UiA University of Agder

Master

Student’s reasoning with programming

A study on elementary grade student’s reasoning when working with geometry problems using programming

ROBERT EIDSÅ

SUPERVISORS
Shaista Kanwal
André Martiny

University of Agder, 2022
Faculty of Engineering and science
Department of Mathematics

Preface

It is almost unbelievable that it has been five years already, the time has passed by extremely fast. But now I am actually writing the preface of a master thesis I have been working on for the last months as a conclusion to those five years. The knowledge utilized on the work on this thesis is traceable back through many inspiring lectures and prior work the last couple years whenever there was a choice of subject for graded papers. When I heard the possibility of doing research into this field, I knew I had to go for it, as my good friend and fellow student Julian Norum Breland said then, it basically had my name on it already.

I am somewhat of an eccentric, and this thesis has been an opportunity to learn more and work with something I am very interested and inspired in already. Although this thesis, and these last quick five years of learning is nearing conclusion, that does not mean I am intending to walk away from gaining further enlightenment and knowledge. As the great Richard Feynman said once in an interview with the BBC, I am paraphrasing a bit here, the pleasure is in finding things out.

I have to direct thanks to the friends and family I am surrounded by who has been supportive by being understanding of the stress and work that has gone into it, and who simply has been curious about it. Also, my colleagues at work who has been understanding and patient, when I have been more or less distant because of at times massive workload on account of writing this thesis.

A special thanks to the supervisors on this thesis Shaista Kanwal and André Martiny, who has been absolutely brilliant to work with. I do not think it could have been better. During meetings I have always come away from it with a better sense of direction and gotten more optimistic for the project. Especially the last meeting. Before this I was feeling like a lot would not work out with the thesis, but after this meeting I had new gusto and a belief that it could be done.

Robert Eidså
Kristiansand, May 2022

Summary

On the LK20 curriculum in Norway programming has been introduced especially in mathematics and science as a way of exploration. This study's focus is on students' reasoning when working with abstract data output from a computer program. The research question is thus:

What characterizes students' reasoning when solving geometry problems through programming?

This is a case study using a group of six elementary grade students. There was first held an arrangement where we explored a Python program that concerned the volume and surface area of a rectangular prism. The program printed out lists of dimensions and box sizes based on what was set as minimum and maximum dimensional values, and some dimensional constrictions to the design of the boxes. The list that was printed out showed the volume and the surface area of these boxes. Afterwards interviews were held with three of the participating students, where the goal was to figure out how the students understand the tasks, and how and if they managed to reason how the sizes and dimensions change based on the constrictions that were set in the programs and the tasks.

The data was analyzed using a framework of deductive and inductive reasoning. In particular, it was analyzed how the rule, case, and result of the students' reasonings where attempted recognized, and how the workings of the reasoning, the antecedent and the consequent plays a role in the students' responses.

The study shows the importance of understanding the mathematical task. Both in understanding what the task asks for in the form of an answer or solution, meaning that it is important to practice and help students understand that "*a solution*" is not the same as "*the solution*". And it is important to understand the mathematical workings of the task before trying to solve it, without understanding the task the reasoning could be done on a false premise.

Sammendrag

På den nye læreplanen i Norge LK20 så har programmering blitt introdusert spesielt i matematikk og naturfag som en måte å utforske på. Denne studiens fokus er på hvordan elever resonerer når de jobber med abstrakte data fra et data program. Forskningsspørsmålet er dermed:

Hva karakteriserer elevers resonnement når de løser geometrioppgaver med programmering?

Denne studien ble gjort som en casestudie hvor det ble brukt en gruppe på seks elever. Det ble først holdt en undervisningsøkt hvor vi utforsket et Pythonprogram som handlet om volumet og overflatearealet på et rektangulært prisme. Programmet printet ut lister med dimensjoner og størrelser på bokser basert på hva som ble satt opp som minimum og maksimum dimensjonale verdier, og noen dimensjonale begrensinger på utformingen av boksene. Listen som ble printet ut viste volumet og overflatearealet av boksene. Etterpå ble det holdt intervjuer med tre av de deltakende elevene, hvor målet var å finne ut hvordan elevene forstod oppgavene, og hvordan og hvis de klarte å resonere seg frem til løsninger til hvordan størrelsene og dimensjonene endrer seg basert på begrensningene som var satt i programmene og oppgavene.

