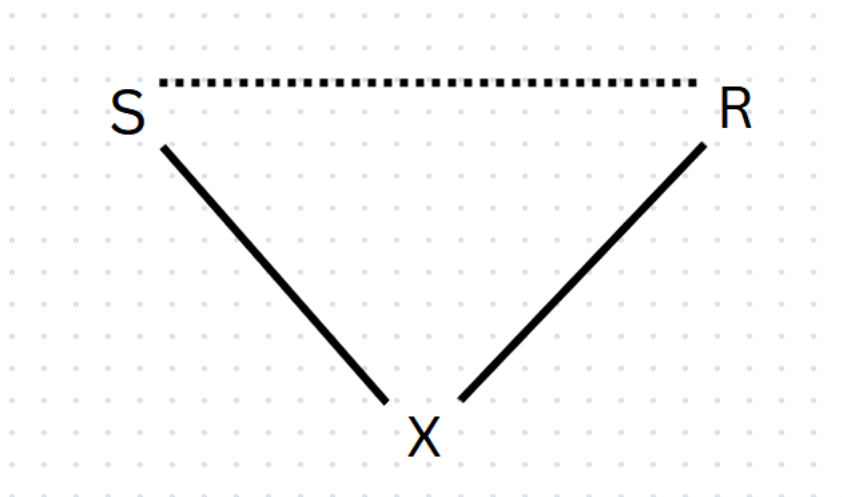
Sarama og Clements (2004) fant at barn ofte var mer kompetente enn programvaredesignerne antok at de skulle være ut fra alderen. Indikatorene for alder knyttet til ulike programmer er ikke lik den kompetansen barn har tilegnet seg i deres alder og i deres proksimale utviklingszone (Referert i Hundeland, 2014, s.211).

2.2 Artefakter og mediering

Sammenhengen mellom stimuli og respons er sentral i den behavioristiske tilnærmingen til læring. Vygotsky introduserte tegnet X i modellen for stimuli og respons. X-en er lagt til for å vise at vi mennesker kan erstatte stimuli med språklige symbol. Mennesker kan reflektere rundt hvilken handling som skal utføres, noe som gjør at responsen ikke alltid er direkte knyttet til stimuli. Dette kalles mediering og er ifølge Vygotsky grunnleggende for avanserte psykologiske prosesser (Imsen, 2005, s. 257). Dette kan illustreres gjennom trekanten i figur 1. Det er grunnleggende i et sosiokulturelt perspektiv at fysiske og intellektuelle/språklige redskaper medierer virkeligheten i konkrete virksomheter (Säljö, 2001, s. 83).

Begrepet mediering kommer fra det tyske ordet Vermittlung (formidling), og antyder at vi mennesker ikke er i ufortolket, direkte og umiddelbar kontakt med omverdenen. Vi mennesker håndterer heller omverdenen ved hjelp av fysiske og intellektuelle redskaper som er en integrert del av våre sosiale praksiser. Mediering er sentralt i den sosiokulturelle tradisjonen, og er kanskje det som skiller sosiokulturell teori mest fra andre teoretiske perspektiver. Mediering handler om at tenkningen vår og våre forestillingsverdener har vokst frem av kulturen vår og dens fysiske og intellektuelle redskaper (Säljö, 2001, s. 84).

Figur 1: Stimulus-mediering-respons



Dersom man skal forstå læring som en del av sosiale praksiser, hvor mennesket tar i bruk ulike redskaper, kan man ikke analysere disse redskapene for seg og deretter studere den «rene» menneskelige tenkningen. Dersom man tar bort redskapene og ser på tenkning eller læring i seg selv, mister man viktigheten disse redskapene har i disse prosessene, og man studerer temmelig hjelpeløse individer som er fratatt sine sosiokulturelle ressurser (Säljö, 2001, s.83).

Framkomsten av digitale verktøy innenfor utdanningspraksiser endrer måten vi lærer og innhenter kunnskap om nye ting (Säljö, 2010, hentet Hundeland et al., 2014, s. 209). Det interessante innenfor den digitale verdenen er evnen vi har til å bruke digitale verktøy på produktive og innsiktsfulle måter og på måter som passer i gitte situasjoner. Læring handler om å mestre bruken av redskapene og bruke dem på passende måter når man samhandler med IKT (informasjons- og kommunikasjonsteknologi)-applikasjoner. På denne måten er det å bli kjent med digitale verktøy viktig for elevene ut fra et utdanningsperspektiv. For å bli kompetente deltakere i samfunnet, som i økende grad er sofistikert og spesialisert, trenger de kommende generasjonene ferdigheter innenfor digitale redskaper og de mulighetene og begrensningene som finnes i disse redskapene (Hundeland, 2014, s. 209).

Vi mennesker er kulturelle vesener som i hverdagslige situasjoner interagerer med og tenker sammen med andre mennesker (Säljö, 2001, s. 17). Samtidig som vi mennesker er biologiske vesener, er vi også en del av en sosiokulturell virkelighet som gjør at vi kan oppnå ting langt utenfor våre biologiske begrensninger (Säljö, 2001, s. 17). Begrepene redskap og verktøy har en spesiell betydning innenfor sosiokulturell læringsteori. Redskaper eller verktøy er i sosiokulturell læringsteori de ressursene, både språklige (intellektuelle) og fysiske, som vi har tilgang til og benytter oss av for å forstå og handle i omverdenen (Säljö, 2001, s.21). Hvordan vi lærer er et spørsmål om hvordan vi tilegner oss ressurser og utføre praktiske oppgaver som er en del av vår kultur og omgivelsene som omgir oss (Säljö, 2001, s.21).

Mennesker utvikler og tar i bruk redskaper, som kan være fysiske eller mentale, for å heve seg over de forutsetningene som ligger i menneskets biologi (Säljö, 2001, s.30). I tillegg hever vi oss over disse forutsetningene gjennom å organisere oss i ulike former for kollektiv virksomhet. Mennesker utvikler ulike ressurser slik som fysiske redskaper, som for eksempel gravemaskin. På samme måte har vi laget redskaper til å utføre matematiske utregninger, slik at vi ikke trenger å regne ut matematiske stykker som ville krevd en veldig stor hjernekapasitet, noen ganger mere enn vi mennesker har. Begrepet *kultur* er en samlebetegnelse på slike ressurser som delvis finnes i individet, delvis i sosial interaksjon og delvis i den materielle omverdenen (Säljö, 2001, s.30). Kultur er altså samlingen av ideer, holdninger, kunnskaper og andre ressurser vi tilegner oss gjennom samhandling med omverdenen (Säljö, 2001, s.30). Innenfor kulturen inngår alle de fysiske redskapene som hverdagen er fylt av, disse kalles artefakter. Eksempler på artefakter er ulike verktøy og instrumenter for veiing, måling eller liknende, ulike former for informasjons- og kommunikasjonsteknologi, framkomstmidler og annet (s.30). Artefakter er altså gjenstander eller produkter som er utviklet av mennesker (Säljö, 2001, s. 31). Utviklingen av fysiske ressurser, artefakter, henger tett sammen med utviklingen av ideer og intellektuell kunnskap (Säljö, 2001, s.30).

Når man ser på utviklingen av artefakter er det viktig at artefakter ikke kan ses på bare som døde objekter (Säljö, 2001, s.82). Vi mennesker har bygd inn kunnskap, innsikt, konvensjoner og begreper, i apparater som vi omgås. Disse menneskelige kunnskapene,

innsiktene, konvensjonene og begrepene blir på denne måten noe vi samspiller med når vi handler (Leontiev 1981, hentet Säljö, 2001, s. 82). Vi lever altså i en kunstig virkelighet som er en verden av menneskelige gjenstander. På samme måte som at vi mennesker har laget og påvirket redskapene vi omgir oss med, påvirker også disse redskapene hvordan vi mennesker kan se på verden. For eksempel har vi utviklet kikkert, som hjelper oss til å se gjenstander eller detaljer som vi ikke kunne sett uten kikkerten som redskap (Säljö, 2001, s. 82). Det er brukeren av redskapet, i dette tilfellet kikkerten, som ser og som avgjør hva som er interessant å se av det som blir mulig å se ved hjelp av kikkerten. Allikevel er det viktig å ikke se bort ifra at vi mennesker gjør erfaringer ved hjelp av medierende redskaper når man skal forstå hvordan mennesker fungerer og lærer i sosiale praksiser. Andre eksempler på medierende redskaper er mikroskop, infrarød kikkert, radar, ekkolodd og lesebriller, som alle bidrar til at vi mennesker kan se eller få oversikt over områder eller detaljer som vi ellers ikke kunne sett. En tydelig illustrasjon av interaksjonen mellom menneskelige funksjoner og praktiske redskaper kan gis ved en rett stokk (Säljö, 2001, s. 82). For en synshemmet person som har lært seg å bruke stokken som et redskap for å forflytte seg, blir denne stokken er viktig redskap for å kunne orientere seg og interagere med omverdenen. Den synshemmede brukeren av stokken «leser» omgivelsene gjennom de signalene som de mottar via stokken, og kan vurdere hvordan underlag den møter på og omgivelsene rundt seg (Säljö, 2001, s. 83). Et viktig spørsmål knyttet til dette eksempelet er; «Hvor ligger tenkningen og registreringen?» (Säljö, 2001, s. 83). Tenkningen og registreringen finnes naturligvis i stor grad i hodet til individet som bruker redskapet, men ikke fullt og helt. Tenkningen til individet knyttes til omverdenen gjennom et redskap, i dette eksempelet en stokk. Stokken i seg selv har ingen interessante egenskaper, men i et system som består av en person som kan bruke redskapet og en stokk (eller en annen artefakt), kan stokken på en sofistisert måte bli en del av kompliserte sosiale og intellektuelle praksiser (Säljö, 2001, s.83).

Vellykkede artefakter fungerer ofte på en måte som gjør at man kan bruke artefaktet uten å forstå de underliggende teknikkene som artefaktet innebærer. Et eksempel på dette er datamaskinen, som veldig mange mennesker kan bruke, men som få vet hvordan faktisk fungerer. Det samme gjelder i arbeid med blokkprogrammering. Man trenger ikke å skjønne de underliggende teknikkene som finnes bak denne programmeringen, men bare hvordan man kan bruke denne programmeringen til det man ønsker å oppnå. Det å lære og utvikle seg i vår tid handler dermed mye om å utnytte kognitive ressurser som finnes i ulike artefakter i form av informasjon, prosedyrer og rutiner. Mediering skjer ikke bare ved bruk av artefakter og teknikk, men i stor grad gjennom ressursene som finnes i det aller viktigste medierende redskapet, språket (Säljö, 2001, s. 85). Vygotsky så på barns bruk av språket som det aller viktigste redskapet for barns utvikling (Vygotsky, 1978, s. 126). Språket bidrar til at barn får tilgang til deler av omverdenen som ellers ville vært utenfor rekkevidde. Gjennom språket kan man for eksempel forberede seg på fremtidige aktiviteter, gjennom blant annet planlegging (Vygotsky, 1978, s. 126). I denne masteroppgaven har jeg valgt å ikke fokusere på språk i analysen av elevenes arbeid. Språket danner allikevel grunnlaget for det som skjer i elevenes arbeid med oppgavene, gjennom oppgavens formulering, utsagn elevene kommer med, og lærerstøtten som elevene får i arbeidet med oppgavene.

Barns interaksjon og deltakelse i samhandling med andre er grunnleggende avhengig av bruken av redskaper. IKT- applikasjoner blir til digitale redskaper gjennom deres medierende funksjon. Redskapene kan mediere matematiske konsepter og ideer. Leont'ev skrev at redskapet medierer aktiviteten og dermed knytter mennesker både til

verden av objekter og til andre mennesker (1979, gjengitt i Hundeland et al., 2014, s. 210). Denne medierende rollen til digitale redskaper er synlig gjennom måten barna samhandler med hverandre gjennom kommunikasjon og med programvaren ved hjelp av de matematiske objektene som er innlemmet i programvaren.

2.2.1 Instrumentell skapelse

Instrumentell skapelse er innenfor forskningsfeltet for matematikkundervisning fokusert på utviklingen fra artefakt til instrument (Rabardel & Samurcay, 2001). Et instrument er en individuell mental konstruksjon av en artefakt. Det vil si at en artefakt blir et instrument når den blir brukt på hensiktsmessig måte i arbeidet med oppgaver. Prosessen fra artefakt til instrument kalles instrumentell skapelse, og innebærer at artefakten blir utviklet til et meningsfylt verktøy for den som skal bruke den (Rabardel & Samurcay, 2001).

Et subjekts instrument kan ses på som en sammensatt enhet som består av en artefaktdel og en skjemadel (Rabardel & Samurcay, 2001). Skjemadelen er en persons mentale skjema som endres i prosessen av utviklingen av instrumenter, altså en instrumentell skapelses-prosess. En persons skjema har en privat dimensjon og en sosial dimensjon. Den private dimensjonen er knyttet til det at skjemaer er dannet hos hvert enkelt individ. Men det eksisterer også en viktig sosial dimensjon. Det er fordi skjemaer utvikles delvis på grunnlag av en kollektiv prosess som både brukerne og designerne av artefakten bidrar til. Denne dimensjonen innebærer også at skjemaer utvikles i sosiale overføringsprosesser, som for eksempel gjennom bruksanvisninger (Verillion & Rabardel, 1995, s. 87). Dersom en elev for eksempel bruker programmeringsspråket MakeCode, da vil den elevens instrumentelle skapelse både bli påvirket av dens egne allerede eksisterende kunnskap og erfaringer, og de rammene som er satt av den som utviklet programmeringsspråket. Rammene i programmeringsspråket MakeCode kan for eksempel være at kodeblokkene ikke kan stå i hvilken som helst rekkefølge for at programmet skal fungere. Koden vil bare fungere dersom kodeblokkene er satt sammen på måter som logisk gir mening til programmet.

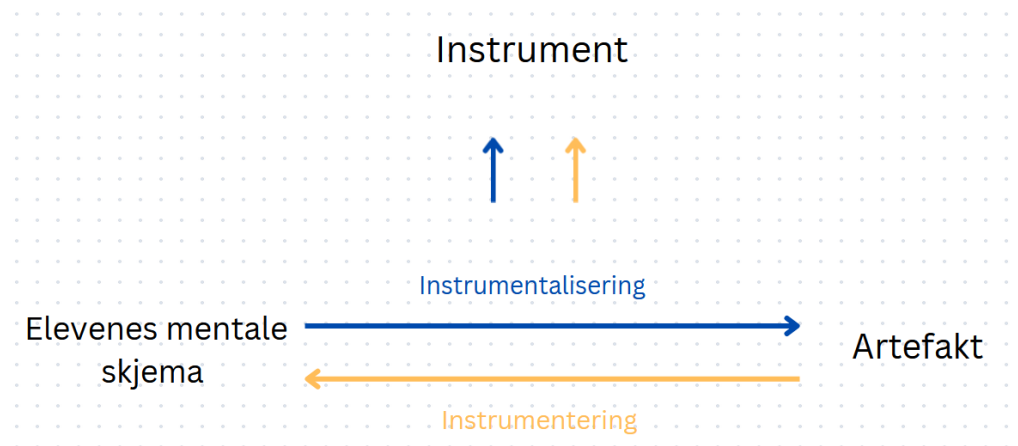
Rabardel (2000, gjengitt i Trouche, 2003, s. 786) beskriver instrumentell innfallsvinkel i matematikk basert på to nøkkelelementer. Det første nøkkelelementet er at en artefakt , gjennom begrensningene det innebærer for brukeren, legger opp til både hvilke fysiske, men også hvilke kognitive prosesser som vil kunne skje. Det andre nøkkelelementet er at en artefakt blir til et instrument gjennom en lang prosess, instrumentell genesis, som har sammenheng med artefaktens og subjektets karakteristikk (Trouche, 2003, s. 786).

En instrumentell skapelses-prosess er en kombinasjon av to delprosesser (Trouche, 2003, s. 789). Disse to prosessene kalles *instrumentalisering* («*instrumentalization*») og *instrumentering* («*instrumentation*») (Rabardel & Bourmaud, 2003, sitert i Trouche, 2003, s. 789). Instrumentaliseringprosessen er prosessen knyttet til artefakten. Denne prosessen kan inkludere flere stadier. Et stadium er oppdagelse og valg av artefaktens funksjoner. Et annet er tilpasning av artefakten. Et siste stadium er transformering av artefakten gjennomført av subjektet. Dette stadiet kan for eksempel innebære endring av artefaktens egenskaper, som gjennom å endre på menylinjen i et program eller opprette tastaturnarveier. I MakeCode kan dette for eksempel være det å tilpasse menyen ut fra hvilke typer blokker en trenger. For eksempel kan elevene velge å legge til utvidelser til micro:bit, som gir tilgang til andre typer blokker, eller elevene kan vise eller skjule avanserte typer blokker. I denne delprosessen endrer subjektet på artefakten for å få den til å oppnå en funksjon, og på denne måten påvirker subjektet artefakten.

Instrumenteringsprosessen er knyttet til subjektets tilegnelse av skjema for å kunne utføre en type oppgave. Subjektets arbeidsmåter blir påvirket av artefakten og de begrensningene som finnes i den. I denne delprosessen benytter subjektet seg av skjemaene sine for bruk av artefakten for å løse oppgaver (Rabardel & Bourmaud, 2003, sitert i Trouche, 2003, s. 789). For eksempel vil en elev i denne delprosessen utvide og endre sine skjema gjennom arbeidet med editoren for programmeringsspråket MakeCode som artefakt. På denne måten vil subjektet også formes av artefakten. Eleven kan for eksempel prøve seg fram med blokker som de ikke har brukt før, og oppdage hvordan disse fungerer. Eleven utvider da sine skjema knyttet til arbeidet med MakeCode, og kan bruke denne kunnskapen i videre arbeid.

Som nevnt vil artefakten bli påvirket av subjektet gjennom instrumentalisering. I tillegg til at subjektets arbeidsmetoder blir påvirket av artefakten gjennom instrumentering. Instrumentering og instrumentalisering virker på denne måten i hver sin retning, som vist i figur 2.

Figur 2: Prosessen med å skape et instrument



2.3 Programmering

Programmering er utformingen av et dataprogram som avgjør hvordan en datamaskin, en robot eller andre elektroniske apparater skal fungere mens programmet er aktivt eller kjører (Vihovde, 2023). Programmering er altså å lage instruksjoner i et dataprogram for å få et eller annet elektronisk apparat til å utføre ønsket oppgave.

I programmering møter man på begreper som program og kode. Et program er et sett med instruksjoner. Et dataprogram er da et sett med instruksjoner som utføres når en datamaskin kjører programmet (Haraldsrud et al., 2020, s. 17). Når programmet kjøres av en datamaskin, leser maskinen kode for kode for å forstå hva den skal gjøre. En kode er i dette tilfellet, og videre i denne masteroppgaven, brukt som et begrep for én enkelt instruksjon. I MakeCode for eksempel en startblokk eller en inndata-blokk. De ulike kodene i MakeCode er beskrevet nærmere i kapittel 2.3.1.

I programmering finnes det ulike språk man kan bruke. Disse språkene blir delt inn i tekstbaserte programmeringsspråk og blokkbaserte programmeringsspråk (Franklin et al., 2016). Eksempler på tekstbaserte programmeringsspråk er Java, Python og C. I slike tekstbaserte programmeringsspråk starter man oftest med en blank skjerm og skriver inn kodene tegn for tegn (Franklin et al., 2016, s. 218). Tekstbaserte

programmeringsspråk inneholder bokstaver, tall og tegnsymboler (Franklin et al., 2016). I tekstbaserte programmeringsspråk må kodene skrives helt riktig for at datamaskinen skal forstå hva den skal gjøre (Haraldsrud et al., 2020, s. 17). Det gjør at dersom en elev glemmer et punktum eller staver et ord feil, vil ikke datamaskinen forstå hvilke handlinger eleven prøvde å instruere den til å utføre. Siden tekstbaserte programmeringsspråk starter med en blank skjerm gir arbeid med slike programmeringsspråk mindre støtte for elevene knyttet til syntaks og instruksjoner (Franklin et al., 2016, s. 218). Blokkprogrammering gir større støtte for elevene knyttet til syntaks, gjennom at de får ferdiglagde kodeblokker hvor de bare må tenke på hvordan disse kodeblokkene kan settes sammen.

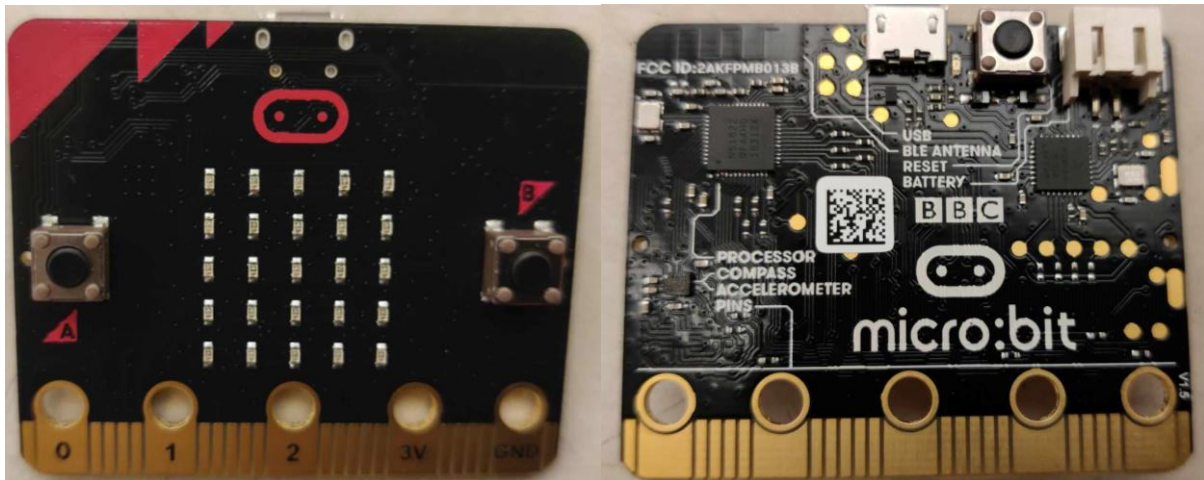
Eksempler på blokkbaserte programmeringsspråk er *MakeCode* og *Scratch*. I arbeid med blokkbaserte programmeringsspråk kan elever utvikle forståelse for grunnleggende programmering uten å måtte skrive inn eller huske kommandoene som trengs for å lage koder og programmer (Franklin et al., 2016, s. 217). Elevene trenger ikke å skrive inn kodene tegn for tegn, men kan heller velge blokker som inneholder ferdigformulerte koder. Disse blokkene er i Scratch og MakeCode fargekodet etter hvilken funksjon blokken har, og de har ulik form ut ifra hvordan de kan settes sammen slik at det gir mening for datamaskinen. Blokkene ligner på puslebrikker, som skal kobles sammen til programmer (se figur 3). Utformingen av blokkene i blokkbaserte programmeringsspråk kan fungere som en støtte for elevene i deres arbeid med programmeringsoppgaver. På grunn av blokkenes utforming og beskrivelse kan blokkbaserte programmeringsspråk oppleves som mer intuitive enn tekstbaserte programmeringsspråk.

Figur 3: Eksempel på blokker som passer sammen i MakeCode



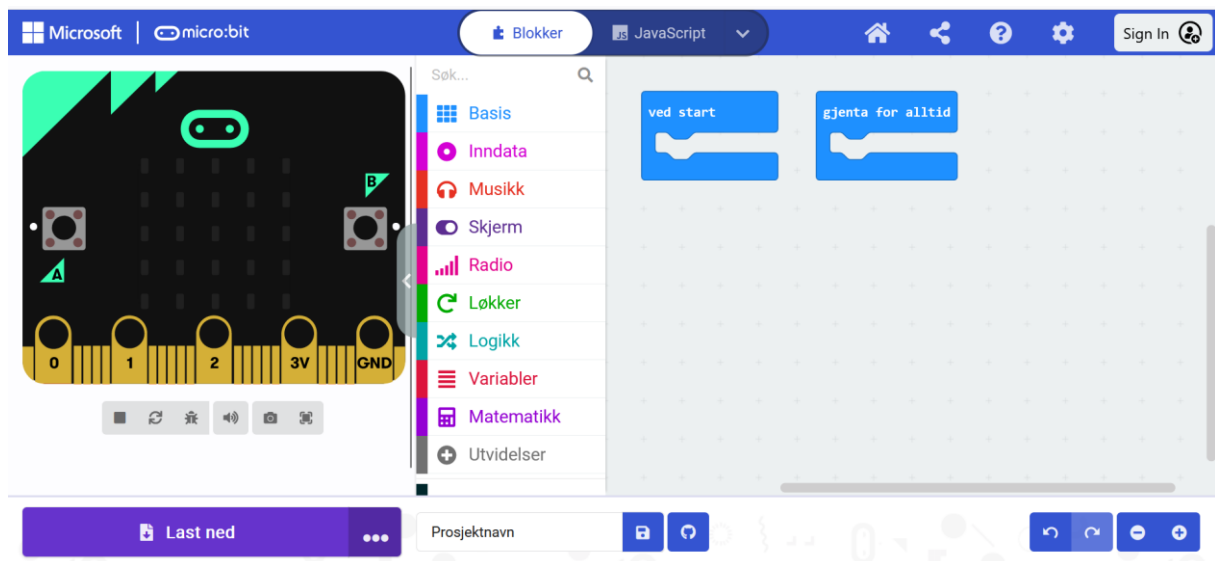
2.3.1 MakeCode og micro:bit

Figur 4: Framsiden (til venstre) og baksiden (til høyre) av en fysisk micro:bit



I figur 4 er det bilde av en fysisk micro:bit. Micro:bit er en fysisk databrikke som kan kobles til en pc og programmeres ved å bruke ulike programmeringsmiljø. I denne masteroppgaven har jeg valgt å bruke programmeringsmiljøet MakeCode. Den fysiske micro:biten har to knapper som er merket A og B, og en skjerm som består av 5x5 piksler. Hver piksel har en lysdiode som kan skrues på eller av. Det er altså 25 lysdioder som kan skrues på eller av. Denne skjermen kan brukes til å skrive bokstaver, tall eller lage andre ikoner. Det at skjermen bare består av 25 dioder setter begrensninger for hvor mye som kan vises på skjermen om gangen, i tillegg til hvor mange detaljer man kan benytte i lagingen av ikoner. Dersom micro:bit blir programmert til å vise et ord, en setning eller et flersifret tall vil bokstavene eller tallene rulle over skjermen. For eksempel dersom man programmerer micro:bit til å vise tallet 345, vil først 3 vises bevegende over skjermen fra høyre til venstre. Når sifferet 3 er rullet nesten over skjermen kommer tallet 4 rullende inn fra høyre. På samme måte kommer 5 rullende på skjermen fra høyre når 4 nesten har forsvunnet på høyre side av skjermen. Altså vises bare ett siffer om gangen, noe som krever at man følger nøye med for å få med seg alle sifrene i flersifrede tall. I tillegg vises hvert siffer på skjermen i bare omtrent ett sekund før det forsvinner igjen. Micro:bit har også innebygde sensorer, bluetooth-funksjon, og høyttaler.

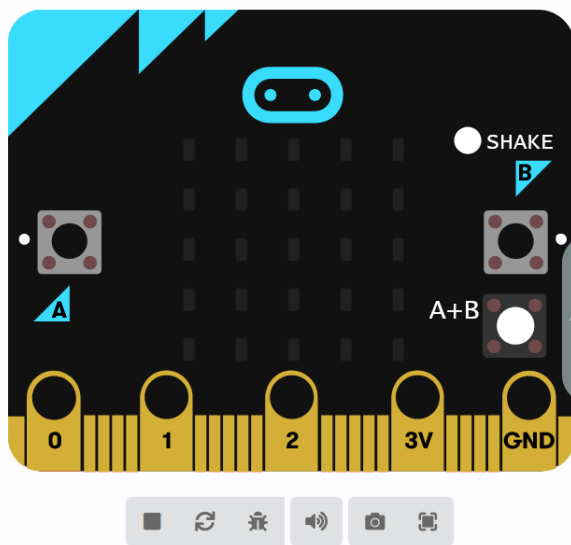
Figur 5: Den offisielle editoren til micro:bit



For å kunne programmere en fysisk micro:bit trenger man en *editor*. En editor er et verktøy for å lage programmer til micro:bit. Micro:bit har en offisiell editor som finnes på makecode.microbit.org (Micro:bit Educational Foundation, u.å.). Denne editoren bruker blokkprogrammeringsspråket MakeCode som hovedprogrammeringsspråk, men det er også mulig å bytte til de tekstbaserte programmeringsspråkene Python og JavaScript. MakeCode inneholder blokker i ulike farger, som er inndelt ut ifra hvilken kategori de hører til i. Blokkene har også ulik form utfra hvilke blokker de kan kobles sammen med, for å danne programmer.

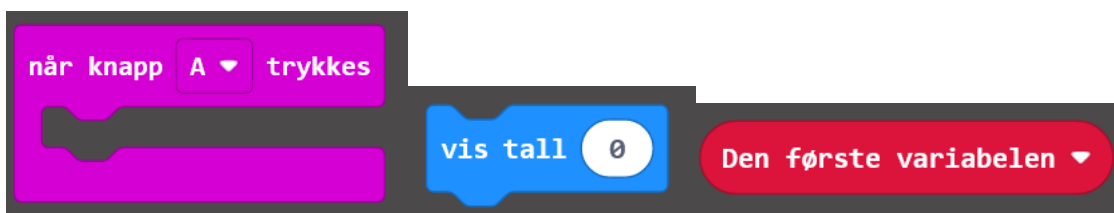
Figur 5 viser den offisielle editoren til micro:bit slik som den vil se ut når man har laget seg et nytt prosjekt på nettsiden makecode.microbit.com. Til venstre på bildet ser man en animasjon av en micro:bit, hvor man kan se hva en micro:bit ville gjort ut fra de instruksjonene som er gitt i det store feltet på høyre side av bildet. Det er altså en simulator for å sjekke om programmet fungerer uten at man må føre programmet over på en faktisk micro:bit. Animasjonen av micro:biten er ikke helt lik den faktiske micro:biten, for at den skal være tilpasset datamaskinformatet. Blant annet dukker det opp knapper man kan trykke på som heter «A+B» og «shake» når man programmerer micro:biten til å utføre handlinger. Disse knappene tilsvarer at man på den fysiske micro:biten trykker på knapp A og B samtidig og at man rister den fysiske micro:biten (se figur 6). Siden animasjonen av en micro:bit simulerer de funksjonene en fysisk micro:bit har, kan man både bruke editoren uten å ha en fysisk micro:bit tilgjengelig eller man kan koble den sammen med en fysisk micro:bit. En fysisk micro:bit kan kobles sammen med en datamaskin med kabel, og på denne måten overfører programmer fra editoren til micro:biten.

Figur 6: Animasjonen av micro:bit med knappene SHAKE og A+B



Videre vil jeg forklare hvordan noen utvalgte blokker i MakeCode fungerer. Blokkene har ulike farger og ulik form, formen avgjør hvilke blokker som kan bygges sammen, og fargen sier noe om hvilken funksjon blokken har. Videre vil jeg forklare de tre formene blokkene i MakeCode kan ha, og tre av fargene som blokkene i MakeCode kan ha. Disse fargene er valgt fordi det er de fargene som er brukt i dette masterprosjektet.

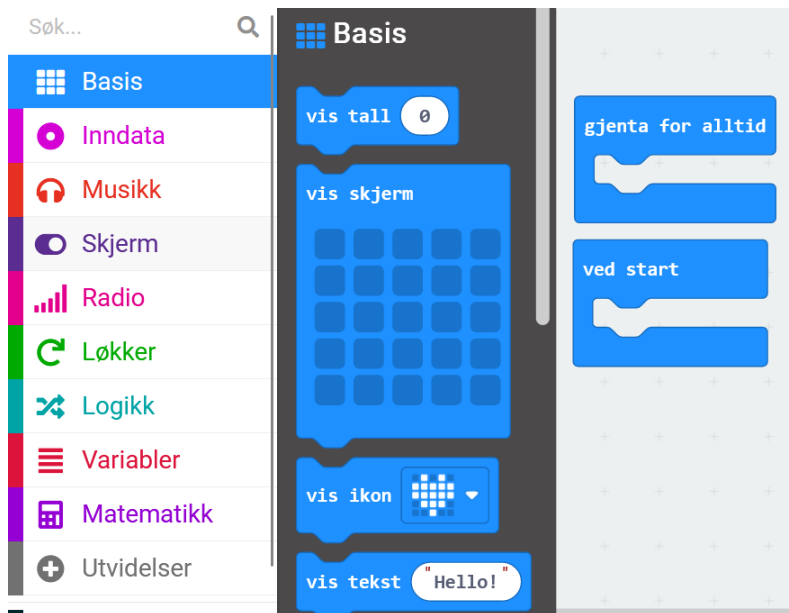
Figur 7: Startblokk, handlingsblokk og verdiblokk



De tre formene som blokkene i MakeCode kan ha vises i figur 7. Jeg har valgt å kalle dem startblokker, handlingsblokker og verdiblokker. Bildet til venstre i figur 7 er en startblokk. Denne typen blokker har mulighet til å ha andre blokker plassert inni seg. Startblokkene kan ikke kobles til andre blokker over og under, noe som er grunnen til at de er flate øverst og nederst. Startblokkene må være til stede for å lage et program. Noen startblokker aktiveres når programmet startes, mens andre, slik som den på figur 7, blir aktivert når en handling utføres. I midten på bildet i figur 7 er det en handlingsblokk. Denne typen blokk kan plasseres inni ulike typer startblokker, og man kan koble flere handlingsblokker sammen. Formen til handlingsblokkene viser også at den kan kobles sammen med andre blokker ved at de har spor over og under, hvor de passer sammen med andre handlingsblokker eller inni startblokker. Handlingsblokkene inneholder en kommando for at noe skal skje, for eksempel at micro:biten skal vise en gitt tekst. Andre handlingsblokker inneholder kommandoer for å vise tall eller ikoner, eller endre eller sette variabler. Variabler blir nærmere beskrevet senere i dette delkapittelet. Enkelte handlingsblokker, slik som den i figur 7, inneholder også et hvitt felt. Disse hvite feltene er felter hvor man kan sette inn en verdi, som kan være enten bokstaver eller tall. Man kan enten trykke på det hvite feltet, og skrive inn en verdi, eller man kan bruke verdiblokker. Verdiblokker har formen som blokken til høyre på bildet i figur 7. Verdiblokkene kan være variabler som er gitt en verdi, et regnstykke, eller være koblet til sensorene i micro:bit.

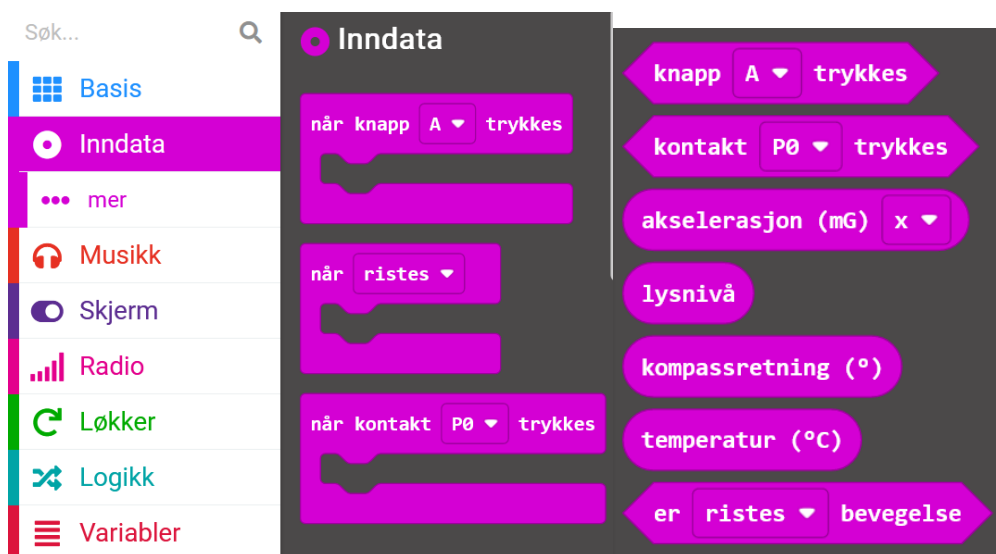
De tre formene som blokkene i MakeCode kan ha; startblokk, handlingsblokk og verdiblokk, kan også passe inn i ulike kategorier. Kategoriene kan inneholde alle typene blokker. Det finnes mange kategorier for blokker, som man kan se på venstre side av figur 8, men jeg velger å forklare tre av disse nærmere. De tre jeg har valgt er de kategoriene som blir brukt i undervisningsopplegget jeg laget til dette masterprosjektet.

Figur 8: Noen basisblokker



Blokkene i MakeCode har farge etter hvilken kategori de tilhører. Som man ser på figur 8 er basisblokkene blå. Basisblokkene inneholder mest handlingsblokker som er kommandoer for å få micro:bit til å vise noe på skjermen. På figur 8 er det for eksempel basisblokker som får micro:bit til å vise tall, ikon og tekst. I tillegg til handlingsblokker finnes det også to startblokker i kategorien basisblokker. Disse er vist på høyre side på figur 8. Disse blokkene gjør at programmet som er laget med dem gjennomføres automatisk, enten når programmet starter, eller kontinuerlig gjennom hele programmet.

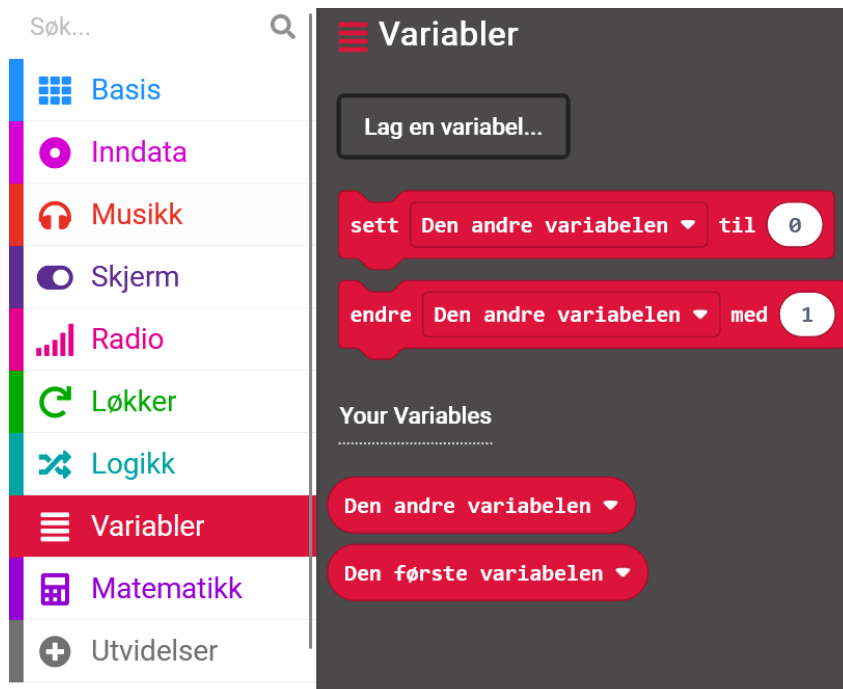
Figur 9: Noen inndatablokker



En annen kategori blokker som MakeCode har er de rosa inndatablokkene. Disse blokkene er knyttet til micro:bitens knapper og sensorer. Kategorien inndatablokkene

innebærer både startblokker og verdiblokker. Startblokkene i inndatakategoriene brukes til å få micro:bit til å kjøre et program ved for eksempel knappetrykk. Inndatakategoriene inneholder også verdiblokker som er tilknyttet sensorene til micro:bit-en, vist til høyre i figur 9.

Figur 10: Variabelblokkene

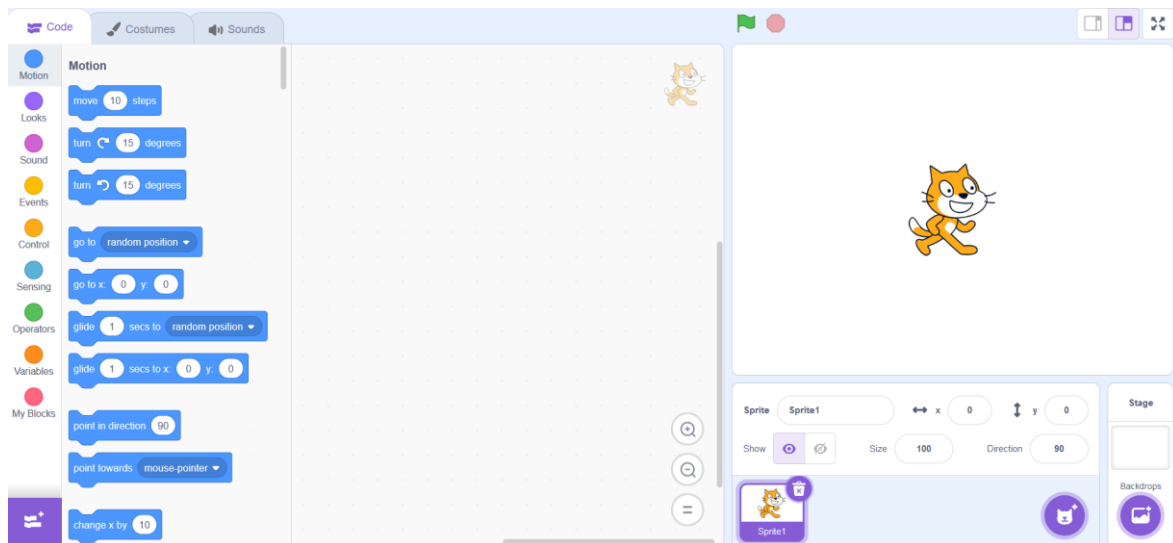


Siste kategorien som jeg skal gå nærmere inn på er variabelblokkene. Innenfor variabelblokkene finnes det handlingsblokker og verdiblokker. For å kunne bruke variabelblokkene må man først lage en variabel ved å trykke på knappen «Lag en variabel» som man kan se på figur 10. Det å lage en variabel innebærer å gi variabelen et navn slik at man har en variabel å utføre handlingene i handlingsblokkene på. Handlingsblokkene i verdiblokkene brukes til å sette eller endre en variabel. Man kan endre eller sette hvilken som helst variabel man har laget ved å trykke på den hvite pila på handlingsblokkene og velge den variabelen man ønsker. Den variabelen som ble sist laget er den som er valgt i handlingsblokkene som standard.

2.3.2 Scratch

Scratch er et annet blokkbasert programmeringsspråk og programmeringsmiljø. I motsetning til micro:bit og MakeCode innebærer ikke Scratch noen fysisk gjenstand som skal programmeres. Scratch ble utgitt i 2007 og ble utviklet av Massachusetts Institute of Technology (MIT) med mål om å være et verktøy for å introdusere programmering til nybegynnere (Haraldsrud et al., 2020, s. 88). Scratch er mye brukt rundt om i verden, også i Norge. I Norge finnes det kodeklubber og organisasjoner, slik som som *Lær Kidsa Koding* som utvikler kurs og undervisningsopplegg basert på bruk av Scratch (Haraldsrud et al., 2020, s. 88).

Figur 11: Scratch



I figur 11 ser man hvordan Scratch ser ut når man åpner et nytt prosjekt på nettsiden scratch.mit.edu. På mange måter er Scratch likt MakeCode, men det finnes noen forskjeller som kan skape forvirring for elever dersom man skal bytte mellom de ulike blokkprogrammeringsspråkene. Elevene som deltok i dette masterprosjektet, hadde erfaring med bruk av Scratch før vi introduserte dem til MakeCode. Dette skapte noen utfordringer i bruken av MakeCode, som jeg beskriver nærmere i delkapittel 3.1. En ting som er ulikt mellom Scratch og MakeCode, er kategoriene som er brukt i de ulike editorene. Begge editorene har en fargekodet inndeling i kategorier, men kategoriene er ulike. Det kan man se ved å sammenlikne kategoriene som er på venstresiden på figur 11 og på figur 8. I tillegg er de mulige blokkene ulike i Scratch fra MakeCode. Siden man i Scratch programmerer bare på skjerm møter man på andre typer blokker som *turn 15 degrees* og *go to x 0 y 0*. Slike blokker eksisterer ikke i MakeCode på grunn av begrensningene som ligger i funksjonene som en fysisk micro:bit har. MakeCode har også noen blokker som ikke Scratch har, for eksempel alle blokker som er knyttet til micro:bitens sensorer.

2.3.3 *Unplugged* programmeringsaktiviteter

Unplugged programmering er aktiviteter som bygger på viktige elementer i programmering, men hvor man ikke benytter seg av datamaskiner. Sammen med praksisgruppen utførte jeg *unplugged* programmeringsaktiviteter med elevene som var med i studien min, som en inngang til arbeidet med programmering. Dette blir forklart nærmere i delkapittel 3.2 Busuttil og Formosa (2020, s. 570) beskriver *unplugged* programmeringsaktiviteter som praktiske aktiviteter som skal hjelpe elever til å forstå nøkkelelementer innenfor informatikk uten å faktisk bruke en datamaskin. Det finnes mange fordeler med *unplugged* aktiviteter, som at man ikke trenger å ha datamaskiner tilgjengelig, noe som gjør slike aktiviteter enklere å gjennomføre for lærere i skolen. I tillegg gir slike aktiviteter støtte i utviklingen av problemløsning og *computational thinking* (Busuttil & Formosa, 2020, s.570). Aktivitetene legger også opp til gruppearbeid, og at elevene selv skal være så engasjerte i oppgaven som mulig (Cortina, 2015 i Busuttil & Formosa, 2020, s.570). Et eksempel på en *unplugged* programmeringsaktivitet er en aktivitet hvor elevene skal jobbe i par og den ene i paret skal føre den andre gjennom en løype. I denne aktiviteten har den som skal føres gjennom løypa bind for øynene, og hen skal bare gjøre det hen får beskjed om fra den

andre eleven. Gjennom en slik aktivitet er målet at elevene skal oppdage hvor presis man må være når man «programmerer» noen til å gjøre noe. I tillegg jobber elevene med problemløsning som likner den man gjør i programmering, ved at de må tenke ut hvilke kommandoer som må gis for å oppnå det resultatet de ønsker.

2.3.4 Tidligere forskning på programmering i matematikkundervisning

I 2018 utførte Sanne Erika Forsström og Odd Tore Kaufman en litteraturstudie innenfor programmering i matematikkundervisning, med fokus på undervisningspotensialet som programmering har i matematikkundervisningen. I deres litteraturgjennomgang så de på 15 artikler som omhandlet elever i alderen 6-16år. Funnene fra litteraturgjennomgangen ble delt inn i fire kategorier (Forsström & Kaufman, 2018, s. 25). De fire kategoriene er elevenes motivasjon til å lære matematikk, elevenes prestasjoner i matematikk, samarbeidet mellom elevene og den endrede rollen til læreren.

I de 15 artiklene som ble sett på i litteraturstudien fantes det ingen generaliserbar sammenheng mellom arbeid med programmering og elevenes motivasjon (Forsström & Kaufman, 2018, s. 25). En del av artiklene fant en sammenheng mellom motivasjon og programmering, men ifølge Forsström og Kaufman (2018, s. 26) kunne ikke disse funnene generaliseres blant annet fordi ingen av studiene ble utført i en vanlig klasseromssituasjon. Flere av artiklene fant at programmeringen virket positivt inn på elevenes prestasjoner i matematikk (Forsström & Kaufman, 2018, s. 26). I flere av artiklene fant de at prestasjonene ble litt bedre for en del elever, men ikke alle (Forsström & Kaufman, 2018, 26). Selv om enkelte artikler viste positiv korrelasjon mellom programmering og prestasjon mener Forsström og Kaufman (2018, s.26) at heller ikke disse resultatene er generaliserbare blant annet på grunn av at effektene bare gjaldt enkelte grupper av elever. Samarbeidet mellom elevene henger sammen med lærerens endrede rolle i arbeid med programmering (Forsström & Kaufman, 2018, s. 27). Samarbeid blir mye brukt i programmeringsaktiviteter, og læreren må derfor både veilede elevene i programmering, og sørge for at elevene samarbeider på en god måte (Forsström & Kaufman, 2018, s. 27). Den endrede lærerrollen innebærer at læreren i større grad veileder elevene, heller enn at læreren skal stå fremfor elevene å undervise.