Dataene ble analysert ved å bruke et rammeverk bestående av deduktiv og induktiv resonering. Spesifikt hvordan regler, case, og resultater av elevenes resoneringer ble forsøkt gjenkjent, og hvordan funksjonene i resonering, antecedent og konsekvens spiller en rolle i elevenes svar.

Studien avdekker viktigheten av å forstå den matematiske oppgaven. Både med tanke på forståelse for hva oppgaven ber om i form av svar, i betydningen av at det er viktig å trene på og veilede elevene til å forstå at *«en løsning»* er ikke det samme som *«løsningen»*. Og det er viktig å forstå de matematiske konseptene i en oppgave før man prøver å løse den, uten forståelse av oppgaven blir resoneringen gjort på feil premisser.

Table of contents

Introduction

This thesis explores students’ reasoning when working with geometry problems using programming. Through an arrangement with a Python program and interviews with a group of students, it tries to study the students’ reasoning process when trying to figure out solutions to the tasks. The students in the study will not have programmed the programs themselves, but they will be provided with skeleton programs and these will be utilized in the arrangement and interviews.

In this paper the motivations for the study and the research question will be presented first. Thereafter comes an overview of the research literature that is relevant for the study, from some of the first research done into programming in schools in the 60’s and 70’s, a brief explanation on computational thinking, and a recent study on computational thinking. Thereafter a presentation of the theoretical framework of reasoning that has been used for analysis. Then there is a description of the methodology of the study, showing the details of how the data has been collected and used. Then the data from the study is presented and analyzed using the framework presented earlier. And then the results from the analysis are presented and reflected on its implications on mathematics education. Lastly there is a conclusion of the paper with a personal comment on the study.

Motivation for the study

One of the motivations for this study is even though there is quite a lot of work that has been done one learning mathematics while utilizing programming, there is little work that has been done with reasoning when using programming in mathematics. And since the Norwegian LK20 curriculum, there is quite a bit of programming that is in the elementary school education. The focus on the curriculum regarding programming is exploration, and algorithmic thinking. This study ties into that in the sense that it tries to understand the possibilities for learning when utilizing programming in mathematics education, by studying the student’s reasoning. It is interesting also to see the outcome of using skeleton programs and how the students reason on this, opposed to having the students program the programs themselves.

The research question

The research question is *"What characterizes students' reasoning when solving geometry problems through programming?"*. This in turn is tied to what kind of potential for learning outcome is present when working on geometrical problems with programming, and what is the focus of the students' thinking when working in this manner. In the words of Seymour Papert, "You can't think seriously about thinking without thinking about thinking about something." (2020, p. 9).

Background and previous research

The beginning of programming in schools and briefly Papert’s constructionism

In the late 60’s Seymour Papert and colleagues at MIT developed Logo, a programming language specifically designed for children (Resnick et al., 1988, p. 15). Logo was designed as a dialect of the programming language Lisp, with simplified syntax (Logo Foundation, 2015). In early 2000’s the programming language scratch was made by the Lifelong Kindergarted Group at MIT, which utilizes block programming instead of text based programming as Logo does (Logo Foundation, 2015). Today scratch is in wide use in schools. However, the interpreted general purpose programming language Python is also being used more and more in education because of its relatively simple and easy to learn syntax.

Papert realized that often times children came into contact with computers, computer were used to program the children, whilst with Logo this was reversed, the children teach the computer, and while doing so the children explore how they themselves think (Papert, 2020, p. 19). He came away with this idea of the child as an epistemologist after he had worked five years with Piaget in geneve (Papert, 2020, p. 19). This eventually led to Papert creating constructionism which Papert and Idit Harel writes about in their book constructionism published in 1991. Constructionism is not just limited to programming in mathematics education, the ideas are transferable to other areas of education as well. As they point out in the first chapter of the book; “The simplest definition of constructionism evokes the idea of learning-by-making and this is what was taking place when the students worked on their soap cultures” (Papert & Harel, 1991, p. 7), it is not specifically only when learning programming but a way of learning. Another important part of constructionism is the recognition of two different styles of working on a project, or a task. The “bricoleur” and the engineer. The first being focused on the artistic and pragmatic way of doing things, trying what tool fits and what don’t, the latter is more analytical and makes a plan before executing it (Papert & Harel, 1991, p. 7) (Papert, 1993, pp. 146-148). This is relevant for this study, because if a student that could be characterized as a “bricoleur”, or shows a tendency to bricolage, this might have an explanation value to the reasoning of the student. While the “engineer” typically will develop a clear plan, and working systematically when solving problems, the “bricoleur” might be leaning more towards a more chaotic, seeing what works and what don’t mentality. Even though it is not necessarily as easy as being one or the other, these things tie into how the student might reason a specific task or problem.

Computational thinking

Computational thinking is described by Wing as the process of formulating a problem so that a machine can do the calculation (Wing, 2006, 2008, 2011). Meaning in practical mathematical terms going from a task, and then making a method of solution that can be written in the form of a computer program, and then executed to solve the mathematical task. Wing also states the most important part of CT, and that is abstraction (2011). “Abstraction gives us the power to scale and deal with complexity” (Wing, 2011, p. 2), it can be thought of as that you don’t have to know how the calculator works to use it. You do not have to understand the complex circuitry and programming going on within the calculator, but it is still possible to use it as a tool to help in understanding of mathematical concepts. This is true for many parts of modern society, be it smart phones, website or other media, cars, the list goes on. Even a programming language is an abstraction of something more complex. For example, we can use the programming language Python, or C, instead of writing in assembler language, or binary code. CT draws on both mathematical thinking and engineering thinking, computers are limited to the physical constrictions of their construction which mathematics is not, on the other hand using software it is possible to build virtual worlds where everything is possible, and only limited by our imagination (Wing, 2011, p. 2).

Connecting mathematical problem-solving with CT and programming

Hadjerrouit and Hansen (2022) propose a three step iterative approach for connecting mathematical problem-solving with CT and programming (p. 200).

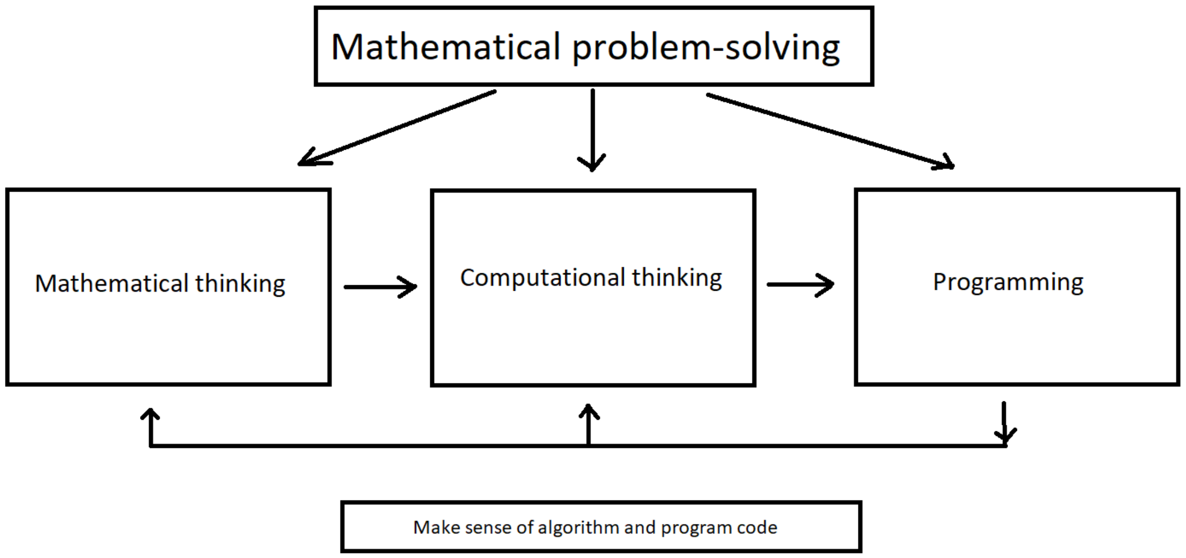


Figure 1: Illustration based on Hadjerrouit and Hansen (2022, p. 201)

Hadjerrouit and Hansen (2022, p. 200) suggest that before the student writes the program, the student need to formulate an algorithm, and before being able to formulate the algorithm it is necessary to have a mathematical understanding of the problem. Hadjerrouit and Hansen also point out that this process is not linear, it is fully possible that in the process of formulating the algorithm and writing the program, that the student will reiterate back and forth between the different stages in the process (p. 201).