Forsström og Kaufman (2018, s. 30) konkluderer med at det trengs mer forskning som knytter programmering til andre sider av matematikkfaget. Det finnes en del forskning som kobler sammen geometri og programmering, men det finnes lite som knytter andre matematiske temaer til programmering (Forsström & Kaufman, 2018, s.30). Programmering er ikke bare et undervisningsverktøy, men også potensielt en metode for å lære og forstå kunnskap (Forsström & Kaufman, 2018, s.28). Allikevel fant Forsström og Kaufman (2018, s. 30) gjennom denne litteraturstudien at det finnes lite eksisterende forskning på undervisningspotensialet i programmering i matematikkundervisningen.

2.4 Tidlig algebra

Tidlig algebra henger sammen med yngre elevers evne til å delta i algebraisk tenkning (Kieran, 2004, s. 149). Denne algebraiske tenkningen utvikles gjennom arbeid med oppgaver innenfor symbolsk algebra, men også gjennom arbeid med oppgaver som for eksempel analyserer forholdet mellom størrelser, oppdager strukturer, undersøker endringer, generaliser, problemløser, modellerer, argumenterer, beviser og forutser (Kieran, 2004, s. 149). Algebraisk tenkning innebærer en annen type tenkning enn den som er nødvendig i ren aritmetisk tenkning, altså tenkning som er knyttet til å utføre ulike operasjoner i matematikk (Kieran, 2004, s. 140).

2.4.1 Multiplikasjon

I starten når elevene utvikler strategier for løsning av oppgaver i multiplikasjon og divisjon kan de allerede en del om addisjon og subtraksjon. De vil derfor relatere de nye strategiene i multiplikasjon til det de allerede kan om løsningen av oppgaver i addisjon og subtraksjon. I introduksjonen av multiplikasjon må elevene lære et nytt ordforråd, da ord som multiplisere og multiplikasjon ikke er brukt i dagligtalen. Multiplikasjon er ofte det som blir introdusert til elevene først i skolen, men divisjon er ofte mer kjent for elevene gjennom fordeling av objekter (Anghileri, 2006, s. 86).

Greer (1992) deler multiplikasjon og divisjon inn i teoretiske modeller, som påvirker hvordan elevene vil tenke i arbeidet med å løse oppgaver. De viktigste av disse modellene er; like grupper, multiplikativ sammenlikning, kartesisk produkt og rektangulære mønster eller produkt (Greer, 1992, s. 278). Anghileri (2006, s. 86) viser til Greers (1992) modeller for multiplikasjon og beskriver at alle situasjonene involverer tre tall i et matematisk forhold til hverandre. De ulike tallene er antall objekter i en mengde, hvor mange slike mengder man har og til slutt det totale antallet objekter til samme i alle mengdene. I multiplikasjon er det det totale tallet som man spør etter, mens man i divisjon kan spørre etter enten antall objekter i en mengde eller antall like store mengder. Like grupper-modellen er ofte elevenes første møte med bruk av multiplikasjon (Greer, 1992, s.276). Et eksempel på en oppgave som er formulert slik at elevene kan bruke like grupper-modellen er «4 elever har 3 fargeblyanter hver. Hvor mange fargeblyanter har de til sammen?». I slike oppgaver har de to tallene helt ulike roller. Mengden elever er multiplikator som opererer på mengden fargeblyanter, for å produsere svaret, antall fargeblyanter elevene har til sammen (Greer, 1992, s.276). Alternativt kan like grupper-modellen bli konseptualisert med utgangspunkt i rate, for eksempel: «Dersom det er 3 fargeblyanter per elev, hvor mange fargeblyanter har 4 elever?». Generelt vil tallet på hvor mange det er i hver gruppe, antall fargeblyanter, bli multiplisert med antall grupper, antall elever, for å finne den totale mengden (Greer, 1992, s.277). Dette er motsatt av den første måten å framstille like grupper-modellen.

En annen modell for multiplikasjon er multiplikativ sammenlikning. Multiplikativ sammenlikning kan uttrykkes som «n ganger så mange som». Et eksempel på multiplikativ sammenlikning kan være; «Lisa har 4 ganger så mange bananer som Pål. Pål har 5 bananer. Hvor mange bananer har Lisa?». I slike oppgaver kan den multiplikative faktoren, som er 5 i eksempelet, ses på som multiplikanden, men man kan også se på en annen måte. Dersom man ser motsatt på det tenker man på at det er 4 av Lisas bananer for hver av Påls bananer, slik at multiplikanden blir 4.

Modellen kartesisk produkt danner en helt annerledes kontekst for multiplikasjon. For eksempel; «Det er 3 jenter og 4 gutter på dans, hvor mange ulike partnerskap er mulig?». Tallene i slike oppgaver er likestilt, og det er ikke nøye hvilket tall man begynner med for å løse oppgaven innenfor konteksten (Greer, 1992, s.277).

Den siste situasjonsmodellen er rektangel-areal. Oppgaver som benytter denne modellen, tar utgangspunkt i at arealet av et rektangel (når man bare ser på hele positive tall) kan representeres ved å dele rektangelet inn i rutenett. For eksempel kan et rektangel ha lengden 5cm og bredden 3 cm. Da kan man lage et rutenett med ruter på 1 cm hver vei, og deretter telle rutene for å finne svaret. Innenfor denne modellen, slik som ved kartesisk produkt, er ikke rekkefølgen på tallene viktig for løsningen av oppgaven, altså man kan ikke skille mellom multiplikand og multiplikator. Slike oppgaver

kan gi elevene gode muligheter til å oppdage spesifikke egenskaper ved multiplikasjon, som for eksempel kommutativitet (Greer, 1992, s.277).

Det er ifølge Rønning og Burheim (2020, s. 131) viktig at elevene har ulike tankemodeller for multiplikasjon tilgjengelig for å kunne gjenkjenne situasjoner som multiplikativ. Enkelte tankemodeller har begrensninger hvor de ikke vil gi mening i alle situasjoner. For eksempel vil gjentatt addisjon bare gi mening som modell for multiplikasjon i situasjoner der antall grupper er et positivt heltall (Rønning & Burheim, 2020, s.131). Disse begrensningene gjør at elevene trenger å kjenne til varierte tankemodeller for multiplikasjon, slik at de kan ta i bruk en modell som passer, også i situasjoner som ikke bare innebærer heltall (Rønning & Burheim, 2020, s.131).

De enkleste typene multiplikative situasjoner som elevene møter vil sannsynligvis være situasjoner hvor det er en-til-mange sammenheng mellom to mengder (Nunes & Bryant, 1996 hentet i Anghileri 2006, s. 86). Et eksempel på en slik sammenheng er at en bil har fire hjul. Denne typen situasjoner er knyttet til forholdstall eller skaleringsfaktor, og er grunnlaget for multiplikativ tenkning i motsetning til additiv tenkemåte. Dette står i kontrast til gjentatt addisjon, selv om gjentatt addisjon ikke er det samme som å legge sammen, slik man gjør i addisjon (Nunes & Bryant, 1996 hentet i Anghileri 2006, s. 86). I multiplikasjon er det to tall som har ulik funksjon, antallet i en mengde og antallet like store mengder, som må koordineres på en mer komplisert måte enn det som trengs i addisjon. Dersom man for eksempel skal telle 6 mengder med 4 hjul, for eksempel dersom man har 6 biler og skal telle mengden hjul. Da må man telle antallet biler, mengder på 4 hver, som objekter som det er 6 av, samtidig som man må telle hvert hjul i hele mengden for å finne antall hjul til sammen. Man må altså telle hvert av hjulene som individuelle objekter, og samtidig kunne telle antallet mengder på 4 objekter. Dette kan relateres til hvor store hopp og hvor mange like hopp på tallinja (Anghileri, 2006, s. 87). Tidlige erfaringer knyttet til dette involverer at barn lærer å telle mengder med objekter i tillegg til å telle de individuelle objektene. Oppdeling og gruppering i like mengder vil kunne hjelpe elevene til å utvikle språket og de numeriske sammenhengene som senere danner grunnlaget for å kunne utføre utregninger. Selv de elevene som har god tallforståelse skjønner ikke nødvendigvis den abstrakte og kompliserte måten tre tall relaterer til hverandre i multiplikasjon og divisjon, slik som de forstår sammenhengene i addisjon.

2.4.2 Mønster i matematikk

Mønster i matematikk handler ifølge Clements & Sarama (2009, i Rivera, 2013, s.3) om å lete etter matematiske regelmessigheter og strukturer. Identifisering og utforsking av mønstre i matematikkfaget gir muligheter for å holde orden, finne sammenhenger og oppdage forutsigbarhet i tilsynelatende uorganiserte situasjoner som man møter på i matematikkfaget. Arbeid med mønster i matematikkfaget gir også muligheten til å gjøre generaliseringer som strekker seg utenfor den informasjonen man har framfor seg. Clements & Sarama (2009, i Rivera, 2013, s. 3) understreker også at mønster i matematikk er mer enn bare en adskilt del av matematikkfaget, men at arbeid med mønster er en prosess, et studieområde og en tenkevane.

Resonnering som omhandler deres læring om verden er spesielt viktig for elevers matematiske utvikling (Mulligan & Mitchelmore, 2013, s. 30). Effektiv matematisk resonnering handler i essensen om elevenes evne til å oppdage mønstre og strukturer i både virkelige situasjoner og symbolske objekter. Gjennom arbeid med mønstre og strukturer i matematikk dukker det opp muligheter for generaliseringer hvor abstraksjon

av ideer og relasjoner i ulike deler av matematikkfaget kan finne sted (National Council of Teachers of Mathematics, 2010, i Mulligan & Mitchelmore, 2013, s. 30).

Nesten all matematikk er basert på mønster og struktur (Mulligan & Mitchelmore, 2013, s. 30). Begrepene mønster og struktur kan allikevel være vanskelig å definere. Mulligan & Mitchelmore (2013) beskriver mønster og struktur som to atskilte sider i arbeidet med mønster i matematikk. De definerer mønster som hvilken som helst forutsigbar regelmessighet som involverer tall, avstand eller måling, og struktur defineres som hvordan ulike elementer er organisert og relatert til hverandre (Mulligan & Mitchelmore, 2013, s. 30).

Forskere skiller ofte mellom ulike typer mønster. De vanligste typene mønster er repeterende mønster, voksende mønster og romlig strukturmønster. Et repeterende mønster innebærer en konstant repetisjonsenhet, som gjentas i det uendelige. For eksempel er mønstrene IOIOIO og UIOUIOUIO repeterende hvor det første har repetisjonsenheten IO og det andre har repetisjonsenheten UIO. Voksende mønster er mønster som øker eller synker systematisk. Eksempler på et voksende mønster hvor mønsteret øker hver gang kan være OIOOIIOOOIII eller 7-14-21 hvor mønstrene øker med like mye hver gang. Et eksempel på et voksende mønster hvor mønsteret avtar er 2-1,5-1-0,5, hvor mønsteret i tallene minker med 0,5 hver gang. Romlige strukturmønstre er uforanderlige og beskriver organiseringen av individuelle elementer i et to- eller tredimensjonalt rom (Wijns et al., 2019, 143). Mønsteraktivitetene i denne masterstudien baserer seg på voksende mønster hvor mønstrene øker systematisk.

2.4.3 Mønster i multiplikasjon

Forskning innenfor multiplikasjon og divisjon har funnet at gjenkjennelsen av strukturer er noe som står sentralt i elevenes arbeid med multiplikative resonnement (Mulligan & Mitchelmore, 2013, s. 32).

Det å multiplisere med for eksempel 4 er knyttet til å telle i firere (4, 8 12, 16, ...). Det å knytte denne hoppetellingen til «3 grupper med 4» eller «2 grupper med 4» gjennom bruk av det å telle firere og vise på tallinja, kan hjelpe å elevene til å utvikle en mental strategi som videre er nyttig i kalkuleringer, og som gjør dem mindre avhengig av å manipulere fysiske gjenstander (Anghileri, 2006, s. 87).

Fra å telle i toere, femere eller tiere, blir elevene oppmuntret til å lære tallmønstrene for andre tall, som 20, 100 eller 15, til så høye tall som de klarer. Utviklingen av slike mønstre går fra å telle hvert enkelt tall, til å internalisere tallene i mønsteret, for til slutt å bare bruke multiplene (Anghileri, 1995, hentet i Anghileri, 2006, s. 88).

Det å undervise tallforståelse innebærer å hjelpe barn til å forstå at multiplikasjon og divisjon begge kan relateres til samme tallmønster og at det er disse sammenhengene mellom tallene som er nøkkelen til vellykket oppgaveløsning (Anghileri, 2006, s. 89).

Det å se på multiplikasjon som bare gjentatt addisjon kan føre til begrensninger for elevene i vanskeligere multiplikasjonsarbeid, for eksempel når man skal multiplisere 0,3 og 0,1. For å kunne inkludere slike tilfeller kan man bruke ideen om forhold mellom tall. Anghileri (2006, s. 90) bruker tallinje til å vise forholdet mellom gruppene i multiplikasjon med 0,2 som eksempel. Dette kan hjelpe elevene oppdage hvorfor $0,2 \cdot 0,3$ ikke blir et høyere tall slik som multiplikasjon med hele positive tall alltid blir.

Tradisjonelt har læring av gangetabell blitt viet mye tid på barneskolen (Anghileri, 2006, s. 94). For noen elever er pugging av gangetabeller en effektiv måte å lære seg de faktaene som de trenger i utgreining av multiplikasjonsstykker. For andre elever er det å huske gangetabellen imidlertid en uhandterlig oppgave, og man må derfor også bruke andre strategier for å også gi disse elevene evnen til rask utregning.

2.4.4 Mønster i tallfølger

I denne masterstudien benyttes tallfølger med multiplikativ struktur i mønsteraktiviteter elevene utførte. Tallfølger inkluderer ikke figurer slik som figurfølger gjør. I tallfølger som ikke har figur vil ikke fortsettelsen på tallfølgen kunne helt sikkert forutses (Stylianides, 2008, s. 11). For eksempel om man har tallfølgen 1, 2, 3, 4, vil man ikke kunne være sikker på om tallfølgen videre fortsetter 5, 6, 7, 8, eller 1, 2, 3, 4, eller fortsetter på en helt annen måte. Tallfølger hvor figur mangler, kaller Stylianides (2008, s. 11) for plausibelt mønster. I slike mønster er det ikke mulig å matematisk være sikker på om en fortsettelse på tallfølgen som passer med tallfølgen som er oppgitt, skal velges over andre mulige fortsettelser på tallfølgen. For unngå dette problemet foreslår Stylianides, (2008, s.11) at man for eksempel spør etter det enkleste, eller mest naturlige, mønsteret til en gitt tallfølge.

2.4.5 Multiplikative tallfakta

Grunnleggende tallfakta innenfor multiplikasjon og addisjon et tallkombinasjoner hvor begge leddene er tall som er mindre enn ti (Van de Walle et al., 2020, s. 216). Innenfor divisjon og subtraksjon er grunnleggende tallfakta de korresponderende kombinasjonene av tall til grunnleggende tallfakta i multiplikasjon og addisjon. For eksempel er $3 \cdot 7 = 21$ sett på som grunnleggende tallfakta siden begge faktorene i multiplikasjonsstykket er tall som er mindre enn ti. De korresponderende divisjonsstykkene, $21 : 7 = 3$ og $21 : 3 = 7$, blir sett på som grunnleggende tallfakta siden de korresponderer til et multiplikasjonstykke hvor begge faktorene er tall under ti.

Elevene beveger seg gjennom en prosess i automatiseringen av kunnskap om grunnleggende tallfakta, fra telling til å bare vite svarene på regnestykker uten å regne (Van de Walle et al., 2020, s. 117). Baroody (2006, s. 22) beskriver tre faser i utviklingen av tallfaktakunnskap (Baroody, 2006, s.22).

Fase 1: Tellestrategier. I denne fasen teller elevene seg frem til svaret. For eksempel vil en elev i denne fasen i arbeid med regnestykket $7 \cdot 4$ telle tre sjuere opp fra sju, enten ved bruk av fingre, klosser eller verbalt.

Fase 2: Resonneringsstrategier. I denne fasen bruker elevene kjent kunnskap for å komme frem til svaret. I møte med oppgaven $7 \cdot 4$ vil en elev i denne fasen for eksempel vite at $7 \cdot 2 = 14$, og gjøre det to ganger for å få $7 \cdot 4$.

Fase 3: Behersker tallfakta. I denne fasen produserer eleven svar på en effektiv måte, altså raskt og korrekt. Eleven «vet» svaret uten å måtte tenke seg om. For eksempel vil en elev i denne fasen i arbeid med $7 \cdot 4$ vite at svaret er 28 uten at den trenger å regne seg frem til svaret.

3 Metode

Målet med dette masterprosjektet er som nevnt tidligere å undersøke hvordan programmering som instrument kan bidra til utvikling av kunnskap knyttet til tallfølger med multiplikativ struktur. For å finne svar på dette utførte jeg en designstudie (Akker et al., 2006) på fire elever på 4. trinn hvor de jobbet med oppgaver innenfor programmering og tallfølger med multiplikativ struktur.

I dette kapitlet vil jeg redegjøre for de valgene jeg har tatt tilknyttet denne studien. Jeg vil forklare hvordan denne studien er planlagt og gjennomført. Til slutt vil jeg også redegjøre for etiske betraktninger som har dukket opp i planlegging og gjennomføring av denne studien.

3.1 Forskningsmetode

For å kunne si noe om elevenes utvikling av forståelse trenger jeg et detaljert bilde av elevenes arbeid med oppgaver med programmering. Derfor har jeg valgt å benytte meg av kvalitativ forskningsmetode, som har som formål å generere dyp innsikt i spesifikke temaer gjennom engasjement med steder og sosiale aktører (Clark et al., 2021, s. 350). Kvalitative metoder er kjennetegnet gjennom at de innhenter informasjon gjennom ord eller språk, med en intensjon om å beskrive hva mennesker gjør i sitt hverdagsliv, og hva disse handlingene har å si for dem (Postholm & Jacobsen, 2018, s. 89). Forskere som benytter seg av kvalitativ metode definerer seg ofte innenfor et konstruktivistisk paradigme (Postholm & Jacobsen, 2018a, s. 90). Innenfor det konstruktivistiske paradigmet finner man blant annet sosial konstruktivisme, som tett knyttet til sosiokulturell teori og Vygotskys tenkning om utvikling og læring (Postholm & Jacobsen, 2018a, s. 51), som er en del av det teoretiske grunnlaget for dette masterprosjektet. Det finnes flere typer konstruktivisme, men felles for dem alle er at verden ikke er objektiv, men at verden heller er noe som vi mennesker mer eller mindre aktivt konstruerer (Postholm & Jacobsen, 2018, s. 51). Siden verden er objektiv, og konstrueres av mennesker, mener mange forskere innenfor konstruktivistisk tilnærming at kvantitative data ikke vil kunne måle den komplekse virkeligheten vi mennesker lever i på en god måte, og kvantitative data vil i alle fall ikke kunne måle menneskers forståelse for verden (Postholm & Jacobsen, 2018a, s. 99). Det er altså behov for at forskningen går dypere inn i det som skal forskes på, for eksempel gjennom en designstudie (Akker et al., 2006) slik som jeg har utført.

3.1.1 Designstudie

Dette forskningsprosjektet er som nevnt utført som en designstudie gjennom to økter hvor fire elever ble tatt ut av klasserommet for å jobbe med et oppgavesett hver økt. Designstudier er betegnelsen jeg har valgt å bruke for det Akker et al. (2006), Walker (2006) og Prediger et al. (2015) betegner som «Design Research». Designstudie har blitt utviklet fra ulike steder under ulike navn, som designforskning, designbasert forskning, designeksperiment og utviklingsforskning (Prediger, 2015, s. 878, egen oversettelse). Andre navn på denne typen forskning er formativ forskning og ingeniørforskning (Akker et al., 2006, s. 4). Designforskning kan beskrives som «En rekke tilnærminger, med hensikt å utvikle nye teorier, artefakter og praksiser, som redegjør for og potensielt påvirker læring og undervisning i naturalistiske omgivelser» (Barab og Squire, 2004 referert i Akker 2006, s. 5, egen oversettelse).

En designstudie kan ifølge van den Akker et al. (2006) karakteriseres gjennom fem punkter. Disse punktene er at designforskning er knyttet til intervensjon, den er iterativ, prosessorientert, bruksorientert og teoriorientert (Akker et al., 2006, s. 4).

Karakteristikkene som er forklart av van den Akker et al. (2006) sammenfaller i stor grad med karakteristikken Prediger et al. (2015) bruker til å beskrive designforskning.

Karakteristikkene Prediger et al. (2015) bruker er intervensjon, teoriorientering, prospektering og refleksjon, iterativitet og praksisorientering. Jeg har valgt å fokusere på punktene gitt av van den Akker et al. (2006), og vil beskrive disse nærmere og knytte disse punktene til mitt masterprosjekt.

Den første karakteristikken van den Akker et al. (2006) beskriver designforskning med er som sagt intervensjon. Punktet om intervensjon handler om at formålet med forskningen er å designe en intervensjon i den virkelige verden (Akker et al., 2006, s. 4). Mitt masterprosjekt har som formål å utvikle og gjennomføre undervisningsopplegg, som skal benyttes i den virkelige verden, i klasserommet, for å forbedre sammenhengen mellom programmering i matematikkfaget og andre matematiske tema, spesifikt temaet multiplikasjon.

Den andre karakteristikken er at designforskning er knyttet til iterasjon. Denne karakteristikken handler om at forskningsprosessen innebærer en syklisk tilnærming til design, evaluering og revidering (Akker et al., 2006, s. 4). Denne sykliske tilnærmingen er knyttet til å designe, redesigne og teste instruksjonsaktiviteter og andre aspekter ved designet (Gravenmeijer & Cobb, 2006, s. 54). Mitt masterprosjekt er syklisk ved at jeg først testet ut et undervisningsopplegg jeg hadde laget, for så å vurdere om jeg hadde fått det jeg ville ut av denne økta. Videre ble opplegget endret og utviklet for så å bli utført på nytt med de samme elevene. Et masterprosjekt er et veldig lite prosjekt, som utføres på relativt kort tid. Dette fører til at det ikke blir like mange sykluser med utvikling av opplegget, som man kanskje ønsker i designforskning.

I tillegg er en karakteristikkk ved designforskning at den er prosessorientert. Det betyr at fokuset er på prosessen knyttet intervensjonen, og å forstå og forbedre intervensjonen, ikke å fokusere bare på hva som var utgangspunktet og hva som kommer ut av intervensjonen (Akker et al., 2006, s. 4). Denne karakteristikken kan knyttes til mitt masterprosjekt ved at det er et fokus på å utvikle et undervisningsopplegg som kan knytte programmering og multiplikasjon sammen. Prosessen er altså mye i fokus, gjennom hvordan elevene benytter seg av verktøyet i arbeidet med opplegget.

Karakteristikken at designforskning skal være bruksorientert handler om at designet blant annet måles ut fra dets praktiske funksjon for brukere i virkelige sammenhenger (Akker et al., 2006, s. 4). I mitt masterprosjekt blir designet av intervensjon i stor grad målt ut ifra hvorvidt undervisningsopplegget støtter elevenes utvikling av multiplikatív tenkning.

Den siste karakteristikken er at designforskning er teoriorientert. Det vil si at designet er i det minste delvis basert på teoretisk grunnlag, i tillegg til at testingen av designet bidrar til utvikling av teori. I masterprosjektet mitt er ikke designet basert på et tydelig teoretisk grunnlag. Det er på grunn av lite utviklet teori innenfor det å knytte programmering og andre matematiske tema sammen. Forskningen min er senere knyttet til teori, hvor fokuset er på å utvikle ny teori rundt forskningsområdet. Derfor mener jeg at prosjektet mitt allikevel passer inn i designforskning.

3.2 Kontekst

Datamaterialet til dette masterprosjektet er samlet inn i løpet av januar og februar 2023. Planen var å samle inn datamaterialet i siste praksisuke på studiet, som var i uke 2, men dette ble av praktiske årsaker ikke mulig. Derfor kom jeg tilbake til praksisskolen for å samle inn datamaterialet i løpet av måneden etter praksisen. Praksisskolen hvor datainnsamlingen foregikk på ligger i en storby. Skolen hadde i underkant av 100 elever per trinn. Det var fire elever som deltok i datainnsamlingen, og disse gikk på 4. trinn. Jeg, sammen med medstudenter, hadde vært i praksis på 4. trinn på denne skolen både på høsten og på våren i løpet av skoleåret 2022/2023. I tillegg hadde vi vært i praksis på samme trinn i et tidligere studieår.