The model illustrates a framework for going from a mathematical problem to writing a program that solves this problem. After the program has been written it is possible to look back at and experiment with the algorithms and program, to get a deeper understanding of the mathematical problem and concept (Hadjerrouit & Hansen, 2022, p. 201). The students in this study, unlike Hadjerrouit and Hansen’s study, they will not have programmed the programs themselves; they have been prewritten prior to the arrangement. It might be thought of as a thought experiment; if the students were already sufficiently proficient in programming, then it might be possible to find reasoning between a mathematical concept and the abstracted output of a computer program. This is in broad view the "Make sense of algorithm and program code" part of the model from Hansen and Hadjerrouit described in prior research.

In the study, Hadjerrouit and Hansen did a case study of two first-year undergraduate students enrolled in a course on programming with applications in mathematics (2022, p. 202), the students were met with a mathematical task they were to solve by using the programming language MATLAB. One of the students made good progress of the task by writing up a plan and making an algorithm, before doing the programming part of the task, while the other student struggled to work systematically with the task, and then was assisted quite a lot with the task during the interview (Hadjerrouit & Hansen, 2022). Hadjerrouit and Hansen summarizes the importance of a good mathematical understanding of the task, to facilitate the usage of computational and algorithmic thinking skills (Hadjerrouit & Hansen, 2022).

Mathematical reasoning and sense-making using programming

Constanta Olteanu did a study into reasoning and sense-making when using scratch as a programming tool (2020). In the study the students worked with Scratch to explore the exterior angle concept in different regular polygons (Olteanu, 2020). The data and the students reasoning was analyzed using a combination of variation theory and a philosophical rhizomatic perspective in the form of dividing tasks and problems into multiple features much like a "rhizome", with many steps interacting with each other (Olteanu, 2020). This will not be covered in detail here, however, some of the results of the study is interesting and relevant for this study.

When using Scratch as a pedagogical tool, the interventions from the teacher must be thoughtful when the student is experiencing difficulties with a critical aspect of concepts, the tasks should be rhizomatic in the sense that they lead the students to critical aspects which could be intervened if needed, the critical aspects must be identified, and there must be a pattern of variation in the tasks the students are presented with (Olteanu, 2020). Also, they note the importance of not simply exposing the students to mathematical tasks, because the students need to develop their own mathematical reasoning through collaboration and communication (Olteanu, 2020). And lastly, the importance of the teachers' own reasoning and sense-making about programming and mathematical concepts (Olteanu, 2020).

Theoretical perspective

Different types of reasoning

Reid and Knipping (2010) present a few different types of reasoning and explains ways this can be used in research in mathematics teaching. They focus mainly on deductive reasoning, inductive reasoning, abductive reasoning, and reasoning by analogy. But they also briefly mention avoiding reasoning by making reference to an authority, avoiding reasoning using a tool or a technique, and transformational reasoning.

The way the reasonings are distinguished is by how the different reasonings use *cases, rules,* and *results* (Reid & Knipping, 2010). Reid and Knipping (2010) describe *cases, rules,* and *results* as:

> "A *case* is a specific observation that a condition holds. A condition describes an attribute of something, or a relation between things. The statement "Chino is a dog" is a case, in which *being a dog* is the condition. A *rule* is a general proposition that states that if one condition occurs then another one will also occur. "Dogs are animals" is a rule. The conditions "being a dog" and "being an animal" are linked. A *result* is a specific observation, similar to a case, but referring to a condition that depends on another one linked to it by a rule. "Chino is an animal" is a *result* in this example." (p. 83)

The two most relevant reasonings for this study is deductive reasoning and inductive reasoning. Deductive reasoning is a rule and a case leading to a result, inductive reasoning is one or several similar cases and results leading to a rule (Reid & Knipping, 2010). The main focus will be on deductive reasoning, but inductive reasoning will be briefly explained and used when applicable for gaining another perspective.

Deductive reasoning

Deductive reasoning comes from deductive logic. Both deductive logic and inductive logic goes all the way back to the ancient Greeks. In general, deductive reasoning gives that a *case*, and a *rule*, gives a *result* based on those. Presented here are two types of syllogisms and their fallacy counterparts. *Modus ponens* (affirming the antecedent), and *modus tollens* (denying the consequent).

For example, “All men are mortal” (rule), and “Socrates is a man” (case), which results in “Socrates is mortal” (result). This is a syllogism of the “modus ponens” type, meaning that if **A (man)** then ***B (mortal)***, and since **A** (Socrates is a **man**), then ***B*** (Socrates is ***mortal***). The rule is made up of an antecedent, and a consequent. The rule in this syllogism states the antecedent as “man” and the consequent as “mortal”, so if man → mortal. It is an *affirmation of the antecedent*, which is a valid logic. If instead given, “All men are mortal”, and “Socrates is mortal”, therefore “Socrates is a man”. In this case the logic would be invalid because it is a *fallacy of affirming the consequent*. Socrates could be a cat, or an alien, and still be mortal.

To see how this could be used to analyze how a student reasons on a mathematical task, let us look at an example. In one of the tasks featured in the teaching arrangement in the study, there is a constriction to the dimensions of the rectangular box. That the sum of the three dimensions x, y, and z of a box must be 15, this is a rule. So, if the sum of the three dimensions is 15 then it is a valid box for the task. Let us say that the case is that x is 1, y is 1, and z is 13, then the result would be that the sum is 15 and it is a valid box. This is a valid logic of affirming the antecedent. But if say the sum of the dimensions are 15, it is not valid to argue that the dimensions of the box are 1, 1, and 13. This would be a fallacy of affirming the consequent.

An example of the “modus tollens” type would be, “All men are mortal”, and “Socrates is immortal”, therefore “Socrates is not a man”. This is also called *denying the consequent*, a valid logic. Since Socrates is not mortal, therefore Socrates cannot be a man, because all men are mortal. The “mortal” part is the consequent. If instead given, “All men are mortal”, and “Socrates is not a man”, therefore “Socrates is immortal”. The logic would then be false, it would be a *fallacy of denying the antecedent*, where the “men” part is the antecedent.

An example of this for analyzing a mathematical reasoning could be that if we have the same rule from the previous example; the sum of the three dimensions of the box must be 15. Then it would be a valid logic of denying the consequent to say that if the dimensions are 1, 2, 13, the sum is not 15, then it is not a valid box for the task. The three dimensions being 15 as the antecedent, and valid box as the consequent. If you say that the sum of the dimensions is not 15, therefore the dimensions must be 1, 2, 13, it would be a fallacy of denying the antecedent.

Reid and Knipping (2010) point out that it is sometimes claimed that deductive reasoning cannot be used to explore because all the information is already present, but they argue that it is possible to *experience* the discovery of a new truth through deductive reasoning. They tie this idea to the idea of a “coded message”, the message can be passed on, but having the message is not the same as knowing the message. Decoding the message will give the experience of making something that was implicitly known, into something that is explicitly known, not unlike learning something new.

Inductive reasoning

Reid and Knipping (2010) describe inductive reasoning as the opposite in many ways of deductive reasoning. While deductive reasoning generally does not lead to new knowledge, inductive reasoning works by going from specific cases to general rules, new knowledge is concluded based on what is known, and inductive reasoning is not certain it is only probable (Reid & Knipping, 2010).

Inductive reasoning might have some explanation value, in the sense that it can show the reasonings of the students as the reasoning develops. When analyzing the reasoning of the students in this study, it is possible to try to see the reasonings in an inductive perspective. The deductive reasoning might be easier to recognize, as this has a bit more explanative structure. This in the sense that the student explains how he or she finds an answer to a mathematical task and argues it must be right. For example, $5 + x = 10$, x must be 5 because you need to find what can be added to 5 to make 10. The inductive reasoning might conclude this rule or way of solving said task based on multiple tests with different numbers and experience with multiple tasks of similar type. Without going into constructivism and the construction of viable cognitive structures, it is a construction of knowledge based on finding patterns and correlations.