Som nevnt tidligere var vi i praksisgruppa i praksis hos samme 4.trinn på høsten i skoleåret 2022/2023 som vi senere kom tilbake til for gjennomføring av datainnsamling. På dette tidspunktet hadde vi allerede kommet fram til at overordnet tema for masterprosjektene være skulle være programmering. Vi hadde også funnet ut at vi skulle bruke micro:bit og MakeCode som utgangspunkt for programmeringen. Derfor benyttet vi muligheten til å introdusere elevene til programmering og bruk av micro:bit. På dette punktet hadde ikke elevene brukt MakeCode og micro:bit, og de hadde nesten ikke jobbet noe med programmering tidligere. Derfor utførte vi i starten ulike *unplugged* programmeringsaktiviteter som en inngang til programmeringen. Unplugged programmeringsaktiviteter er praktiske aktiviteter som skal hjelpe elevene til å forstå nøkkelelementer innenfor informatikk uten å bruke datamaskin (Busuttill & Formosa, 2020, s.570). En av *unplugged* aktivitetene vi gjennomførte med elevene var at de ble delt inn i par, så fikk den ene utdelt et bilde som de skulle forklare til den andre uten at den andre fikk se på bildet. Gjennom en slik aktivitet blir det fort tydelig for elevene hvor viktig det er å gi presise kommandoer, noe som også er viktig i programmering. Før vi kom tilbake for siste praksisperiode jobbet elevene en del med programmeringsspråket Scratch. Noe som skapte noen forvirringer når elevene nå skulle bruke MakeCode. Elevene møtte på forvirring i letingen etter blokker, og i hvordan blokkene må sitte sammen. Utfordringene dukket opp på grunn av at kategoriene ikke er lik i de ulike programmeringsmiljøene, og fordi alle blokkene må sitte sammen i Scratch for at programmet skal fungere, noe som ikke er tilfellet i MakeCode. Disse utfordringene ble mindre for elevene etter hvert i arbeidet med oppgavene, når de ble vant til utformingen og funksjonene i MakeCode.

De fire elevene som deltok i studien, ble tatt ut av klassen i to økter for å jobbe med oppgaver som jeg hadde utformet. Disse to øktene hadde litt over en uke mellom seg. Tidspunktene for datainnsamlingen ble bestemt i samarbeid med praksislærer, for at elevene skulle gå glipp av minst mulig annen undervisning. Lengden på øktene ble bestemt av hvor lang tid elevene brukte på oppgavene, men det kunne ikke ta lengere tid enn 60 minutter for at ikke elevene skulle gå glipp av for mye annen undervisning. Den første datainnsamlingen varte 36 minutter og den andre varte i 50 minutter med en lang pause i midten. Den første økten ble gjennomført i elevenes siste time på en mandag. Den andre økten ble gjennomført på både første og siste del av elevenes skoledag på grunn av at elevene skulle på en oppvisning som jeg ikke visste om på forhånd.

3.2.1 Deltakelse i LAB-Ted

Denne masteroppgaven er en del av forskningsprosjektet *Learning, Assessment and Boundry crossing in Teacher education* (LAB-Ted). Dette forskningsprosjektet er et samarbeid mellom Universitetet i Tromsø, UiT, og Norges Tekniske og Naturvitenskapelige Universitet, NTNU. Forskningsprosjektet er støttet av Norges forskningsråd. LAB-Ted har som et av formålene å utvikle samarbeid mellom universitetet, skolen og lærerstudenter, for å utvide kapasiteten for praksisorientert og profesjonsrettet forskning i lærerutdanningen (Norges teknisk-naturvitenskapelige universitet, u.å.). På grunnlag av dette målet ble temaet for denne masteroppgaven valgt i tett samarbeid med veiledere fra lærerutdanningen, praksislærer og rektor på praksisskolen og andre studenter i praksisgruppa. På denne måten er målet at masteroppgavene blir skrevet med utgangspunkt i tematikk som er både relevant og interessant for det som faktisk skjer i skolen, i tillegg til at interessene til veilederne på lærerutdanningene og studentene selv blir opprettholdt.

3.2.2 Utvalg

Som nevnt var det fire elever som deltok i studien til dette masterprosjektet. Disse elevene ble valgt ut i samarbeid med praksislærer. Det første kriteriet for utvelgelse til å delta i studien var at praksislærer hadde mottatt samtykkeskjema fra foreldrene. Videre var et viktig kriterium at elevene ville være med. Alle elevene som ble spurt ville være med, men dersom de ikke hadde ønsket å være med, ville jeg ha spurt noen andre. Utenom disse kriteriene ville jeg finne elevene som var gjennomsnittlig i matematikk. Det at man velger elever ut fra at de ikke skiller seg fra normen i stor grad kalles *Typical case sampling* (Clark et al., 2021, s. 379). Mer spesifikt var kriteriene at elevene ikke skulle ha matematikkvansker, og heller ikke holde på med programmering som hobby.

For elevenes anonymitet har jeg gitt dem pseudonymer. Elevenes pseudonymer er Jan, Bjørn, Marit og Eva.

3.3 Metode for datainnsamling

Datainnsamlingen til denne studien ble utført i to undervisningsøkter. I begge øktene ble de samme fire elevene tatt med på et grupperom sammen med meg. Begge øktene ble filmet med videokamera, og det ble tatt lydopptak. I begge øktene fikk elevene utdelt oppgaveark hvor de skulle skrive svarene de kom frem til på oppgavene. Disse oppgavearkene samlet jeg inn etter hver økt. I tillegg skrev jeg ned observasjoner etter øktene blant annet knyttet til det elevene hadde gjort på skjermene, da dette ikke kom så godt med på videoene. Disse observasjonene inneholdt også andre observasjoner som jeg tenkte kunne bli relevant for oppgaven min. Det å reflektere etter at man har observert vil ofte være lurt (Postholm & Jacobsen, 2018b, s. 58). Dette gjelder spesielt dersom man er deltakende observatør, noe som gjør det vanskelig å ta notater underveis (Postholm & Jacobsen, 2018b, s. 58). Det er i dette arbeidet viktig å reflektere og skrive notater innenfor rammene av problemstillingen som danner utgangspunktet for studien (Postholm & Jacobsen, 2018b, s. 58). Derfor noterte jeg ned refleksjoner rundt gjennomføringen i etterkant av hver økt, med utgangspunkt i hvordan elevene brukte programmeringen som et instrument i arbeidet med oppgavene.

3.3.1 Oversikt over innsamlet datamateriale

Tabell 1: Oversikt over innsamlet datamateriale

Innsamlet materiale	Økt 1	Økt 2	Til sammen
Innsamlet oppgavesett	Ett oppgavesett per elev Totalt: 4 oppgavesett	Ett oppgavesett per elev. Totalt: 4 oppgavesett	8 oppgavesett
Videoopptak	Video 1: 23:05 min. Video 2: 13:23 min. Totalt: 36:28 min.	Video 3: 21:08 min. Video 4: 19:58 min. Totalt: 41:06 min.	1:17:34
Lydopptak	Lydopptak 1: 35:53 min.	Lydopptak 2: 28:36 min. Lydopptak 3: 21:40 min. Totalt: 50:16 min.	1:26:9
Transkripsjon	11 sider	17 sider	28 sider
Egne refleksjoner	1 side	1 side	2 sider

Datamaterialet som er samlet inn i dette masterprosjektet vises i tabell 1. I tabellen ser man hva slags materiale som er samlet inn i hver økt, og hvor mye som ble samlet inn totalt. Varigheten på video- og lydopptak varierer på grunn av at øktene ble avsluttet når alle elevene var ferdige med oppgavene, uavhengig av tidspunktet. I økt 2 kan man se at det er stor forskjell mellom videoopptakene og lydopptakene. Dette er fordi kameraet gikk tom for strøm, og jeg derfor bare har lydopptak av den siste delen av økten.

3.3.2 Observasjon

Observasjon er ifølge Adler og Adler (1994) sett på som den mest fundamentale måten å samle inn data på (Gjengitt i Postholm & Jacobsen, 2018, s. 113). Innenfor kvalitativ forskning skjer observasjonene i naturlige situasjoner slik som de utspiller seg (Postholm & Jacobsen, 2018, s. 113). Observasjonene i mitt masterprosjekt vil jeg argumentere for at er hentet fra en tilnærmet naturlig situasjon. Med det mener jeg at elevene kjente hverandre og meg, og utføringen ligner enhver undervisnings-økt hvor de skulle jobbe med oppgaver, selv om elevene ble tatt ut av sin vanlige undervisning og satt på et annet rom enn de var vant til. For at min gjennomføring skulle ligne de undervisningssituasjonene elevene var vant til mest mulig fungerte jeg som lærer i tillegg til forsker/observatør i gjennomføringen av datainnsamlingen. På denne måten kunne jeg veilede elevene på samme måte som læreren deres ville ha gjort i en vanlig time.

3.3.3 Åpen observasjon

Åpen observasjon er motsetningen til strukturert observasjon. I åpen observasjon er problemstillingen relativt åpen, og man er på den måten åpen for hva som vil skje i situasjonen innenfor rammen til et bestemt fokus. I dette masterprosjektet er forskningsspørsmålet «Hvordan kan programmering som instrument bidra til elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur?». Dette forskningsspørsmålet legger opp til at jeg gikk relativt åpent inn i observasjonene for å lete etter eksempler på hvilken måte programmering fungerte som et instrument i elevenes arbeid. På denne måten er observasjonene mine ganske åpne. Det er allikevel viktig å være klar over at observasjonen aldri vil være helt åpen og fri for forforståelse. De tidligere erfaringene og opplevelsene i tillegg til teori som man leser knyttet til det som skal observeres er med på å farge og fokusere forskerblikket (Postholm Jacobsen, 2018b, s. 53). Dermed er det mer riktig å snakke om *hvor* åpen forskningen er og ikke om den er det. Min forskning er ganske åpen, hvor fokuset for oppgaven, altså forskningsspørsmålet, var klart på forhånd, men hvor observasjonen var åpen innenfor dette forskningsspørsmålet. Åpen observasjon er også knyttet til induktiv datainnsamlingsmetode, hvor mer induktiv observasjon innebærer en mer åpen observasjon (Postholm & Jacobsen, 2018b, s. 53). Denne masteroppgaven er i stor grad induktiv, noe jeg kommer inn på senere i kapittelet.

3.3.4 Deltakende observasjon

Gold (1998 henvist i Postholm & Jacobsen, 2018a, s. 115) kategoriserer ulike observatørroller man kan ha i fire kategorier basert på grad av deltakelse og grad av avstand til situasjonen som det forskes på. De fire kategoriene er fullstendig observatør, observatør som deltaker, deltaker som observatør og fullstendig deltaker. I rollen fullstendig observatør har ikke observatøren noen tilknytning til situasjonen som observeres. I rollen som observatør som deltaker er observatøren i hovedsak observatør og deltar ikke i aktiviteten som observeres, men den kan svare høflig på spørsmål fra de som observeres. I rollen som deltaker som observatør er observatøren en del av den situasjonen som observeres, men med større fokus på observasjonen enn dersom man tar rollen fullstendig deltaker. I rollen fullstendig deltaker er observatøren en del av det som observeres, for eksempel en lærer som observerer egen undervisning (Postholm & Jacobsen, 2018a, s. 115).

I gjennomføringen av undervisningsøktene hvor jeg samlet datamaterialet til dette masterprosjektet inntok jeg både rollen som deltaker som observatør og som fullstendig deltaker. Planen før datainnsamlingen ble gjennomført var å beholde rollen som deltaker som observatør. Det var for å kunne observere hva elevene gjorde i timen i høyere grad enn det som er mulig ved fullstendig deltakelse. Grunnen til at jeg tok rollen som fullstendig deltaker var at elevene i arbeidet med enkelte av oppgavene trengte så mye faglig støtte at det å holde avstand til situasjonen som ble observert ikke var mulig dersom elevene skulle få til å gjennomføre alle oppgavene. Jeg fungerte altså som støtte for elevenes læring, da oppgavene var litt for vanskelige for dem å løse uten min støtte. Min støtte i elevenes læring var også et etisk valg. Som NESH (Den nasjonale forskningsetiske komité for samfunnsvitenskap og humaniora) (NESH, 2021, s.22) skriver skal barnets beste alltid være et grunnleggende hensyn i all forskning. Jeg vurderte at elevenes beste i deres lærings situasjon var at de fikk støtte i oppgavene som var for vanskelige. Min forskning skulle altså ikke gå ut over deres behov for støtte i deres læringsprosess. Siden jeg under gjennomføringen av datainnsamlingen i stor grad

var deltakende i situasjonene, som støtte i elevenes arbeid med oppgavene, ble mye av observasjonene gjort i ettertid gjennom å se på videoopptakene fra øktene. Videoopptakene, og transkripsjonene av disse opptakene, ble derfor det viktigste utgangspunktet for analysen.

3.3.5 Observasjonsdata som video

Fordelen med å bruke video i datainnsamlingen er at man samler inn en gjengivelse av situasjonen som ikke er tolket av noen (Tjora, 2021, s. 117). Jacobsen (2003, henvisning i Tjora, 2021, s. 117) hevder at man gjennom videoopptak får en korrekt gjengivelse av situasjonen, men Tjora (2021, 117) peker på at dette kan diskuteres. Han peker på at kameravinkel, hvor mye kameraet fanger opp og kvaliteten på det som tas opp gjør at man må betrakte videomaterialet som en av mange mulige representasjoner av en situasjon. Videodata vil allikevel gi en mer komplett og detaljert beskrivelse av en situasjon enn andre typer observasjon (Tjora, 2021, s. 117). Den viktigste fordelen ved å bruke videoopptak er at man kan se på opptakene i ettertid, og på denne måten kan kontrollere egne inntrykk, gjenoppleve det som ble observert, og i tillegg oppdage nye fenomener som ikke ble oppdaget i observasjonssituasjonen (Tjora, 2021, s. 118). Gjennom videoopptak kan man også bevege seg frem og tilbake i situasjonen, noe som bidrar til at man kan oppdage mindre synlige nyanser i det som skjer i situasjonen, som man ellers ikke ville oppdaget (Tjora, 2021, s. 118). I arbeidet med dette masterprosjektet ga videoene meg muligheten til å gå gjennom materialet flere ganger, og til å oppdage nye fenomener som jeg ikke ville oppdaget uten videoene.

3.4 Utforming av oppgavene til datainnsamlingen

Opgavene som danner utgangspunktet i datainnsamlingen til dette masterprosjektet, ble utformet i samarbeid med veilederne til masteroppgaven og praksislærer. Jeg laget oppgavene selv, men endret utformingen en del ut fra tips jeg fikk fra veilederne og praksislærer. Oppgavene ble utformet med utgangspunkt i å finne ut noe om elevenes bruk av programmering som et instrument i arbeid med oppgaver som omhandler tallfølger med multiplikativ struktur.

Under datainnsamlingen jobbet elevene hver for seg, med hver sitt oppgaveark og hver sin Chromebook, men de ble oppfordret til å snakke sammen dersom de trengte hjelp med oppgavene. Elevene fikk utdelt ett og ett ark med oppgaver under gjennomføringen. De fikk ikke utdelt neste ark før de sa seg ferdig med oppgavene på det foregående arket.

3.4.1 Beskrivelse av oppgavene som ble brukt

Til sammen jobbet elevene med 11 oppgaver i løpet av begge gjennomføringene av innsamling av datamateriale. Oppgave 1-5 ble jobbet med av elevene i løpet av den første økten, mens oppgave 6-11 ble jobbet med av elevene i løpet av den andre økten. Alle oppgavene omhandler tallfølger som er en del av «åtte-gangen» og «sju-gangen», altså tallfølgene 8, 16, 32, 40, 48, ... og 7, 14, 21, 28, 35, Noen av oppgavene hadde mye høyere tall, men alle tallene var en del av disse tallfølgene.

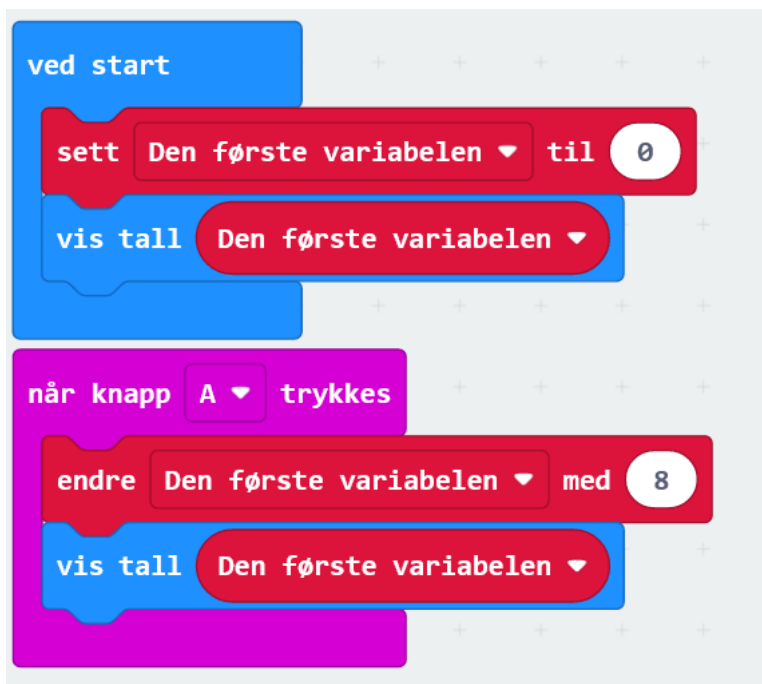
Fokuset for den første økten var blant annet å undersøke om elevene i arbeid med tallfølger i programmering oppdager at tallfølgene har en multiplikativ struktur. For å undersøke om elevene oppdager at tallfølger med de lavere tallene er multiplikasjon og ikke tallfølger med de høyere tallene, innebar flere av oppgavene også tall som var

høyere enn 80 og 70. Hypotesen min var at elevene ville oppfatte oppgaver med tall opp til 80 og 70 som multiplikasjonsoppgaver, fordi de hadde jobbet med disse tallene som multiplikasjon før, men at de kanskje ikke ville se på de tallene over ti-gangen som multiplikasjon da dette ikke var noe de var vant til. I tillegg var formålet med oppgavene å se hvordan elevene brukte programmeringen som verktøy når de arbeidet med oppgaver som innebærer tallfølger med multiplikativ struktur.

Den andre økten ble utviklet som en videreføring av det elevene jobbet med i den første økten. Oppgaveformuleringene ble endret i et forsøk på å hjelpe elevene til oppdage at arbeidet med disse oppgavene er arbeid med multiplikasjon.

Oppgave 1

Figur 12: programmet i oppgave 1



I den første oppgaven fikk elevene oppgitt koden som er vist i figur 12. Så fikk de spørsmål om hva de trodde kom til å skje dersom knapp A ble trykket på, først en, så seks ganger. Denne oppgaven hadde som formål å sjekke om elevene forsto koden, og også gi elevene muligheten til å utvikle en forståelse for koden. Det var viktig at elevene forsto koden siden denne typen kode var grunnlaget for alle oppgavene gjennom begge øktene. I tillegg fikk elevene til slutt i den første oppgaven et spørsmål om de kunne skrive et regnestykke til at de trykker seks ganger på knapp A. Dette spørsmålet var valgt for å sjekke om elevene tenkte på denne typen oppgave som multiplikasjon. Dersom de tenkte på denne sammenhengen som multiplikasjon var min hypotese at de ville skrive denne sammenhengen som $6 \cdot 8 = 48$, eller en variasjon av dette.

Oppgave 2

I oppgave to skulle elevene lage koden som ble gitt i oppgave 1, og sjekke hva som skjedde dersom de trykte først én gang, så 6 ganger. Formålet med denne oppgaven var at elevene skulle sjekke om de hadde tenkt riktig i arbeidet med oppgave 1. Samtidig var tanken at elevene skulle bli kjent med hvordan programmet og koden fungerte, for å lettere kunne bruke det som et verktøy i arbeidet med de neste oppgavene.

Oppgave 3

I denne oppgaven skulle elevene få micro:bit til å vise tallfølgen 88, 96, 104, 112, 120 ved å bruke variabler. Målet med denne oppgaven var å undersøke om elevene like lett ville løse oppgaver med tall som er større enn åtte multiplisert med ti, som tallene under åtte multiplisert med ti. I tillegg ville jeg undersøke om elevene knyttet så store tall til multiplikasjon, siden de ikke hadde jobbet så mye med multiplikasjon med tall over «ti-gangen» før.

Oppgave 4

I denne oppgaven skulle elevene beskrive tallfølgen som ble gitt i oppgave 3 skriftlig, og finne de tre neste tallene. Denne oppgaven var en utvidelse av oppgave 3, og hadde som formål å undersøke om elevene ville beskrive tallfølgen som multiplikasjon med åtte. Gjennom å beskrive tallfølgen som alltid øker med åtte hver gang, ønsket jeg også at noen av elevene kunne oppdage at det de jobbet med faktisk er multiplikasjon, slik at de også kunne tenke på de neste oppgavene som multiplikasjon.

Oppgave 5

I denne oppgaven fikk elevene oppgitt tallfølgen; ..., 21, 28, 35, 42, ..., 77, 84, 91, Elevene skulle se på hvor mye det var mellom tallene i tallfølgen, og finne ut hvilket tall som må stå i «endre variabel»- blokk for å få micro:biten til å vise denne tallfølgen. I tillegg skulle elevene beskrive tallfølgen.

Formålet med denne oppgaven var å sjekke om elevene klarte å bruke det de hadde gjort på de tidligere oppgavene, hvor de skulle finne mønsteret i en tallfølge bare med andre tall. Formålet var også at elevene skulle overføre strategiene de forhåpentligvis ville utvikle i arbeid med de tidligere oppgavene til å løse denne oppgaven som hadde lik struktur med andre tall. I tillegg ville jeg sjekke i hvilken grad elevene faktisk klarte å finne mønsteret i en tallfølge når økningen ikke var gitt, slik som den var i de tidligere oppgavene. Det at jeg valgte å ha med tall både over og under syv multiplisert med ti, var for å få et inntrykk av om elevene ville se på de ulike delene av tallfølgen på en annen måte når tallene var over eller under 70.

Oppgave 6

Denne oppgaven var den første oppgaven i den andre økta, og det var over en uke siden de hadde jobbet med programmering. Oppgaven var at elevene skulle få micro:biten til å vise tallfølgen 8, 16, 24, 32, 40, Formålet med denne oppgaven var å hjelpe elevene til å huske det vi jobbet med den første økten, og å repetere noe av det vi gjorde første økten.

Oppgave 7

I denne oppgaven fikk elevene oppgitt samme kode som i oppgave 1, som vises i figur 12. Siden denne oppgaven tok utgangspunkt i samme kode som oppgave 1, likner oppgavene veldig. Forskjellen var at spørsmålene videre i oppgaven er snudd. Altså spurte denne oppgaven etter hvor mange ganger man må trykke på knapp A for å få 40 og 72, og ikke hva man får dersom man trykker x antall ganger på knapp A, slik som i oppgave 1. Denne endringen i oppgaveformulering ble gjort fordi det i oppgave 1 var litt for lett for elevene å bare bruke programmet til å sjekke hva riktig svar var, og ikke faktisk tenke matematisk. Jeg så også etter første økt at elevene i stor grad benyttet seg av gjentatt addisjon for å løse oppgaven, og jeg håpet at denne formuleringen ville legge

opp til at elevene skulle måtte benytte seg av andre strategier enn gjentatt addisjon for å løse oppgaven.

I løsningen av denne oppgaven legger opp til divisjon like mye som multiplikasjon ved at elevene kunne tenke 40 delt på 8 blir 5, og 72 delt på 8 er 9. På denne måten er dette en divisjonsoppgave i større grad enn en multiplikasjonsoppgave. Men siden multiplikasjon og divisjon er motsatte operasjoner kunne elevene også bruke multiplikasjon til å løse denne oppgaven ved å si at man må trykke 5 ganger på knapp A for å få 40 siden $5 \cdot 8 = 40$, og 9 ganger på knapp A for å få 72 siden $9 \cdot 8 = 72$. På denne måten håpte jeg at elevene skulle knytte denne oppgaven til multiplikasjon, siden det er fokuset for mitt masterprosjekt.