Method

Research design

As described in Bryman's social research methods a case study is a detailed and intensive analysis of a single case (Bryman et al., 2021, p. 59). In this study the focus is a small group of students in 9th grade in an upper elementary Norwegian school, and how they reason between programming and the mathematical concept of area and volume of a geometric object.

It is a qualitative study using observations and a semi-structured interview to report on the students reasonings. The data thereafter coded by writing in the margins of the transcript of the interviews and marking relevant passages in the conversations. This is then analyzed by studying the connections the students make between the program's output and the mathematical ideas.

Participants for the data collection

Disclosure regarding insider/outsider

It is necessary to disclose that the researcher, me, have been working as a temporary teacher on the school which the data collection has taken place. Bryman et al state that "…professionals who decide to carry out research in their workplace could be defined as research insiders." (p. 133). However, I was not familiar with the particular students in question prior to the research, on account of not having had any lectures in the students' class. On the other hand, it is most likely that the students have observed me walking in the halls of the school in between classes, and therefore knowing me as part of the school staff, rather than an outside researcher. This is unlikely to have adverse or coloring effects on the results.

Context and participants

Since the mathematical concept that is being studied is geometry, the participants needed for the data collection requires 9th grade students. The reason for this is that the curriculum for upper elementary mathematics in Norway has specific concepts on different years. Of the upper elementary curriculum, the only year that have geometry is 9th grade. The relevant concepts from the current curriculum LK20 on 9th grade mathematics are as follows (translated by me) (Kunnskapsdepartementet, 2020):

- Describe, explain, and present structures and developments in geometric patterns and number patterns.
- Explore the characteristics of different polygons and explain the terms geometric similarity and congruence.
- Explore, describe, and argue the correlation between the sides of a triangle.
- Explore and argue how the solutions of geometric problems changes if the prerequisites for the solution are changed.
- Explore and argue about formulas for area and volume of three-dimensional objects.

There are two arguments for striving to use students from this grade: Firstly, it benefits the students, since they are already expected to learn and study geometry this year. Secondly, it also benefits the transferability of the research, being more realistic in the way of what is to be expected of the students' comprehension of the mathematical concepts of that year.

Description of the participants, and recruitment to the project

The school of which the participants were collected from, is an outside main city upper elementary school in the southern part of Norway.

Before the particular students were enrolled in the project, I contacted the teacher of one of the 9th-grade classes at the school. This teacher is both the mathematics teacher and the contact teacher of the class. This made it fairly easy to plan and make sure the students were available for the arrangement and interviews.

A few days before the data collection I visited the class and asked if there were anyone interested in participating in a research project regarding programming and geometry. One of the reasonings behind this way of recruiting is that it might reap participants that are interested and open to trying something different, as opposed to feeling they are missing out on the regular scheduled classes. A few students raised their hands. The plan was to get around five students, and six students had raised their hands, so that worked out fine. After everyone had gotten the consent form (see appendix), they were instructed to read it fully and sign it, and also get signatures from their parents. They were also informed that it is fully optional, and they can opt out at any time without any consequences to them.

During the data processing the names of the participants were anonymized according to an alphabetical system in the order the data was processed. The first data that was processed was the interviews, so these participants got anonymized names first. Starting with A, then a name was made up for that letter, then B, then a name was made up for that letter, and so on. Alf was interviewed first but mentioned Bjorn who was not interviewed during the interviewed, therefore Bjorn got anonymized before Christine and Dan who was interviewed after Alf. There were six participants in total: **A**lf, **B**jorn, **C**hristine, **D**an, **E**mma, and **F**ilipa.

Data collection

Before the data collection there was multiple things that did not go as planned on account of the ongoing covid-19 pandemic at the time. Which ended up affecting the time that was available for doing the arrangement and doing the interview, ultimately ending in not being able to interview all of the participants in the study. However, the data that ultimately was collected still ended up having some interesting findings.