Til slutt i oppgave 7 skulle elevene lage koden i MakeCode, og sjekke om svaret var riktig. Laging av koden i dette tilfellet innebar å kopiere koden fra oppgavearket, da den var oppgitt. De skulle også forklare hvorfor svaret eventuelt ikke ble riktig. Dette var for å legge opp til refleksjon knyttet til hva som eventuelt gikk galt i deres løsning av de første delene av oppgaven.

Oppgave 8

Denne oppgaven liknet på oppgave 7, men hadde et tall over åtte multiplisert med ti. Oppgaven var at elevene skulle bruke koden gitt i oppgave 7 og oppgave 1, se figur 12, og finne ut hvor mange ganger man må trykke på knapp A for å få 88. Tallet 88 ble valgt for å utfordre en vanlig oppfatning av at 8-gangen stopper ved $8 \cdot 10$. I tillegg ville jeg bruke et relativt høyt tall slik at det ville være vanskeligere for elevene å bare trykke seg fram til svaret ved å trykke på knapp A, uten å tenke ut hva de trodde var det riktige antallet ganger.

Oppgave 9

Denne oppgaven tok fortsatt utgangspunkt i koden i figur 12. Oppgaven var «Hvor mange ganger må du trykke på knapp A for å få micro:biten til å vise tallene 808, 816, 824, 832, ... ? Kan du vise denne sammenhengen i et regnestykke?» Valget av så høye tall ble gjort fordi det ville gjøre det vanskelig, om ikke umulig, for elevene å bare trykke seg frem til svaret. På denne måten måtte elevene tenke på en annen måte og finne en annen strategi for løsningen av oppgaven, og jeg håpet at denne løsningsstrategien ville innebære multiplikasjon på en eller annen måte. For at elevene skulle kunne løse denne oppgaven på en enkel måte måtte elevene ha forkunnskaper om at $8 \cdot 100 = 800$, noe jeg forventet at i det minste noen av dem ville ha.

Denne oppgave ba også elevene om å vise sammenhengen mellom tallene i oppgaven som et regnestykke. Dette var for å undersøke om elevene klarte å eksplisitt knytte deres arbeid med oppgaven til arbeid med multiplikasjon.

Oppgave 10

Figur 13: Koden i oppgave 10



I denne oppgaven fikk elevene utdelt koden vist i figur 13. Oppgaven var hvor mange ganger knapp A må trykkes for å få 14 og 42. Denne oppgaven er lik oppgave 7, men med en endring på sju i stedet for åtte ved å trykke på knapp A. Formålet med denne oppgaven var å undersøke om elevene ville klare å løse oppgaver som er lik tidligere oppgaver, bare med et annet tall. I tillegg ville jeg undersøke om elevene har forstått koden, og hvordan koden henger sammen med denne typen oppgaver. Oppgaven hadde altså som formål å undersøke om elevene på de tidligere oppgavene bare hadde trykt tilfeldig, eller om de har forstått noe om bruken av programmeringen som en artefakt.

Oppgave 11

Her ble samme kode som figur 13. brukt. Elevene skulle finne ut hvor mange ganger knapp A må trykkes for å få 714. Denne oppgaven likner på oppgave 9, bare at denne tar utgangspunkt i tallene i sju-gangen, ikke åtte-gangen. Jeg ville undersøke om elevene klarte å overføre kunnskapen de brukte for å løse oppgave 9, til en lik oppgave med andre tall.

3.5 Analysemetode

Analysen i et kvalitativt forskningsprosjekt starter med en gang forskeren begynner å lese dokumenter, drar ut i forskningsfeltet eller starter å intervju informanter. Hvordan det første materialet som samles inn forstås er viktig for hvordan datainnsamlingen fortsetter (Postholm & Jacobsen, 2018, s. 145). Siden jeg utførte innsamlingen av datamaterialet i to økter med over en uke imellom, utførte jeg en foreløpig analyse av observasjonene etter første datainnsamlingsøkt. Denne analysen baserte seg bare på observasjoner jeg hadde fra timen, sammen med elevenes svar på oppgavene på papir. Grunnen til det er at jeg ikke hadde tid til å transkribere videoopptak og lydopptak mellom øktene. Den foreløpige analysen etter første økten dannet grunnlaget for de endringene som ble gjort i oppgavene før den andre økten. I denne foreløpige analysen

mellom øktene var utgangspunktet i hvor stor grad jeg kunne se at elevene brukte programmeringen som et instrument i arbeidet med oppgavene, og om elevene knyttet dette arbeidet til multiplikasjon.

Det å kategorisere og kode datamaterialet er kjerneaktiviteter i analyseprosessen ved en kvalitativ analyse (Nilssen, 2012, s. 78). Hensikten med denne analysen er først og fremst å sortere datamaterialet som er samlet inn gjennom en studie for å kunne gjøre materialet forståelig (Postholm & Jacobsen, 2018, s. 139). I analysen av datamaterialet i dette masterprosjektet har jeg gjennomført en åpen koding. Denne analysemetoden baserer seg på en induktiv tilnærming (Clark et al., 2021, s. 526). Det vil si at den ikke baserer seg på forhåndsdefinerte kategorier for å dele inn datamaterialet i. For å kunne gjennomføre den åpne kodingen måtte jeg først transkribere datamaterialet for å få oversikt, og for å kunne se hvilke deler som var relevant for problemstillingen min. Videre ble transkripsjonene delt inn i sekvenser som hadde noe med programmering eller multiplikasjon å gjøre, før disse sekvensene ble delt inn i kategorier. Disse sekvensene og kategoriene blir nærmere beskrevet under åpen koding.

3.5.1 Transkripsjon

Som nevnt tidligere var videoopptakene mine viktigste innsamlede datamateriale. Jeg gjorde også lydopptak under gjennomføringen av datainnsamlingen, men disse ble bare brukt for å fylle ut lyden fra videoopptakene i tilfeller hvor lyden var dårlig. For å få oversikt over innholdet i videoopptakene, og lettere kunne sortere det som skjedde i situasjonene, gjennomførte jeg transkripsjon av videoene. I transkripsjon blir handlinger og tale gjort om til tekst (Nilssen, 2012, s. 46). I transkripsjonen gjør forskeren mange valg, for det første fordi forskeren har en tolkning av hva som er sett på som viktig. For det andre vil tonefall, mimikk, gester og liknende ikke komme naturlig frem når man gjør handlinger og kommunikasjon om til tekst (Nilssen, 2012, s.46). For å minske det som blir borte i transkribering av video er det viktig å få med pauser, om det som blir sagt for eksempel er spørrende eller nølende, om det legges trykk på enkelte ord og eventuelle gester, blick, mimikk og øyekontakt (Nilssen, 2012, s.49). Dette er noe jeg prøvde å få med i min transkripsjon, men tidlig fant ut at ikke var praktisk mulig å få med så mye av. Min transkripsjon har derfor mest fokusert på å få frem det elevene sa i arbeidet med oppgavene, og bare litt på tonefall, mimikk og ikke-verbale uttrykk. For eksempel ble det notert i transkripsjonen når elevene tok lange pauser, nølte veldig med å si det de sa og når de pekte på Chromebook eller oppgavearket. Mindre pauser, andre bevegelser og nyanser i tonefall ble ikke notert.

Transkribering burde gjøres av forskeren selv (Nilssen, 2012, s. 47). En grunn til det er at transkribering er en viktig del av analyseprosessen. Gjennom transkriberingen dukker det opp nye tanker om det innsamlede datamaterialet, og ideer til koding kan også komme i transkriberingsprosessen. En annen fordel med å transkribere selv er at man blir skikkelig kjent med datamaterialet. En tredje fordel er at det noen ganger er en fordel å kjenne konteksten til situasjonen man skal transkribere (Nilssen, 2012, s.48). Jeg transkriberte selv, og føler at dette var en viktig start på analysen av datamaterialet, ved at jeg fikk en oversikt over hva som faktisk fantes i datamaterialet mitt. Denne oversikten var viktig i min oppgave siden jeg analyserte ved hjelp av åpen koding, noe jeg kommer tilbake til i avsnittet om analysemetode.

Transkripsjonen av videoopptakene ble til 28 sider med utsagn og noen kommentarer om mimikk, tonefall og andre ikke-verbale uttrykk. Utdrag fra transkripsjonene blir presentert i resultatkapittelet sammen med en beskrivelse av konteksten, for eksempel hvilken

oppgave elevene jobbet med. Observasjoner som ikke er direkte tale er skrevet slik *Kommentar*. Steder hvor jeg har fjernet utsagn eller deler av utsagn på grunn av at de ikke var relevant for det funnet som det skrives om er markert med [...]. Elevenes utsagn er gjort om til standardisert bokmål for å lette leseligheten og å gjøre transkriberingsprosessen enklere. I tillegg kan det å unnlate dialektuttrykk være et tiltak for å sikre anonymiteten til deltakerne (Tjora, 2021, s. 186). Dette vil spesielt være aktuelt dersom en eller flere deltakere bruker spesielle dialektuttrykk.

Før videre analyse ble alle kommentarer og handlinger som ikke var knyttet til arbeid med matematikkoppgaver tatt bort. For eksempel snakket elevene om hva de skulle ha i neste time, og om gjenstander de så i rommet. Slike utsagn ble tatt bort fra transkripsjonen før videre analyse av datamaterialet. Dette var for å gjøre transkripsjonene mer ryddige når analysen skulle gjennomføres, uten at noe relevant ble tatt bort.

3.5.2 Inndeling i sekvenser

Etter å ha transkribert ferdig og kortet ned mengden datamateriale til å bare inneholde elevenes arbeid med programmeringen og oppgavene, var neste steg å finne ut hva det kunne fortelle meg. I dette steget var målet å finne mening i datamaterialet, som fortsatt var på mange sider med transkripsjon. For å kunne klare å finne noe i datamaterialet måtte mengden transkripsjon kortes ned. For å korte ned transkripsjonene startet jeg med å lese gjennom alle sidene med tekst, og dele materialet inn i sekvenser. Sekvenser er inndelinger av transkripsjonen som avgrensar deler av transkripsjonen der én type ting skjer (Nilssen, 2012, s. 106). For å dele inn i sekvenser begynte jeg med å lese gjennom alle sidene med transkripsjon, for å markere hva som kunne være interessant for forskningsspørsmålet mitt. Det vil si at alt i transkripsjonene som handlet om elevenes bruk av programmering som verktøy og alt som omhandlet multiplikasjon ble understreket. Gjennomgangen av all transkripsjon ble gjort flere ganger for å være sikker på at jeg fikk med meg alt som kunne brukes i videre analyse. I tillegg ble elevenes skriftlige arbeid og observasjonsnotatene mine brukt til å finne de delene av transkripsjonene som var relevante og til å gi mere informasjon til enkelte av sekvensene om konteksten rundt utsagnene. Sekvensene som jeg understreket var både enkeltutsagn fra elever, og korte samtaler mellom elevene og mellom lærer og elever. Jeg forsøkte å lage sekvenser som var lengere enn bare enkeltutsagn, men i transkripsjonene var det mange utsagn fra elevene som ikke hang sammen med noen av de andre utsagnene, som allikevel var interessante med tanke på forskningsfokuset mitt. Alle sekvensene ble kopiert inn i et eget dokument for å kunne analyseres nærmere. Dette gav 50 sekvenser hvor 33 var enkeltutsagn og 17 sekvenser med flere enn ett utsagn. Videre i analysearbeidet ble bare disse sekvensene brukt, mens resten av datamaterialet ble valgt bort. Dermed satt jeg igjen med bare sekvenser som handlet spesifikt om elevenes bruk av programmering som verktøy og elevenes arbeid med multiplikasjon.

3.5.3 Åpen koding

Som nevnt tidligere brukte jeg en *induktiv tilnærming* til analysen. I induktiv tilnærming til analyse er teori utfallet av analysen, og ikke det som styrer analysen (Clark et al., 2021, s. 20). Utfordringen med kvalitative studier med en induktiv tilnærming er å redusere, forenkle og finne mening ut fra den komplekse virkeligheten (Nilssen, 2012, s. 82). Koding er det første steget i prosessen med å redusere en stor mengde datamateriale til noen få kategorier eller temaer som fanger det viktigste i materialet

(Nilssen, 2012, s. 82). I åpen koding skal man møte datamaterialet med et åpent sinn og en åpen holdning til hva datamaterialet forteller deg (Nilssen, 2012, s. 78). Selv om fullstendig åpenhet er det man jobber mot, er det aldri mulig å være helt åpen (Nilssen, 2012, s. 78). Det er fordi forskeren alltid vil ha med seg erfaringer og verdier som påvirker hva som blir sett på som viktig forskningen.

Gjennom den åpne kodingen prøvde jeg å finne svar på forskningsfokuset mitt, som er å se på hvordan programmering som et instrument kan bidra til elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur. I arbeidet med inndelingen i sekvenser hadde jeg gjort en grovanalyse. Videre i kodingsprosessen så jeg derfor på disse sekvensene, og prøvde å finne mønster i sekvensene, og noen koder som de kunne deles inn i. Først lagde jeg koder for alt som kunne være relevant, og endte opp med for mange koder til å kunne skrive om alle. Videre analyse av disse kodene førte til at jeg satt sammen noen av kodene til nye koder, og at noen av kodene ble tatt bort. Kodene som ble satt sammen til nye koder handlet om det samme temaet, og de kodene som ble tatt bort viste seg å ikke være relevant for forskningsfokuset mitt. Etter disse sammenslåingene av koder og å ha valgt bort noen av kodene hadde jeg sju kategorier. Videre analyse gjorda at jeg fant ut at disse kategoriene kunne plasseres som underkategorier i to hovedkategorier. Hovedkategoriene ble da «Elevenes bruk av programmering» og «Elevenes arbeid med multiplikasjon». I tabell 2: er hovedkategoriene og underkategoriene beskrevet.

Tabell 2: Kategorier for funn i analysen

Navn på hovedkategorier	Navn på underkategorier	Beskrivelse
Bruk av programmering	Bruker programmeringen til å utforske store tall.	Elevene utforsker store tall, uten at det var gitt i oppgaven.
	Bruker egenlagde koder som ikke var hensiktsmessig i løsning av oppgavene.	Elevene møter på utfordringer i løsningen av oppgaver, på grunn av at de har laget en kode som ikke kunne brukes.
	Bruker programmeringen til å sjekke om det de tenkte var riktig.	Elevene gjør oppgaven uten bruk av programmet, og bruker programmet til å sjekke om det de tenkte er riktig.
	Utfordringer med å bruke programmeringen i arbeid med store tall.	I arbeid med store tall hadde elevene vanskeligheter med å bruke programmeringen som et instrument i løsningen av oppgavene.
Arbeid med multiplikasjon	Bruk av gjentatt addisjon.	Elevene bruker gjentatt addisjon i arbeidet med oppgavene.
	Utfordring med å bruke multiplikasjon i løsningen av oppgavene.	Elevene nevner ikke multiplikasjon i arbeidet med oppgavene, og bruker ikke multiplikasjon i utregningen av oppgavene.
	Bruker oppgavearket eller programmet til å løse oppgavene på en enklere måte enn multiplikasjon.	Elevene bruker oppgavearket eller programmeringen til å løse oppgavene på en enkel måte, som ikke krever bruk av multiplikasjon.

3.6 Etske betraktninger

Etske betraktninger er en essensiell del av forskningsprosessen (Clark et al., 2021, s. 107). Når man forsker på barn dukker det opp flere etske betraktninger, som for eksempel knyttet til om barn har kapasitet til informert samtykke (Clark et al., 2021, s. 128; Den nasjonale forskningsetiske komite for samfunnsvitenskap og humaniora, 2021, s. 22). En løsning på dette er å be foresatte om skriftlig samtykke, noe jeg også har gjort knyttet til dette masterprosjektet.

Dersom man skal utføre forskning hvor det skal behandles personopplysninger som navn, stemme eller bilde, må man melde forskningen inn til SIKT (tidligere NSD). Dette er blant annet for å sikre etterlevelse av personvernreglementet (SIKT, u.å.). I mitt masterprosjekt har jeg brukt både lyd- og video-opptak. Dette er personopplysninger, og prosjektet måtte derfor meldes inn til Sikt. Studentene i praksisgruppa samarbeidet med å sende felles søknad, og fikk godkjenning før vi startet innsamlingen. For å få godkjenning ved innmelding av forskningsprosjekt hos Sikt, er det blant annet fokus på lagring av sensitiv informasjon, for å sikre at ikke personopplysninger kommer på avveie. Lagringen av personopplysninger til dette masterprosjektet ble gjort på passordbeskyttet skylagring hos NTNU. Innsamlingen av datamaterialet foregikk på kamera og lydopptakere lånt av NTNU, og deretter raskt overført til skylagringen og permanent slettet fra kamera og lydopptaker. På denne måten var det ikke mulig for noen andre å få tak i personopplysninger knyttet til elevene, og deres anonymitet og personvern opprettholdes.

Ifølge Diener og Crandall (1978, gjengitt i Clark, et al., 2021, s. 113) kan etiske betraktninger rundt forskning deles inn i 4 hovedområder. Disse områdene er; skade på deltakere, mangel på informert samtykke, invasjon av privatlivet og bedrag. Disse områdene handler i hovedsak om brudd på etiske prinsipper. De fire områdene overlapper i mange tilfeller, for eksempel ved at mangel på informert samtykke kan tenke seg å i enkelte tilfeller innebære bedrag av deltakerne.

Skade på deltakerne i forskningen kan bety ulike ting, enten fysisk skade, hemming av utvikling, tap av selvtillit, stress og det å få subjektene til å utføre forkastelige handlinger (Diener & Crandal, 1978, i Clark et al., 2021, s.113). I forhold til fysisk skade er rammene for mitt masterprosjekt satt i skolesammenheng, hvor det er lagt opp et trygt miljø rundt elevene. Hemming av utvikling kan knyttes til elevenes faglige utvikling. Dette har jeg prøvd å motvirke ved at elevene skulle tas ut av matematikktimer, slik at de ikke fikk veldig annerledes undervisning enn de andre elevene i klassen. I tillegg var dette punktet noe som ble vurdert i forhold til lengden på øktene, som jeg fant ut at ikke skulle gå over en av elevenes vanlige økter, slik at de ikke skulle gå glipp av viktig undervisning for å delta i prosjektet. I forhold til tap av selvtillit og stress var det viktig i studien å understreke for elevene at det ikke var viktig hvor mye de fikk til av oppgavene, og at det ikke var de som skulle *testes* men opplegget som jeg hadde laget. På denne måten kunne elevene fokusere på å gjøre så godt de kunne uten å føle seg dårlig eller stresse på grunn av det de ikke fikk til. Det å utføre forkastelige handlinger var aldri aktuelt knyttet til dette masterprosjektet.

En viktig del av det å forebygge skade på deltakerne i forskning er å holde identiteten og innsamlet datamateriale knyttet til deltakerne konfidensielle (Clark et al., 2021, s. 113). For å sikre deltakernes anonymitet har jeg som nevnt tidligere gitt deltakerne pseudonymer. I tillegg er identifiserende ord, som navn på andre de kjenner og navn knyttet til skolen eller området de befant seg i, endret i anonymiseringen av datamaterialet. Det ubehandlede datamaterialet, video og lydopptak, ble lagret kun på sikker server hos NTNU, slik at ingen andre enn meg skulle kunne se disse filene. På denne måte kunne jeg sikre deltakernes anonymitet på en god måte.

For å motvirke mangel på informert samtykke er et informasjonsskriv som forklarte prosjektene sendt til foresatte i klassen hvor vi skulle utføre forskningen til våre masterprosjekt (Se vedlegg 3). På denne måten hadde foreldrene muligheten til å gjøre en mest mulig informert beslutning om dette var noe de ville at barnet deres skulle være med på. I tillegg ble elevene fortalt muntlig om hva prosjektet handlet om, og de ble

spurt om de ville delta. På denne måten hadde også elevene mulighet til å takke nei til å være med dersom de ønsket det.

Invasjon av privatlivet henger tett sammen med informert samtykke, da deltakerne gjennom informert samtykke får vite hva forskningen innebærer, og hvilke typer private tema som kan komme gjennom deltakelse (Clark et al., 2021, s. 124). Allikevel må en som forsker tenke over hvordan spørsmål som blir stilt kan virke for privat å snakke om, selv om datamaterialet blir behandlet konfidensielt. Temaene i mitt masterprosjekt er ikke knyttet til områder som er ansett som private, som for eksempel inntekt, seksualitet eller resultater i skolen.

3.6.1 Troverdighet og transparens

For at forskningen skal bli oppfattet som troverdig, er det viktig at hele forskningsprosessen synliggjøres (Nilssen, 2012, s. 141). En kvalitativ studie vil aldri kunne gjentas på akkurat samme måte en gang til (Nilssen, 2012, s. 141). Det er fordi mennesker vil opptre annerledes i møte med ulike mennesker, og fordi mennesker utvikler seg over tid (Nilssen, 2012, s. 141). Det gjør at denne masterstudien ikke ville gitt samme resultat dersom den ble gjennomført med andre elever, eller på et annet tidspunkt. Det er mange faktorer som påvirker det resultatet jeg fikk i min masterstudie, som elevenes dagsform, deres erfaringer og elevenes faglige utvikling. I tillegg kan elevenes arbeid med oppgavene ha blitt påvirket av at deres arbeid ble filmet og at øktene ble utført med en student som de ikke hadde hatt undervisning med så mye tidligere. Selv om det derfor ikke er mulig å kopiere og utføre denne studien å få likt resultat, håper jeg at studien min gjennom åpenhet rundt prosessen kan bli oppfattet som troverdig innenfor konteksten den er utført. Denne åpenheten innebærer det å være transparent gjennom hele forskningsprosessen (Nilssen, 2012, s. 140), altså å synliggjøre alt som er gjort og tenkt underveis i forskningsprosessen. Jeg har prøvd å være transparent gjennom dette metodekapitlet ved å forklare konteksten, og valgene jeg har gjort i forskningen og i analysen i detalj, slik at de som leser om dette masterprosjektet vet rammene og begrensningene som finnes i denne studien. Transparens kan også støtte overførbarhet, ved at de som leser studien kan vurdere relevansen ut fra sin egen kontekst, og vurdere om de vil prøve ut lignende undervisningsopplegg (Postholm & Jacobsen, 2018a, s. 238).

4 Resultater

Fokuset for dette masterprosjektet er hvordan programmering fungerer som et instrument i arbeid med tallfølger med multiplikativ struktur. Gjennom åpen koding har jeg fokusert på hvordan elevene bruker programmering som artefakt, og hvordan de ser en sammenheng til arbeid med multiplikasjon, og kommet frem til to hovedkategorier for funn:

- Bruk av programmering
- Arbeid med multiplikasjon

Innenfor disse hovedkategoriene har jeg laget åtte underkategorier for funn:

- Bruker programmeringen til å utforske store tall.
- Bruker egenlagde koder som ikke var hensiktsmessig i løsning av oppgavene.
- Bruker programmeringen til å sjekke om det de tenkte var riktig.
- Utfordringer med å bruke programmeringen i arbeid med store tall.
- Utfordringer med forståelsen for programmeringen som hinder for løsningen av oppgavene.
- Bruk av gjentatt addisjon.
- Elevene bruker oppgavearket eller programmet til å løse oppgavene på en enklere måte enn multiplikasjon.
- Elevenes utfordring med å bruke multiplikasjon i løsningen av oppgavene.

4.1 Bruk av programmering

Kategorien *bruk av programmering* innebærer funn som viser elevenes ulike måter å ta i bruk programmeringen. Formålet til denne kategorien er å se på hvilke måter programmeringen ble brukt som et instrument i elevenes arbeid med oppgavene. I arbeidet med oppgavene brukte elevene programmeringen på ulike måter.

Måtene elevene valgte å bruke programmeringen på er delt inn i underkategoriene (1) bruker programmeringen til å utforske store tall, (2) bruker egenlagde koder som ikke var hensiktsmessig i løsning av oppgavene, (3) bruker programmeringen til å sjekke om det de tenkte var riktig og (4) utfordringer med å bruke programmeringen i arbeid med store tall.

4.1.1 Bruker programmeringen til å utforske store tall

Elevene bruker programmeringen til å prøve å få størst mulig tall. Gjennom analysen kom det frem flere utsagn fra elevene, mest fra Bjørn, som kommenterer hvor høye tall de har fått programmet til å vise. For eksempel i utdrag 1 under hvor Bjørn og Jan jobber med oppgave 2, men også snakker om tall som ikke var relevant for oppgaven.

Utdrag 1

Bjørn: 8. 1, 2, 3, 4, 5, 6. Jeg hadde rett. Eller *peker på Jan* oi oi. 64,

Jan: 160, 184

Bjørn: Hundre og *Trykker gjentatte ganger på Chromebook* 104

Bjørn: Jeg skal gjøre de andre oppgavene imens. Du bare trykker dritmasse så gjør vi de andre oppgavene. ...

I utdraget ser man at elevene på eget initiativ begynner å trykke masse på knappen i editoren, for å få høyere tall. Dette synes også i andre enkeltutsagn elevene kom med når de jobbet med ulike andre oppgaver, som man kan se under. Jeg har valgt å ikke ta med samtalene som foregikk rundt disse utsagnene, da disse utsagnene var kommentarer som ikke hang sammen med andre utsagn.

Utdrag 2

Bjørn: 600

Eva: Jeg har kommet meg til 1000. Jo sjekk, en null, null, null.

Bjørn: *Jubler når han har fått et høyt tall på skjermen*

Bjørn: Må bare komme til tusen.

Bjørn: Okay, snart tusen.

Bjørn: 192, endelig. Og så 200.

Bjørn: Snart er jeg på 500. 500, halvveis til 1000!

Bjørn: 1000! Jeg kom til 1000.

Jan: Jeg er på 808.

Utdrag 1 og 2 viser et utvalg funn i analysen hvor elevene nevner store tall som ikke har sammenheng med oppgavene de jobbet med. Det viser hvordan elevene, spesielt Bjørn, brukte programmeringen til å utforske store tall, uten at det var formålet med oppgaven.

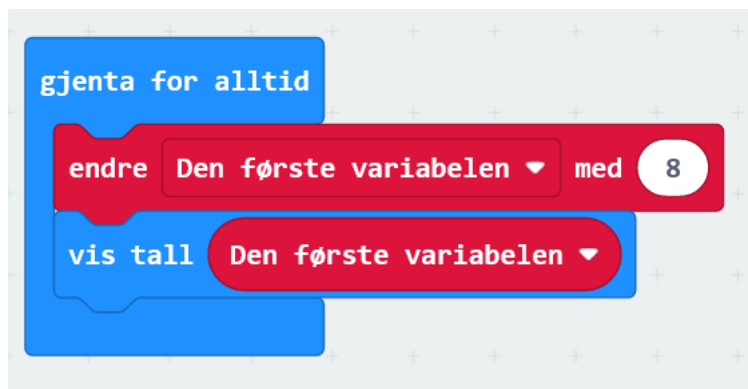
4.1.2 Bruker egenlagde koder som ikke var hensiktsmessig i løsning av oppgavene

Den første oppgaven elevene fikk i den andre økten var oppgave 6, hvor de skulle lage en kode for å få micro:bit til å vise en tallfølge, denne oppgaven er nærmere beskrevet i delkapittel 3.5.1. I analysen fant jeg at de fleste elevene i løsningen av denne oppgaven ikke løste oppgaven slik jeg hadde tenkt. Som nevnt i delkapittel 3.5.1 var formålet med denne oppgaven å hjelpe elevene til å huske hva vi hadde jobbet med i den første økten. Flere av elevene husket ikke det vi jobbet med i den første økten, og de endte opp med andre koder enn den vi jobbet med første økt. Koden som to av elevene lagde var ikke hensiktsmessig å bruke i arbeidet med de neste oppgavene, og elevene valgte å endre koden til den som ble gitt i oppgave 7, for å kunne bruke koden videre.

Bjørn endret ikke koden sin for å følge det som sto på oppgavearket. Under har jeg tatt med et eksempel hvor Bjørn møter en utfordring når han bruker sitt egenlagde program for å løse oppgave 7.

Dette utsagnet kom fra Bjørn mens han jobbet med å sjekke hvor mange ganger man må trykke på knapp A for å få 72, altså en del av oppgave 7. Bjørn brukte koden som er vist i figur 14. Denne koden gjør at micro:bit, eller simuleringen av micro:bit, viser tallene i åttegangen fra 8 og oppover, helt til den blir stoppet ved å trykke på stoppknappen i editoren til MakeCode.

Figur 14: Bjørns kode



Det at endringen av variabelen vil fortsette uendelig, eller til den blir stoppet vet å trykke på stoppknappen, i motsetning til koden jeg hadde laget hvor endringen bare skjer hver gang knapp A blir trykt, skaper utfordringer. Det ser vi i utsagnet:

Bjørn: jeg kommer til å, å nå så jeg på den *ser på arket til Jan* 5, 48, 56, nei, jo 48, 56 eh, 64, 72, 72, åja, da er det ni ganger, eller 10? *teller på fingrene*

Før dette utsagnet har Bjørn kjørt programmet sitt for å sjekke hvor mange ganger endringen skjer for å få 72. Allikevel ser vi i utsagnet at han er usikker på hvor mange ganger han må trykke. Det kan være på grunn av at koden kjører automatisk slik at det er vanskeligere å telle hvor mange ganger endringen skjer.

4.1.3 Bruker programmeringen til å sjekke om det de tenkte var riktig

Gjennom analysen fant jeg flere eksempler på at elevene brukte programmet til å sjekke om svaret deres var riktig. Både oppgave 2 og oppgave 7 legger opp til at elevene skal bruke MakeCode, mer spesifikt de kodene som er gitt i oppgavene, til å sjekke om det de hadde tenkt var riktig. I løsningen av oppgave 2 og oppgave 7 brukte alle elevene programmeringen til å sjekke om det de hadde gjort var riktig, enten de hadde gjort det før de kom til oppgaven, eller fulgte oppgavene mer spesifikt. I tillegg fant jeg eksempler fra transkripsjonene på at elevene også i arbeid med andre oppgaver brukte programmeringen til en sjekk av løsningene de hadde gjort på oppgavene.

I utdraget under finner vi et eksempel hvor Marit og Eva jobber med oppgave 4, og læreren oppfordrer elevene til å bruke programmeringen til å sjekke om det de hadde skrevet var riktig. Dette er et eksempel på oppgaver hvor det ikke var gitt i oppgaven at micro:bit skulle brukes

Utdrag 3

Marit: Skal jeg skrive her hva det tallet blir?

Lærer: Ja

Eva: *Til Marit* Har du gjort den?

Lærer: *Til Eva* Se her skal du få den til å, så, først må du se hvor mange er det imellom denne tallfølgen. Fra 88 til 96, hvor langt er det mellom. Da må du tenke litt først. Da kan vi gjøre det enkelt med *Teller med fingrene* 88, 89, 90, 91, 92, 93, 94, 95, 96.

Eva: *Viser arket til Læreren* Er det riktig?

Lærer: Tror det, vet ikke. Hva viste micro:biten?

Marit: Skal jeg sjekke micro:biten?

Lærer: Du kan bruke den.

Marit: *Til seg selv* 88, 96, 104, 112, 120, 128, 136 *Høyere, til Læreren* Jeg har klart det.

I utdrag 3 kan man se at Marit til slutt bruker programmet til å sjekke svaret sitt, etter oppfordring fra lærer.

I det utdrag 4 under jobber Marit med oppgave 2, hvor hun skal bruke micro:bit til å sjekke om det hun trodde kom til å skje på oppgave 1 var det som faktisk skjedde. Dette er et eksempel på elevenes løsning av oppgavene som ba om at de skulle sjekke det de hadde skrevet ned på oppgaven før var riktig.

Utdrag 4

Lærer: [...] Så må du se om, se på oppgave 1, var det akkurat det du tenkte skulle skje?

Marit: Ja

Lærer: Også må du bare skrive her om det var

Marit: Nei det var ikke det.

Lærer: Ikke helt? Da må du skrive det. Hva var det annet som skjedde?

Marit: Ehh.*Mumler litt* Er det sånn at hvis jeg når jeg trykker. 40. Ååh det var som jeg hadde tenkt. [...]

I utdrag 4 ser man at Marit er usikker på om det hun har gjort er riktig. Først sier hun at det var det, så ikke, så sjekker hun ved hjelp av MakeCode og finner ut at hun hadde riktig svar hele tiden. På denne måten brukte hun programmeringen til å sjekke om det hun hadde gjort var riktig.

4.1.4 Utfordringer med å bruke programmet i arbeid med store tall

Denne kategorien inneholder situasjoner hvor elevene jobber med høyere tall, og ikke klarer å bruke programmet på en god måte.

For eksempel kan man se Jan og Bjørns utfordringer i utdrag 5 under. Her diskuterer Jan og Bjørn hvordan de skal løse oppgave 9, hvor de skulle finne ut hvor mange ganger de må trykke på knapp A for å få micro:biten til å vise tallrekken 808, 816, 824, 832.

Utdrag 5

Jan: Jeg tror jeg får det til. Jeg satte den til gjenta for alltid, så teller jeg.

Bjørn: Jan, jeg blir med. Jeg joiner deg, Jan. Må jeg gjøre det?

[...]

Jan: Jeg vet ikke hvor mange jeg har.

Bjørn: Jeg stoppet.

Lærer: Nei, det er problemet, man klarer ikke å telle så langt. Derfor må dere finne en annen strategi.

I utdraget ser man at Jan og Bjørn prøver å trykke, og telle, til de kommer fram til 808. De oppdager fort at dette ikke er en god strategi, da de ikke visste hvor mange ganger de hadde trykt når de kom et stykke på veien. Dette viser at elevene ikke klarte å bruke koden i løsningen av oppgaven med høyere tall.

Gjennom arbeidet med oppgave 9 og 11, som er de eneste to oppgavene som inneholder tall over 800 og 700, hadde elevene utfordringer med å komme frem til et svar. Dette kan man se på utdraget fra observasjonsnotatene mine etter den andre økten.

Elevene slet med å løse oppgave 9 og 11. Elevene prøvde ulike strategier, som å dele tallet i mindre deler, og å jobbe seg oppover fra tall de visste hvor mange ganger de måtte trykke for å få. Allikevel var det ingen av elevene som klarte å komme frem til et svar på egenhånd. Til slutt måtte jeg forklare at 800 er 100 ganger 8. Da klarte alle elevene å komme frem til et svar med mer og mindre videre støtte.

Som man ser fra utdraget fra observasjonsnotatet måtte jeg forklare elevene at 800 er 100 ganger 8. Under har jeg tatt med et utdrag fra transkripsjonen som viser hva jeg sa til elevene i denne situasjonen.

Lærer: Når du har 800, dersom du ganger noe med 100. Dersom du ganger 8 med 100 får du 800. Det betyr at dersom du trykker, dersom du har 100 8-ere, så får du 800. Så når du har trykt 100 ganger får du 800.

Det at elevene ikke klarte å løse oppgave 9 og oppgave 11 uten forklaring fra lærer, viser at elevene slet med å bruke programmeringen i arbeidet med store tall, og at programmeringen kanskje ikke ga elevene nok støtte når de skulle løse disse oppgavene.

4.1.5 Utfordringer med forståelsen av programmeringen som hinder for løsningen av oppgavene

Denne kategorien inneholder situasjoner hvor elevenes mangel på forståelse for programmet og programmeringen i seg selv ser ut til å være et hinder for arbeid med og løsningen av oppgavene.

I utdrag 6 under kan man for eksempel se at Bjørn først ikke skjønner hva koden betyr, noe som hindrer hans løsning av oppgaven.

Utdrag 6

Bjørn: Det blir 0 visst

Lærer: Hmm? *Reiser seg og går bort til Bjørn*

Bjørn: Jeg får det ikke til.

Lærer: Hva du, hva som blir null?

Bjørn: Når du trykker 6 ganger.

[...]

Lærer: Husker du hva den endre variabelen med 8 betydde? *Peker på arket mens hun forklarer*

Bjørn: Nei

Lærer: Nei? *Noen sekunders pause* Du endrer den ikke til 8, du endrer den med 8. Som betyr det samme som å plusse på 8.

Bjørn: Så det blir bare pluss hele tiden?

Lærer nikker

Utdrag 6 viser at Bjørn ikke forsto hva koden og programmet betyr, og dermed ikke klarer å løse oppgaven. Programmeringen ble her et hinder for å kunne arbeide med

multiplikasjonen som ligger i oppgavene, da han ikke visste hvordan han skulle gå videre i arbeidet.

Et annet eksempel er utdrag 7 under, hvor Bjørn jobber med oppgave 2. Denne oppgaven innebar at elevene først måtte kopiere koden fra arket, og inn på Chromebooks for å kunne bruke den videre.

Utdrag 7

Bjørn: *Tar opp oppgavearket* Ved start

Lærer: Et tips til når dere skal lage den samme koden er å se på fargene. Ut fra hvor dere finner de ulike blokkene.

Bjørn: Hmm, den er jo ikke her. *Ser nøye på arket i noen sekunder* kan du komme hit litt?

Lærer: Ja, klart det*Går bort til Bjørn*

Bjørn: Hvorfor står det ved start vis tall? Jeg finner den ikke.

Lærer: Det er to ulike blokker.

Bjørn: åhh

Lærer: Da finner du dem.

Bjørn: Dem er jo helt ser ut som de sitter sammen.

Lærer: Det er fordi de har samme farge, derfor ser det veldig sånn ut.

Bjørn: Vis tall 0, nei.

Lærer: Og når dere skal ha inn en variabel må dere lage en variabel. Som vi gjorde tidligere.

Bjørn: Dette er vanskelig.

Lærer: Det er litt vanskelig.

Bjørn: Vis tall. Hva med dem? *Peker på arket* Hvor skal den sitte inn?

Lærer: Det er en variabel.

Bjørn: Åja

I denne samtalen mellom lærer og Bjørn ser man at Bjørn trenger hjelp til å finne blokkene han trenger. Denne oppgaven måtte elevene få til for å kunne gjøre de andre oppgavene, og på denne måten var Bjørns manglende erfaring med programmering et hinder for arbeidet med oppgavene.

I det siste utdraget jeg har valgt å ta med i denne kategorien er Marits program som ikke fungerer i utdrag 8 under.

Utdrag 8

Marit: *Henvendt til lærer* Det her funker ikke. Skal det skje noe når jeg trykker på knapp A?

Student: Hmm *Ser på programmet på Chromebook* Skal vi se her. Jeg tror vi prøver å trykke stopp, for jeg klarer ikke å se at du har gjort noe feil. Den viser 0 fra start. Hmm. Jo, men du har. Se her. *Peker på skjermen* Endre endre variabelen. Endre den første variabelen. Du har laget en ny variabel, men du må ha den første variabelen. Det var veldig nært. Nå kan du se om, ja. Så må du se om, se på oppgave 1, var det akkurat det du tenkte skulle skje?

Marit trengte hjelp med å finne ut hvorfor programmet ikke fungerte. Feilen hennes kom fra at hun hadde laget to ulike variabler, hvor hun hadde kalt den ene «endre variabelen». Hun satte da den ene variabelen til 0 og endret den andre variabelen med 8, uten at denne var satt til noe verdi til å starte med. Dette viser en manglende forståelse

for programmet og variabler, noe som gjorde at løsningen av oppgavene stoppet opp en stund. På denne måten ble denne utfordringen et hinder for arbeidet med oppgavene, og dermed også utforskningen av multiplikasjonen som ligger i oppgavene.

4.2 Arbeid med multiplikasjon

Forskningsfokus for dette masterprosjektet er å finne ut hvordan programmering som et instrument kan bidra til elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur. Derfor har jeg i analysen også sett på elevenes arbeid med multiplikasjon. Multiplikasjon ble bare eksplisitt nevnt en gang i løpet av begge øktene, når en elev sa «Det er på en måte åttegangen». Allikevel så jobbet elevene med tallfølger med multiplikativ struktur, selv om de kanskje ikke skjønnte det selv. Innenfor kategorien *Arbeid med multiplikasjon* finnes underkategoriene: (1) *Bruk av gjentatt addisjon*, (2) *Brukte oppgavearket eller programmet til å løse oppgavene på annen måte enn multiplikasjon* og (3) *Elevene knytter ikke arbeidet med oppgavene til multiplikasjon*.

4.2.1 Bruk av gjentatt addisjon

I analysen kommer det frem at elevene i stor grad benyttet seg av gjentatt addisjon for å løse oppgavene. Under har jeg valgt ut noen eksempler fra analysen som viser elevenes bruk av gjentatt addisjon som strategi for å løse oppgavene.

For eksempel ser vi i utsagnet under at Eva jobber med oppgave 1 i den første økten.

Eva: Når a trykkes så endres variabelen med 8, så den starter på 0 også når man trykker på A blir den 8.

Her ser vi at Eva ikke snakker om endringen som skjer når man trykker på knapp A som multiplikasjon. Hun tar heller utgangspunkt i at den starter på null og øker med 8, og blir 8. Dette tyder på at eleven tenker på dette regnestykket, som kan skrives som $8 \cdot 1$, som $0+8$ i stedet, altså som addisjon. Dette gir samme løsningen, men endrer måten man løser regnestykker med høyere tall.

I det neste utsagnet jobber Marit med oppgave 1.

Marit: 8, 8 pluss 8 er 16. pluss 8 er tretti eh. 16 pluss 16. 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32. pluss 8 er 40 pluss 8 er 48.

I utsagnet ser man at Marit løser oppgave 1 ved å plusse på åtte 6 ganger, altså ved hjelp av gjentatt addisjon.

I det neste utdraget jobber Bjørn med oppgave 4. Oppgaven er blant annet at han skal beskrive tallfølgen «88, 96, 104, 112, 120». Bjørn ville begynne på neste oppgave uten å ha svart på denne delen av oppgaven, og læreren prøver i derfor å på han til å beskrive tallfølgen.

Lærer: [...] Kan du forklare den, beskrive tallfølgen? Kan du forklare til noen slik som jeg ba han [en annen elev] om? Hvordan kan du

Bjørn: $8+8+8+8$

I en beskrivelse av tallfølgen sier Bjørn at den er $8+8+8+8$. Dette viser at han bare ser på disse tallene som gjentatt addisjon, uten å ta hensyn til hvor tallfølgen begynner.

I utdrag 9 snakker Bjørn med lærer om oppgave 5. Her skal han beskrive tallfølgen «...,21, 28, 35, 42,..., 77, 84, 91,...».

Utdrag 9

Bjørn: Okei, greit, så åtte mellom hvert tall.

Lærer: mhm?

Bjørn: Også skal jeg skrive noe under?

Lærer: Men er det, hmm, jeg tror du må se nøyere på tallene igjen

Bjørn: nei, 8 pluss. *Liten pause for å se om læreren bekrefter at det er riktig* Ja sju

Lærer: Ja, det var kanskje ikke åtte i den.

Bjørn: *litt usikkert* sju

Lærer: *Peker på arket* Du må se, tell imellom her.

Bjørn: Tallet mellom der, da er det sju da.

Lærer: Ja, for det er på den oppgaven vi lurte

[...]

Bjørn: Sånn ferdig. Sju

Her ser vi at Bjørn bare ser på hvor mye det er mellom hvert tall, uten å se på hvor tallrekken starter. På svararket skrev han «7 mellom hvert tall», noe som liknet på det de andre elevene skrev på sine svarark. Dette viser at elevene, gjennom et fokus på multiplikasjon som gjentatt addisjon, fikk et fokus på hva som var mellom tallene, uten at de klarte å beskrive tallfølgen på noen annen måte.

Disse utdragene er et utvalg av elevenes utsagn som viser deres bruk av strategien gjentatt addisjon. Gjentatt addisjon var den eneste multiplikasjonsstrategien som elevene benyttet seg av, og den ble benyttet ved løsningen av flere oppgaver gjennom begge øktene.

Det at elevene bruker gjentatt addisjon kan være på grunn av måten oppgavene er lagt opp. Koden som ble gitt i oppgavene innebar at elevene måtte trykke gjentatte ganger på knappen på micro:bit for å vise de ulike tallene. Dette ligner på gjentatt addisjon, siden det legges til ett og ett tall hver gang knappen trykkes. Det er derfor ikke så rart at elevene bruker gjentatt addisjon som strategi for å løse oppgavene.

4.2.2 Elevene bruker oppgavearket eller programmeringen til å løse oppgavene på en annen måte enn multiplikasjon

Analysen viser at elevene ved flere anledninger brukte programmet eller oppgavene som ble utdelt på andre måter enn jeg forutså, og på måter som gjorde at de enklere kom frem til svaret uten å måtte gjøre oppgaven slik jeg hadde tenkt. I disse tilfellene brukte elevene programmet eller oppgavearket til å lese av svaret, eller trykke seg frem, for å løse oppgavene.

I dette utdraget jobber Bjørn med oppgave 3, i første datainnsamlingsøkt.

Utdrag 10

Bjørn: Nå står det jo her. Få mikro:biten til å vise tallfølgen 80 [leser oppgaven], det har jeg allerede gjort.

Lærer: Har du gjort det?

Bjørn: Ja, jeg bare trykket veldig mye.

Lærer: Mm, ja fordi du var så langt opp.

Bjørn: 280

Lærer: Da vil jeg også at du skal beskrive den. Hva. Litt sånn hva er endringen i den.

Bjørn: jeg bare trykket veldig mye egentlig da

Lærer: Men se her. Kan du beskrive tallfølgen? Og hva vil være de tre neste?

Viser til oppgave-arket Det har du jo. *peker på Chromebook* Men

Bjørn: Da vil den neste være 125. De tre eller de to?

Lærer: De 3 *peker på oppgavearket*

Bjørn: De vil være hundre-og-tjue hundre-og-trettiseks og hundre-og-tretti og hundre-og-førti ... Hva var det de var igjen? Ja, 136, fader. * skriver tallene ned på arket*

I utdrag 10 ser man at Bjørn snakker med læreren, og forklarer at han allerede har gjort oppgaven når han kommer til den. Det er fordi han har trykt på knapp A mange ganger for å få høyest mulig tall, og siden oppgaven var «Få mikro:biten til å vise tallfølgen [...]», var det å trykke forbi disse tallene nok for Bjørn til å si at oppgaven var gjort.

I utdrag 11 jobber Marit med oppgave 7, i andre datainnsamlingsøkt. Hun var kommet til siste deloppgave hvor elevene skulle lage koden, og sjekke om løsningen de hadde på deloppgavene som kom før i oppgave 7, var riktige. Her ser vi at Marit allerede hadde testet ut om svaret var riktig når hun kom til oppgaven hvor dette skulle gjøres. Det er fordi hun brukte programmet som en støtte i løsningen av de første deloppgavene, og som en hjelp til å løse dem. I en kort samtale med læreren sier Marit at hun allerede hadde kjørt programmet.

Utdrag 11

Marit: Hæ, men jeg kjørte jo den her nettopp?

Lærer: Men da skriver du bare at det var det samme.

I utdrag 12 under jobber Bjørn og Marit med oppgave 7, som er en del av andre runde datainnsamling. I utdraget har Marit oppdaget at hun kunne se på lappen de fikk utdelt som oppgave 6 og bruke den til å telle seg fram til løsningen av første delen av oppgave 7.

Utdrag 12

Bjørn: *Leser oppgaveteksten* Uten å trykke på micro:biten, hvor mange ganger tror du at du må trykke på a for å få 40? 8, eh, 8 så får man, Først så får man 8 så får man 16 så får man *Bruker fingrene til å holde styr på hvor mange ganger han har lagt til 8*

Marit: 5 ganger! Du kan bare se på lappen.

Bjørn: Ååh. *Ser på lappen fra oppgave 1* Først får man 8 så 16, så 24, så 32, så 40. Ferdig.

Lærer: Ja, da må du bare skrive ned hvordan du kom frem til svaret. Du telte på lappen, brukte lappen som hjelp.

Lærer: Kan du skrive hvordan du kom frem til det?

Bjørn: Hvordan jeg kom fram til det. Jeg bare

Marit: *Leser høyt det hun skriver* Jeg så på lappen. Vi fikk hehe

Bjørn: Så på lappen. Så på billappen. Hehe. Ferdig. [...]

Utdrag 12 er en samtale mellom Bjørn og Marit, hvor Marit forteller Bjørn at han kan se på lappen for å finne ut hva svaret på oppgaven var, og at han dermed ikke trengte å tenke eller regne seg frem til løsningen.

4.2.3 Elevene knytter ikke arbeidet med oppgavene til multiplikasjon

Gjennom begge øktene var det ingen av elevene som nevnte at de knyttet arbeidet med programmering av tallfølger med multiplikativ struktur, til annet arbeid i multiplikasjon.

I oppgave 1 og oppgave 9 skulle elevene lage et regnestykke som passet med situasjonen de jobbet med. Det var ingen av elevene som klarte å lage et regnestykke, og elevene virket ikke som de forsto hva det å lage et regnestykke til oppgavene innebar.

I et utdrag under spør lærer om Jan kan lage et regnestykke, men han forstår ikke hva det innebærer.

Lærer: [...]Da lurer jeg bare på om du kan skrive det som et regnestykke? Kan du tenke det til hva dette er slags regnestykke? Og dersom du ikke klarer det kan du gå videre. Jeg bare lurer på om du har en ide.

Jan: jeg skjønner ikke.

Det at Jan ikke skjønner hva det vil si å lage et regnestykke kan være fordi han, og de andre elevene, ikke knyttet disse oppgavene til noen regneoperasjon, altså multiplikasjon, slik som jeg prøvde å legge opp til.

Ved bare en anledning ble gangning nevnt eksplisitt i løpet av begge timene. Denne situasjonen kan ses i utsagnet under, hvor Marit jobber med oppgave 2.

Marit: Ehh. *Mumler litt* Er det sånn at hvis jeg når jeg trykker. 40. Ååh det var som jeg hadde tenkt. Det er på en måte åttegangen.

Dette utsagnet er det eneste hvor gangning er eksplisitt nevnt. Det at Marit i tillegg sier at det *på en måte* er åttegangen kan tyde på at hun ser på det hun gjør i arbeidet med programmeringsoppgavene som noe som bare ligner på arbeid med åttegangen, men at det ikke faktisk er det.

Begrep som multiplikasjon eller gangning er ikke nevnt i løpet av noen av innsamlingene. Allikevel bruker elevene begrepet ganger mange steder i løpet av innsamlingene. Begrepet ganger er knyttet til hvor mange ganger man må trykke på knapp A for å få de ulike tallene som oppgavene ber om. Dette er arbeid som kan knyttes til multiplikasjon, men det blir ikke eksplisitt nevnt, og elevene ser ikke ut til å knytte dette arbeidet med oppgavene til arbeid med multiplikasjon som de har gjort tidligere. Det synes blant annet ved at ingen av elevene klarte å lage regnestykker til arbeidet de gjorde knyttet til oppgave 1 og 9.

5 Drøfting

Fokuset for denne masterstudien har vært å undersøke hvordan programmering fungerer som et instrument i arbeidet med tallfølger med multiplikativ struktur. Jeg utformet et undervisningsopplegg som skulle undersøke dette, som ble gjennomført sammen med fire elever på 4. trinn. I denne studien har jeg fokusert på hvordan disse fire elevene brukte programmeringen gjennom det blokkprogrammeringsverktøyet micro:bit med det blokkbaserte programmeringsspråket MakeCode i arbeidet med oppgavene som ble gitt og hvordan de arbeidet med multiplikasjon i løsningen av oppgavene.

Undervisningsopplegget inneholdt oppgaver som alle tok utgangspunkt i tallfølger med multiplikativ struktur med utgangspunkt i koder som var laget i editoren for programmering av micro:bit, micro:bit.org. Gjennom arbeidet med alle oppgavene var gjentatt addisjon en mye brukt strategi. Dette er ifølge Rønning og Burheim (2020) en strategi innenfor den primitive modellen for multiplikasjon som kalles *like grupper*. Rønning og Burheim (2020) kommenterer at det er en fordel for elevene at de kjenner til flere modeller for multiplikasjon, slik at de lettere kan kjenne igjen situasjoner der multiplikasjon kan brukes. Før gjennomføringen av undervisningsopplegget hadde jeg en hypotese om at elevene ville bruke andre strategier enn gjentatt addisjon i arbeidet med oppgavene, og oppgavene var forsøkt laget på en måte som la opp til at elevene også skulle bruke andre strategier for multiplikasjon.

Videre i dette drøftingskapittelet vil jeg diskutere funnene som er presentert i resultatkapitlet opp mot Rabardels teori om instrumentell skapelse. Instrumentell skapelse er en prosess hvor en artefakt blir utviklet til et meningsfullt verktøy, også kalt instrument, for den som skal bruke artefakten (Rabardel & Samurcay, 2001). Denne prosessen består av to delprosesser som kalles instrumentalisering og instrumentering. Instrumentalisering innebærer at subjektet påvirker artefakten, og instrumentering er hvordan subjektet blir påvirket av artefakten. Jeg vil i hovedsak se på instrumentering, altså hvordan elevene ble formet av programmering som artefakt i arbeidet med oppgavene.

5.1 Er programmering som artefakt til hjelp eller hindring i arbeidet med tallfølger med multiplikativ struktur?

I presentasjonen av resultatene fra analysen kan man se at programmeringen i enkelte tilfeller var et instrument i elevenes arbeid med oppgavene, mens programmeringen i andre tilfeller ser ut til å være mer til hinder i elevenes arbeid med tallfølger med multiplikativ struktur.

Et artefakt blir et instrument når den blir brukt på en hensiktsmessig måte i arbeid med oppgaver (Rabardel & Samurcay, 2001). I delkapittel 4.1.1 *Bruker programmeringen i utforskningen av store tall* ser man at elevene bruker programmeringen som et instrument på en annen måte enn jeg hadde forutsett. Elevene bruker programmeringen som et instrument i utforskningen av store tall heller enn som et instrument for å utvikle kunnskap om tallfølger med multiplikativ struktur slik som jeg ønsket. Elevene er i en instrumenteringsprosess hvor de formes av artefakten, og utvikler sine skjema om store tall ved å bruke de funksjonene som finnes i programmeringen som artefakt. For eksempel ser man i utdrag 2 i resultatkapittelet at Bjørn utforsker store tall når han sier «1000! Jeg kom meg til 1000!». Gjennom denne utforskningen kan det tenkes at Bjørn utvikler sine skjema om store tall, som han kanskje ikke har utforsket så mye før, og at

han utvikler sine skjema om hvordan programmeringen kan brukes i utforskningen av store tall.

Elevene brukte også programmeringen som verktøy ved å sjekke om det de tenkte var riktig. Det ser vi eksempler på i delkapittel 4.1.3. I for eksempel *Utdrag 3* ser man at Marit i samtale med lærer bruker programmeringen til å sjekke om det hun har gjort var riktig. I utdraget er Marit og Eva usikker på om det de har gjort er riktig, og til slutt bruker Marit programmeringen til å sjekke om det hun hadde tenkt var riktig. Dette er en måte å bruke programmeringen som verktøy i løsningen av oppgavene som jeg ikke hadde forutsett før gjennomføringen av undervisningsøktene. Marit bruker programmeringen som et instrument i arbeidet med oppgaven ved at hun på en hensiktsmessig måte bruker programmeringen som artefakt til det hun trengte i arbeidet med oppgavene, nemlig å sjekke om det hun tenkte var riktig. På denne måten blir Marit formet av artefakten, programmeringen, og hun utvikler sine skjema om bruken av artefakten til å innebære at man kan sjekke om det man trodde var riktig svar, var det.

I eksemplene i resultatkapittelet om at elevene lagde egne koder som ikke var hensiktsmessig i løsningen av oppgavene finner man eksempler hvor programmeringen som artefakt virket mer som et hinder i løsningen av oppgavene enn som et instrument for elevene (se delkapittel 4.1.2). For eksempel laget Bjørn en kode som ikke lignet på den jeg hadde laget, og som jeg hadde tenkt at de skulle bruke videre. Bjørn møter på problemer når han skal bruke sin egenlagde kode i de videre oppgavene. Han har også utfordringer med å se hvordan han kan endre koden for å få en kode som er mer hensiktsmessig. På denne måten var programmeringen, eller kanskje heller Bjørns manglende forståelse for programmeringen, et hinder i arbeidet med tallfølger med multiplikativ struktur, som kunne ha utviklet hans forståelse for multiplikasjon. I Bjørns instrumenteringsprosess har han ikke utviklet de skjemaene han trenger for å kunne utnytte programmeringen som et instrument i arbeidet med oppgavene.

Gjennom analysen fant jeg eksempler hvor elevenes mangel på forståelse av programmering var et hinder for arbeidet med oppgavene. For eksempel kan man i *Utdrag 6* i resultatkapitlet (se delkapittel 4.1.5) se at Bjørn ikke forstår hva blokken «endre variabelen med 8» betydde. Etter at læreren har forklart hva denne blokken betyr klarer Bjørn å løse oppgaven. Bjørn klarte ikke å løse oppgaven før han hadde fått denne forståelsen for hva blokken betydde. I instrumenteringsprosessen hans manglet han utviklede skjema om hva de ulike blokkene betydde, og hvordan man kunne bruke dem. Han klarte til slutt i utdraget å forstå hva denne blokken betydde etter forklaring fra lærer. Dermed fikk han utvidet sitt skjema om blokkene i programmeringen, men han måtte ha denne forståelsen på plass før han kunne jobbe med oppgavene om tallfølger med multiplikativ struktur. Dermed var mangelen på forståelse for programmeringen et hinder for at programmeringen kunne være et verktøy for elevene i utviklingen av kunnskap for tallfølger med multiplikativ struktur.

I delkapittel 4.2.2 *Elevene bruker oppgavearket eller programmeringen til å løse oppgavene på en annen måte enn multiplikasjon* ser man eksempler på at oppgaveformuleringene og programmeringen ble brukt på andre måter enn jeg hadde forutsett, til å løse oppgavene på andre måter enn multiplikasjon. For eksempel jobbet Marit med en oppgave i *Utdrag 11*, hvor hun allerede har trykt seg fram til de tallene hun skulle vise, og derfor ikke trengte å tenke på hvor mange ganger hun skulle trykke for å vise tallfølgen på micro:biten. Dette viser en svakhet med oppgaveformuleringen, som jeg kommer tilbake til i delkapittel 5.3 *Metodediskusjon*. Slik som oppgavene var formulert kunne elevene løse en del av oppgavene uten å bruke multiplikasjon. Det gjør

at programmering, slik som den er brukt i dette masterprosjektet, ikke fungerte så godt som instrument for elevene i løsning av oppgaver med tallfølger med multiplikativ struktur. Det at elevene bruker programmeringen eller oppgavearket til å løse oppgavene kan også tyde på at de ser på programmeringen som et hinder i oppgaveløsningen, som de unngår å bruke dersom de kan.

Gjennom hele analysen kom det frem at elevene ikke så på det de gjorde i undervisningsøktene som multiplikasjon. Dette gjør at det som ble gjort i undervisningsøktene ikke utvidet elevenes kunnskap om multiplikasjon, slik som jeg hadde ønsket, da elevene ikke så på det de gjorde som multiplikasjon. Det at elevene ikke så at det vi jobbet med var multiplikasjon gjorde også at elevene ikke kunne bruke sine allerede utviklede skjemaer om multiplikasjon i arbeidet med oppgavene.

I undervisningsøktene brukte vi en kode som elevene ikke klarte å bruke i arbeidet med store tall. Eksempler på elevenes vanskeligheter kan man se i delkapittel 4.1.4 *Utfordringer med å bruke programmet i arbeid med store tall*. Utdrag 5 viser et eksempel hvor Bjørn og Jan prøver å bruke koden til å løse en oppgave med tall over 800, og ikke får til å løse oppgaven. Bjørn og Jan prøver i dette eksempelet å bruke programmeringen som instrument i løsningen av denne oppgaven, men finner ut at det ligger noen begrensninger i artefakten som gjør det vanskelig. Artefakten påvirker da Bjørn og Jan til å utvikle deres skjemaer for å løse oppgaven på en annen måte enn de først prøver. De kommer ikke fram til et svar på oppgaven uten hjelp fra lærer, og de bruker ikke programmeringen til å løse oppgaven. Programmeringen fungerte altså ikke som instrument for elevene i arbeidet med denne oppgaven.

Gjennom analysen kan man se eksempler hvor programmeringen som et instrument bidrar til elevenes løsning av oppgavene. Det er imidlertid ingen eksempler hvor man kan si sikkert at programmeringen fungerer som instrument i elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur. Det er flere eksempler hvor det heller ser ut som at programmeringen er et hinder for elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur.

5.2 Elevenes valg av løsningsstrategi

Elevenes instrumenteringsprosess er synlig i analysen gjennom at elevenes valg av løsningsstrategi er tilpasset artefakten. Elevene har mentale skjema for hvordan artefakten kan brukes og for hvordan spesifikke typer oppgave skal løses. I instrumenteringsprosessen blir disse skjemaene brukt, samtidig som de blir utvidet gjennom ny erfaring. Elevene velger løsningsstrategi i arbeidet med oppgavene ut fra de skjemaene de har om hvilke muligheter de vet finnes i programmeringen, og skjemaene de har om hvordan man løser oppgavetypen de får utdelt. I arbeidet med oppgavene ble det tydelig at elevene benytter seg av begrensede strategier for løsning av regnstykker i multiplikasjon. Elevene bruker, som man ser i kapittel 4.2.1 *Bruk av gjentatt addisjon*, gjentatt addisjon som strategi i arbeidet med oppgavene. Som nevnt tidligere er gjentatt addisjon en del av like grupper modellen for multiplikasjon, noe som blir sett som en primitiv modell for multiplikasjon (Rønning & Burheim, 2020). Anghileri (2006, s. 90) skriver at gjentatt addisjon kan føre til begrensninger i elevenes forståelse for multiplikasjon. For eksempel er det ikke er mulig å benytte seg av strategien gjentatt addisjon dersom man skal multiplisere to tall som er mellom null og en, slik som $0,3 \cdot 0,4$. Da trenger elevene en annen forståelse for multiplikasjon, som i dette tilfellet bygger på ideen om forholdet mellom tall. Rønning og Burheim (2020) skriver at det er en fordel for elevene å kunne ulike strategier for løsning av oppgaver i multiplikasjon (Rønning &

Burheim, 2020). Programmeringen, slik som den er brukt i dette masterprosjektet, gir dermed begrensede muligheter for å utvikle elevenes kunnskaper i multiplikasjon.

Et av formålene med oppgavene jeg ga elevene var som tidligere nevnt at de skulle utvikle grunnleggende tallfakta innenfor multiplikasjon. Grunnleggende tallfakta er ifølge Baroody (2006, s.22) i fase 3 å kunne gi svar på regnestykker uten å måtte tenke seg om. Jeg ville at elevene skulle lære grunnleggende tallfakta uten å måtte sitte å pugge gangetabellen. Pugging av gangetabellen er ifølge Anghileri (2006, s. 94) uoverkommelig for noen elever, og jeg ville da gi elevene en annen måte å utvikle de grunnleggende tallfaktaene innenfor multiplikasjon. Ved å bruke gjentatt addisjon regner elevene ut addisjonsstykker, som gjør at de ikke jobber med grunnleggende tallfakta i multiplikasjon. Elevene jobbet altså ikke med multiplikasjon på den måten jeg hadde ønsket.

Elevenes instrumenteringsprosess er begrenset av de skjemaene de har for bruk av programmering som verktøy og for løsning av oppgaver med tallfølger med multiplikativ struktur. Elevene klarer ikke ut fra de skjemaene de har om programmering og tallfølger med multiplikativ struktur og finne andre løsningsstrategier enn gjentatt addisjon. På denne måten virket ikke programmeringen som et medierende artefakt, eller et instrument, i arbeidet med oppgavene.

Oppgavestrukturen er viktig for hvordan elevene løser oppgavene. Noe som kan være grunnen til at elevene valgte å benytte seg av gjentatt addisjon. Alle oppgavene som elevene ble gitt i undervisningsøktene tok utgangspunkt i en kode hvor det legges til sju eller åtte hver gang man trykker på knapp A. Det ligger altså i koden at det legges til sju eller åtte et gitt antall ganger, noe som kan ha bidratt til at elevene benyttet seg av strategien gjentatt addisjon. Dette blir litt nærmere diskutert i kapittel 5.3

Metodediskusjon.

5.3 Metodediskusjon

Dersom jeg skulle ha gjennomført datainnsamlingen til dette masterprosjektet på nytt, er det flere ting jeg ville ha vurdert å gjøre annerledes. Det å bruke micro:bit som artefakt skapte noen utfordringer. Micro:bit har en liten skjerm slik at det er begrenset hva den kan vise på en gang. Det gjorde at hvilke typer oppgaver som kunne gis også begrenses. For eksempel kunne oppgaver inneholdt større tall, men det ville vært vanskelig å vise på micro:biten. Elevenes mangel på kunnskaper i micro:bit og MakeCode som programmeringsspråk skapte også noen begrensninger i hvor komplisert koden kunne være.

Oppgavestrukturen og oppgaveformuleringen på enkelte oppgaver var uklare for elevene. For eksempel var det å «beskrive tallfølgen» en formulering som elevene ikke skjønnte. Dersom jeg skulle gjøre det på nytt ville jeg ha forklart hva som ligger i å beskrive en tallfølge, på oppgavearket. Det at elevene brukte gjentatt addisjon kan være på grunn av måten koden var laget, hvor det alltid startet på 0 og ble lagt til sju eller åtte hver gang man trykket på knapp A. Dette får kanskje elevene inn på tenkemåten for gjentatt addisjon, noe som de da fortsetter med gjennom undervisningsøktene.

I oppgavene jeg laget er det flere steder hvor jeg spør elevene om hvilke tall som kommer videre i tallfølgene. Det at jeg leter etter bare ett riktig svar er misvisende, da man ifølge Stylianides (2008, s.11) ikke kan være matematisk sikker på en fortsettelse på en tallfølge som ikke har figur. For å unngå dette problemet kunne jeg ifølge

Stylianides (2008 s. 11) ha formulert oppgaven som å spørre etter den enkleste eller mest naturlige fortsettelsen på tallfølgen.

Det ble tydelig gjennom undervisningsøktene og analysen av disse, at elevene ikke så på arbeidet de gjorde som multiplikasjon. Det hadde derfor kanskje vært bedre om jeg startet undervisningene med å fortelle elevene at oppgavene handlet om multiplikasjon, slik at elevene kunne ha brukt deres skjema om multiplikasjon, i arbeidet med oppgavene. Dette hadde kanskje endret metodene elevene brukte for løsning av oppgavene, noe som videre kunne ha utviklet deres kunnskap om tallfølger med multiplikativ struktur.

I gjennomføringen av undervisningsøktene fikk elevene hvert sitt oppgaveark og de jobbet i hovedsak individuelt. Dette gjorde det vanskelig å finne ut hvordan elevene tenkte, da de ikke sa så mye underveis i arbeidet som de sannsynligvis ville ha gjort dersom de hadde jobbet sammen i par. I tillegg er læring ifølge sosiokulturell læringsteori en sosial prosess hvor individer tilegner seg konsepter, redskaper gjennom samhandling og kommunikasjon (Hundeland et al., 2014, s.208). Dermed ville det vært naturlig om elevene fikk utnytte den læringen som ligger den ekstra kommunikasjonen de hadde måttet delta i dersom de hadde jobbet i par.

Undervisningsopplegget ble gjennomført i to økter, og med fire elever. Dette er et veldig lite forskningsprosjekt, med få deltakere. Derfor kan man ikke si noe om hvordan dette undervisningsopplegget hadde fungert med andre elever, eller en hel klasse.

6 Konklusjon

I denne masteroppgaven har jeg undersøkt hvordan programmering som instrument kan brukes i utvikling av elevers kunnskap om tallfølger med multiplikativ struktur. I studien ser det ut som at programmeringen i liten grad bidro som et instrument i elevenes utvikling av kunnskaper om tallfølger med multiplikativ struktur. Elevene løste oppgavene som ble gitt med å bruke blokkprogrammeringsspråket MakeCode, men som man ser i diskusjonsdelen klarte ikke elevene å bruke programmeringen som instrument i arbeidet med tallfølger med multiplikativ struktur, og programmeringen var på mange måter et hinder for elevenes arbeid med multiplikasjon, heller enn et instrument for elevene i arbeidet med oppgavene.

I analysen fant jeg at elevene i hovedsak brukte gjentatt addisjon som strategi for å løse oppgavene med tallfølger med multiplikativ struktur. Gjentatt addisjon er en metode som kan føre til begrensninger for elevene i arbeid med multiplikasjon med to tall mellom 0 og 1, for eksempel $0,4 * 0,1$ (Anghileri, 2006). Elevene trenger andre måter å tenke på for å løse slike oppgaver, noe undervisningsopplegget i dette masterprosjektet ikke hjelper elevene med.

Elevenes bruk av programmeringen skapte en del utfordringer for utviklingen av elevenes kunnskap om tallfølger med multiplikativ struktur. Mangel på forståelse for programmeringen, lagring av koder som ikke var hensiktsmessig for videre oppgaver og utfordringer med å bruke programmeringen i arbeid med store tall er eksempler på dette. Gjennom disse utfordringene kan man se at programmeringen som instrument i elevenes utvikling av kunnskap om tallfølger med multiplikativ struktur, ikke fungerte på en hensiktsmessig måte.

Elevene brukte programmeringen som instrument i læringen av andre ting enn tallfølger med multiplikativ struktur. Elevene brukte programmeringen til å utforske store tall, og til å sjekke om det de hadde tenkt var riktig. I disse eksemplene fungerer programmeringen som et instrument for elevene i arbeidet med oppgavene, men på andre måter enn det som var formålet med dette masterprosjektet.

I sin litteraturstudie om programmering konkluderte Forsström og Kaufman (2018) at det trengs mer forskning på hvordan programmering kan brukes som arbeidsmåte i sammenheng med andre matematiske tema enn geometri. Jeg har forsøkt å se på hvordan programmering kan brukes som instrument for utvikling av kunnskap om tallfølger med multiplikativ struktur. Jeg har funnet at det dukker opp mange utfordringer når man bruker blokkprogrammering i arbeidet med multiplikasjon, men det er mange andre måter det kan prøves ut. Dette masterprosjektet har bare testet ut et undervisningsopplegg på to økter med fire elever, og det trengs derfor andre studier for å kunne konkludere med om programmering kan være et instrument for elevene i arbeid med multiplikasjon. I tillegg trengs det forskning på hvordan andre matematiske temaer kan ses i sammenheng med programmering som arbeidsmåte.

Referanser

- Akker, J. & Gravemeijer, K., Mckenney, S. & Nieveen, N. (2006). Introducing educational design research. I J. Akker, K. Gravemeijer, S. McKenney & N. Nieveen (Red.) *Educational Design Research* (s. 3–7), Taylor and Francis e-Library.
- Anghileri, J. (2006). *Teaching number sense* (2. utg.). Continuum International Publishing Group.
- Baroody, A. J. (2006). Why children have difficulties mastering the basic number combinations and how to help them. *Teaching children mathematics*, 13(1), 22–31. <https://doi.org/10.5951/TCM.13.1.0022>
- Busuttil, L., & Formosa, M. (2020). Teaching Computing without Computers: Unplugged Computing as a Pedagogical Strategy. *Informatics in Education*, 19(4), 569-587. <https://doi.org/10.15388/infedu.2020.25>
- Clark, T., Foster, L., Sloan, L. & Bryman, A. (2021). *Bryman's social research methods* (6. utg.). Oxford University Press.
- Den nasjonale forskningsetiske komite for samfunnsvitenskap og humaniora (2021, 16. desember) *Forskningsetiske retningslinjer for samfunnsvitenskap og humaniora*. Forskningsetikk <https://www.forskningsetikk.no/retningslinjer/hum-sam/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora/>
- Forsström, S. A. & 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>
- Franklin, D., Hill, C., Dwyer, H. A., Hansen, A. K., Iveland, A., & Harlow, D. B. (2016, februar). *Initialization in scratch: Seeking knowledge transfer. Proceedings of the 47th ACM Technical Symposium on computing science education*, 217–222. <https://doi.org/10.1145/2839509.2844569>
- Gravenmeijer, K. P. E. & Cobb, P. (2006). Design research form a learning design perspective. I J. Akker, K. Gravemeijer, S. McKenney & N. Nieveen (Red.) *Educational Design Research* (s. 45–85) Taylor and Francis e-Library.
- Greer, B. (1992). Multiplication and division as models of situations. I D. A. Grouws (Red.), *Handbook of research on mathematics teaching and learning: A project of the National Council of Teachers of Mathematics* (s. 276–295). Macmillan Publishing.
- Haraldsrud, A. D., Sveinsson, H. A. & Løvold, H. H. (2020). *Programmering i skolen*. Universitetsforlaget.
- Hundeland, P. S., Carlsen, M., & Erfjord, I. (2014). Children's engagement with mathematics in kindergarten mediated by the use of digital tools. I U. Kortenkamp, B. Brandt, C. Benz, G. Krummheuer, S. Ladel & R. Vogel (Red.),

- Early mathematics learning selected papers of the POEM 2012 Conference* (s. 207–222.) Springer.
- Imsen, G. (2005). *Elevens verden : innføring i pedagogisk psykologi* (4. utg.). Universitetsforlaget.
- Kieran, C. (2004). Algebraic thinking in the early grades: What is it? *The mathematics educator*, 8(1), 139–151. <https://gpc-maths.org/data/documents/kieran2004.pdf>
- Kunnskapsdepartementet. (2019). *Læreplan i matematikk (MAT01-05)*. Fastsett som forskrift. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/lk20/mat01-05?lang=nob>
- Micro:bit Educational Foundation (u.å.). Microsoft MakeCode. Hentet 04. november 2023 fra <https://microbit.org/code/>
- Mulligan, J. & Mitchelmore, M. (2013) Early awareness of mathematical pattern and structure. I L. D. English & J. T. Mulligan (ed.) *Reconceptualizing early mathematics learning* (s. 29–46) Springer. https://doi.org/10.1007/978-94-007-6440-8_3
- Nilssen, V. (2012). *Analyse i kvalitative studier: Den skrivende forskeren*. Universitetsforlaget.
- Norges teknisk- og naturvitenskapelige universitet. (u.å.). *LAB-TEd: Learning, Assessment and Boundary crossing in Teacher Education*. Hentet 07. juli 2023 fra <https://www.ntnu.edu/ilu/lab-ted>
- Tjora, A. (2021). *Kvalitative forskningsmetoder: i praksis*. (4. utg.). Gyldendal Akademisk.
- Postholm, M. B. & Jacobsen, D. I. (2018a). *Forskningsmetode for masterstudenter i lærerutdanning*. Cappelen Damm Akademisk.
- Postholm, M. B. & Jacobsen, D. I. (2018b). *Læreren med forskerblikk: Innføring i vitenskapelig metode for lærerstudenter*. Cappelen Damm Akademisk.
- Prediger, S., Gravemeijer, K. & Confrey, J. (2015). Design research with a focus on learning processes: an overview on achievements and challenges. *ZDM Mathematics Education* 47, 877–891. <http://doi.org/10.1007/s11858-015-0722-3>
- Rabardel, P. & Samurçay, R. (2001, 21.–23. mars). From artifact to instrument-mediated learning. *New challenges to research on learning*. International symposium organized by the Center for Activity Theory and Developmental Work Research. University of Helsinki, Finland.
- Rivera, F. (2013). *Teaching and learning patterns in school mathematics*. Springer Netherlands. <https://doi.org/10.1007/978-94-007-2712-0>
- Rogoff, B. (1995). Observing sociocultural activity on three plans: Participatory appropriation, guided participation and apprenticeship. I J. V. Wertsch, P. del Rio, & A. Alvarez (Red.) *Sociocultural studies of mind* (s. 139–164). Cambridge University Press.

- Rønning, F. & Burheim, O. T. (2020). Betydningen av en faglig og fagdidaktisk føranalyse i utviklingen av et undervisningsforløp. I V. Nilssen & S.-M. Høyenes (Red.), *Samtaleorientert matematikk: Et samspill mellom didaktiske og adidaktiske situasjoner* (s. 121-157). Fagbokforlaget.
- Senter for IKT i utdanningen. (2016, november). *Programmering i skolen* (notat, 1. reviderte utg.).
https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Stylianides, G. J. (2008). *An analytic framework of reasoning-and-proving. For the learning of mathematics*, 28(1), 9–16.
- Säljö, R. (2001) *Læring i praksis. Et sosiokulturelt perspektiv*. Cappelen akademisk.
- Trouche, L. (2003). From artifact to instrument: Mathematics teaching mediated by symbolic calculators. *Interacting with Computers*, 15(6), 783–800,
<https://doi.org/10.1016/j.intcom.2003.09.004>
- Van de Walle, J. A., Karp, K. S. & Bay-Williams, J. M. (2020). *Elementary and middle school mathematics: Teaching developmentally* (10. utg). Pearson Education.
- Verillon, P., & Rabardel, P. (1995). Cognition and artifacts: A contribution to the study of thought in relation to instrumented activity. *European Journal of Psychology of Education*, 10(1), 77–101. <https://doi.org/10.1007/BF03172796>
- Vihovde, E. H. (2023, 14. september) programmering (IT). I *Store norske leksikon*
https://snl.no/programmering_-_IT
- Vygotskij, L. S. (1978). *Mind in society. The development of higher psychological processes*. Harvard University Press.
- Walker, D. (2006) Toward productive design studies. I J. Akker, K. Gravemeijer, S. McKenney & N. Nieveen (Red.) *Educational design research* (s. 45–85) Taylor and Francis e-Library.
- Wijns, N., Torbeyns, J., Smedt, B. & Verschaffel, L. (2019) Young children’s patterning competencies and mathematical development: A review. I K. M. Robinson, H. P. Osana, & D. Kotsopoulos (Red.), *Mathematical learning and cognition in early childhood: Integrating interdisciplinary research into practice* (s. 139–161) Springer International Publishing. <https://doi.org/10.1007/978-3-030-12895-1>

Vedlegg

Vedlegg 1: Oppgaveark første undervisningsøkt

Oppgave 1

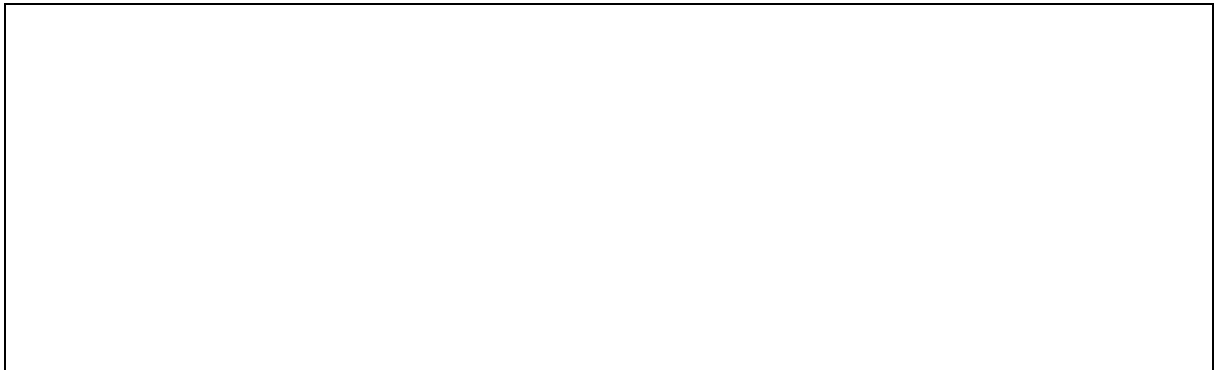
Se på denne koden under:

```
ved start
  sett Den første variabelen til 0
  vis tall Den første variabelen

når knapp A trykkes
  endre Den første variabelen med 8
  vis tall Den første variabelen
```

Hva tror du vil skje dersom du trykker en gang på knapp A?

Hva tror du vil skje dersom du trykker 6 ganger på knapp A? Kan du skrive dette som et regnestykke?



Gå inn på makecode.microbit.org og lag et nytt prosjekt.

Oppgave 2

Lag koden på bildet i oppgave 1 i microbit.

Sjekk hva som skjer når du klikker på knapp A først en gang, så 6 ganger.

Skjedde det du trodde ville skje?

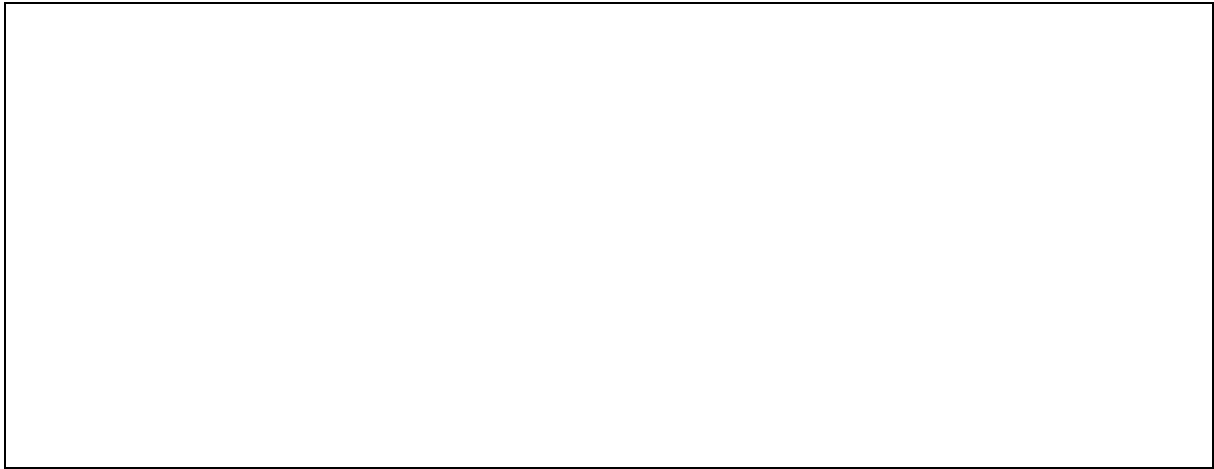


Oppgave 3

Få microbiten til å vise tallfølgen 88, 96, 104, 112, 120 ved å bruke variabler.

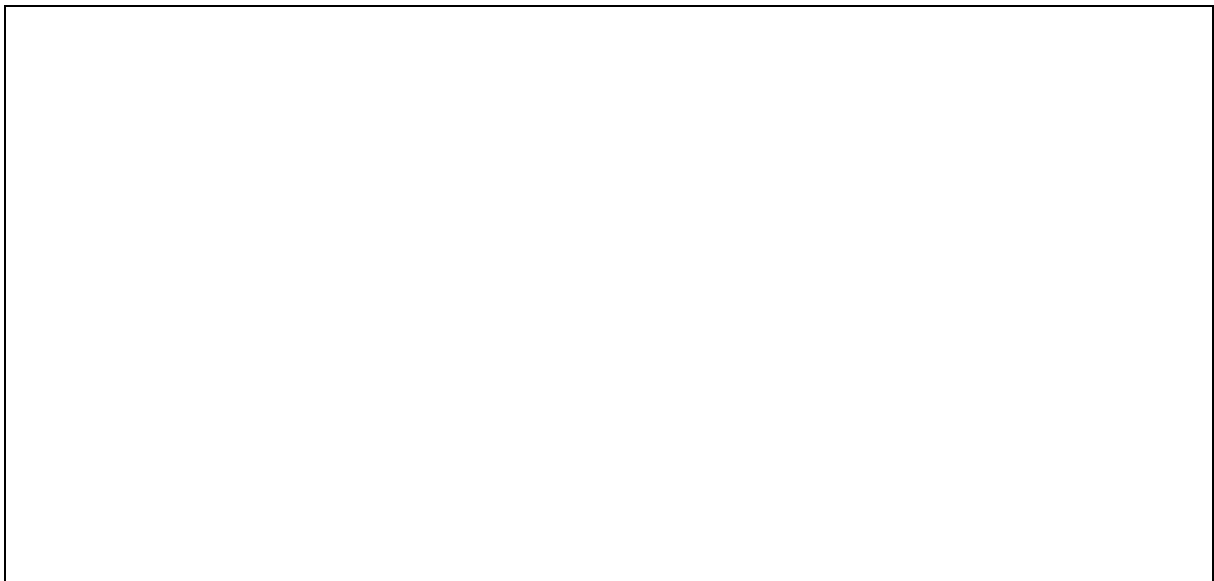
Oppgave 4

Beskriv tallfølgen i oppgave 3. Hva vil være de 3 neste tallene i tallfølgen?



Oppgave 5

Se på tallfølgen: $\dots, 21, 28, 35, 42, \dots, 77, 84, 91, \dots$. Se på hvor mye det er mellom de ulike tallene i tallfølgen. Hvilket tall må stå i «endre variabel»- blokka for å vise denne tallfølgen? Beskriv denne tallfølgen.



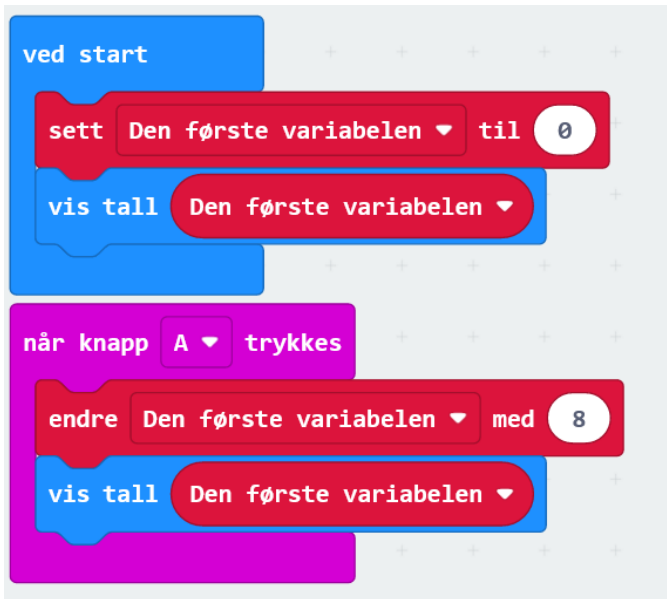
Vedlegg 2- Oppgaveark andre undervisningsøkt

Oppgave 6

Lag en kode i mikrobit som får mikrobit til å vise tallfølgen 8, 16, 24, 32, 40,

Oppgave 7

Se på denne koden:



```
ved start
  sett Den første variabelen til 0
  vis tall Den første variabelen

når knapp A trykkes
  endre Den første variabelen med 8
  vis tall Den første variabelen
```

Uten å trykke på mikrobiten: Hvor mange ganger tror du at du må du trykke på knapp A for å få 40? Forklar hvordan du kom fram til svaret:

Hvor mange ganger tror du at du må trykke på knapp A for å få 72? Hvordan kom du fram til svaret?

Lag koden i makecode. Test om svaret ditt var riktig ved å trykke så mange ganger som du trodde skulle gi 40 og 72. Husk å trykk på stoppknappen mellom testene dine, slik at koden begynner på 0 igjen.

Viste mikrobiten 40 og 72? Hvis ikke, hvorfor tror du det ble feil?

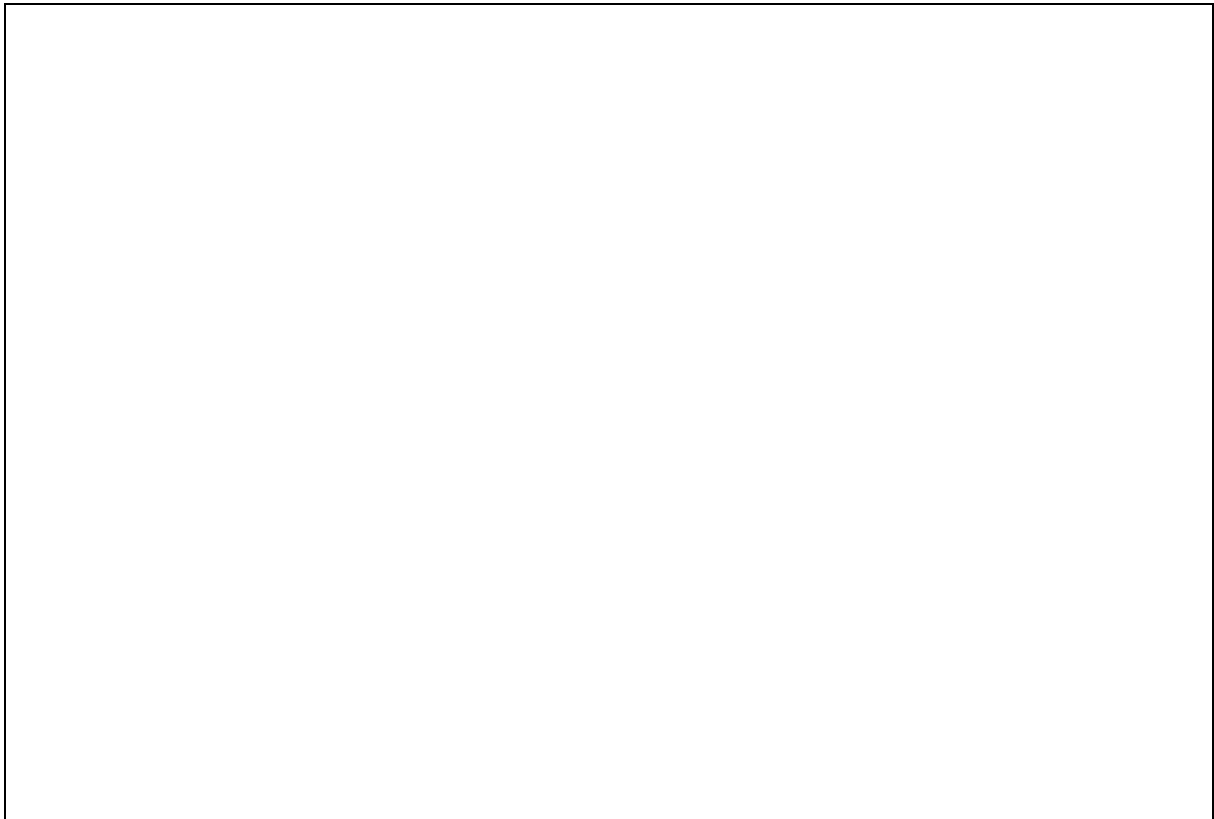
Oppgave 8

Vi bruker samme kode som er gitt i oppgave 2. Hvor mange ganger må man trykke på knapp A for at mikrobiten skal vise tallet 88?



Oppgave 9

Vi bruker samme kode som i oppgave 2. Hvor mange ganger må du trykke på knapp A for å få microbiten til å vise tallene 808, 816, 832,...? Vis utregning.



Oppgave 10

Gitt denne koden:

```
ved start
  sett Den andre variabelen til 0
  vis tall Den andre variabelen

når knapp A trykkes
  endre Den andre variabelen med 7
  vis tall Den andre variabelen
```

Hvor mange ganger må knapp A trykkes for å gi tallet 14? Forklar hvordan du tenkte for å komme frem til svaret ditt.

Hvor mange ganger må knapp A trykkes for å gi tallet 42? Forklar hvordan du tenkte.

Vedlegg 3: Informasjonsskriv til foresatte

Vil du delta i forskningsprosjektet

“Bruk av micro:bit i matematikkundervisning på barneskolen”?

Dette er et spørsmål til deg, som foresatt, om barnet ditt kan delta i et forskningsprosjekt hvor formålet er å undersøke ulike undervisningsopplegg i matematikk som bruker micro:bit. micro:bit er et programmeringsverktøy spesielt rettet mot barnetrinnet. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Vi er fire 5.-årsstudenter på Grunnskolelærerutdanningen på NTNU som skal skrive masteroppgaver våren 2023. I den anledning ønsker vi å samle inn datamateriale for å forske på bruk av programmering i matematikkfaget på barneskolen. Innsamlet datamateriale vil være grunnlag for masteroppgavene våre, og alle personopplysninger vil bli anonymisert. Denne forskningen innebærer observasjon, intervju, samt video- og lydopptak av elevene og læreren i klassen.

Vi ønsker å undersøke nærmere:

- Hvordan bruke micro:bit i matematikkundervisning?
- Hvordan kan matematikk inngå i programmering?
- Kan bruk av programmering i matematikkundervisning bidra til økt motivasjon for matematikkfaget generelt?

Masteroppgavene blir veiledet som en del av arbeid i forskningsprosjektet LAB-Ted.

Det anonymiserte datamaterialet innsamlet til masteroppgavene kan inngå i annen forskning knyttet til LAB-Ted-prosjektet.

Hvem er ansvarlig for forskningsprosjektet?

Norges teknisk-naturvitenskapelige universitet er ansvarlig for prosjektet.

Hvorfor får du spørsmål om å delta?

Elever som har [REDACTED] som matematikklærer ved 4. trinn ved [REDACTED] barneskole får tilbud om å delta i prosjektet.

Hva innebærer det for deg å delta?

Eleven vil delta i den planlagte matematikkundervisningen. I enkelte økter, der micro:bit blir brukt, vil det bli tatt video- og lydopptak av elevenes arbeid. Dette bli lagret på en

kryptert, ekstern harddisk, og vil slettes etter at materialet er transkribert og anonymisert.

I disse øktene vil vi snakke med elevene om deres tanker og arbeid med micro:bit. Dette kan enten foregå underveis i klasserommet, eller dersom det er praktisk umulig, vil vi be elevene om en samtale på grupperom etter endt økt. Potensielle spørsmål kan være:

- Hvordan synes du det fungerer å bruke micro:bit i undervisningen?
- Hvordan tenkte du i løsningen av denne oppgaven?
- Synes du programmering er en motiverende arbeidsform i matematikkfaget?
- Hvordan brukte du kodebiten i løsningen din?
- Syns du det er noe som er vanskelig med bruk av micro:bit?

Hvis du som foresatt ønsker en fullstendig intervjuguide, ta direkte kontakt med oss.

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.

Elevene vil også bli spurt om de ønsker å delta i forskningsprosjektet, og elever som ikke ønsker dette vil ikke delta. Alle elevene vil få samme undervisningsopplegg, men det vil ikke bli samlet inn datamateriale av elever som ikke ønsker å delta.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

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

Personer som vil ha tilgang til opplysningene som blir samlet inn er: studentgruppa, praksislærer og veiledere fra NTNU. Alle personopplysninger vil bli anonymisert, og datamaterialet vil bli lagret på en innelåst og kryptert ekstern harddisk.

Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?

Prosjektet vil etter planen avsluttes/oppgavene blir godkjent, noe som etter planen er juni 2023. Alle former for opptak som blir gjort under gjennomføring av prosjektet, vil anonymiseres fortløpende og selve opptakene vil slettes senest ved prosjektslutt.

Siden dette prosjektet er en del av et større prosjekt (LAB-Ted), vil de anonymiserte data fra prosjektet bli beholdt noe utover prosjektperioden, men vil slettes senest ved utgangen av juni 2025. Temaet for prosjektet er høyaktuelt for utviklingsarbeid og forskning i norsk skole, og derfor vil det anonymiserte materialet bli beholdt noe utover perioden masterprosjektene forgår. Formålet med den videre lagringen er at forskere ved NTNU, ILU, spesifikt Benedikte Grimeland, Yvonne Grimeland, Oda Tingstad Burhheim og Torunn Klemp kan bearbeide materialet videre for å publisere forskningsarbeid slik at resultatene fra prosjektet når en større del av lærerutdanningsfeltet i Norge. Det er kun de nevnte forskere som vil ha tilgang til materialet. Materialet vil bli lagret på kryptert disk som ikke er koblet til nett, innelåst hos NTNU, og eid av NTNU.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra NTNU har Personverntjenester vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter

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

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

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

- NTNU ved Yvonne Grimeland. Tlf. 48114352. Epost: Yvonne.grimeland@ntnu.no
- Vårt personvernombud: Thomas Helgesen. Tlf. 93079038. Epost: thomas.helgesen@ntnu.no

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

- Personverntjenester på epost (personverntjenester@sikt.no) eller på telefon: 53 21 15 00.

Med vennlig hilsen

Guro Kjeldstad
(forsker)

Vilde Dahle
(forsker)

Ellen-Marie Kristiansen
(forsker)

Eirun Flaten Sandvin
(forsker)

Yvonne Grimeland
(veileder)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «Bruk av micro:bit i matematikkundervisning på barneskolen», og har fått anledning til å stille spørsmål.

Jeg samtykker til at mitt barn har tillatelse til:

- å delta i intervju
- å delta i lyd-opptak
- å delta i video-opptak

Jeg samtykker til at opplysninger om mitt barn behandles frem til prosjektet er avsluttet

(Signert av prosjektdeltakers foresatt, dato)