The collection of data is divided into two parts. The first is a teaching arrangement where the goal was for the group of students to work and familiarize themselves with the geometrical problem, and the programming used in the problem. The second part is an interview where the goal is to figure out what the students got out of the arrangement they participated in. The arrangement and the interviews where on two different successive days, first the arrangement, and then the interviews the day after. Both the interviews and the arrangement were victims of time constraints, on account of the busy class schedule. The interviews especially, being in the end of the last day before a holiday and there being a schoolwide happening right after the interviews.

The arrangement

Recording and setup

The recordings of the arrangement did end up not being used for analysis, instead the data used in the study was from the interviews only. However, the arrangement was recorded on video using a Go-pro camera. The reasoning mostly behind using a video recording as well as audio recording is that it makes it easier when transcribing the arrangement, for example who's talking to who on the recording. Although it was not used for analysis it was useful when writing the description of the arrangement. In the interviews afterwards video recording is not necessary because there is not more than two people present in the recording, the interviewer, and the interviewee.

Originally the plan was also to make screen recordings on the student's laptops, so that it would be possible to see what and how the students worked with the programs. Windows have this functionality built in, in the menus of "Xbox game bar". It is available by holding down the "windows" key and pressing the "G" key on the keyboard. However, it turned out that on the municipal school pc's the students' use, this function has been disabled. This on the other hand, as further elaborated on in the next sections, turned out to be less crucial.

Participants, the room, and seating

The group room that was used was not ideal, it had a whiteboard, and a few chairs. There were not enough chairs for everyone, so some of the students ended up sitting on the desks around the room with their computer on their lap. Another thing that would have been nice to have would be a projector or a tv for showing the students examples and guiding with a big screen, although it worked out fine without. Instead of the big screen the whiteboard was used to write some of the commands. There was also prepared beforehand, a USB memory stick with a link to the python interpreter on trinket.io, and a text file with the ready-made programs that also helped mitigating the need for a big screen. The content of this memory stick is elaborated more on in the next sections.

Hiccups, stressed for time, and rectifying improvisation

The school timetable was set up as 60 min classes, so after the equipment was set up there was still over 50 minutes left of class. As we started setting up the computers for the arrangement and got started with the first task, it became clear that we will not be able to get through all of the tasks of the problems that was planned (See appendix for the problem sheet). First, all of the students got one USB memory stick each, with prewritten Python scripts and link to the Python interpreter on trinket.io. Then everyone was instructed to double click the link to get to the Python interpreter, and then open up the text file on the memory stick and copy the script to the input textbox on the trinket site (See the appendix for the prewritten Python scripts).

Originally the plan was to work through all of the tasks on the problem sheet. But it was clear after a little while that we would not be able to get through all of them. First, we tried the function that prints out the surface area of a box. Then we made Python print out a list of dimensions and surface area of all boxes where the sums of the three dimensions are 15, and the numbers for x, y, and z are integers. At this point the realization came that we would not be able to do all of the tasks on the problem sheet. Since the purpose of the arrangement was for the students to think about the correlation between the surface area and the volume when there are constraints on the dimensions of the box, we moved on to the last task on the problem sheet.

In hindsight the problem sheet was too elaborate and had too many subtasks. Luckily the tasks were fairly modular, and this made it easily possible to skip ahead.

The problems and the programming activity

Before starting the programming, some time was spent explaining the x, y, and z dimensions of a box by explaining it on the whiteboard. Thereafter, a short explanation of what a defined function is in Python. The names for the functions were set up in Norwegian to be more descriptive and be easier understood by the students. The first program that was run was the function overflate(x,y,z) (surface, or surface area) with some dimensions for x, y, and z by writing print(overflate(x,y,z)) at the bottom of the input textbox on trinket, this prints out the volume of a box with these dimensions. Making sure everyone got an output from this, we moved on to the next task.

The first proper task was to get the script to print out a list of the dimensions and the surface area of the box using integer values for x, y, and z. There was also a constriction that the sum of the numbers for x, y, and z had to be 15. This will for example give 1, 1, and 13, or 6, 7, and 2 as possible values for x, y, and z. The sum 1 + 1 + 13 = 15, and 6 +7 +2 = 15. To accomplish this the students was instructed to write the following at the bottom of the trinket input textbox:

dimensjonerOgOverflater(dimensjonSorter(dimensjonsgenerator(1, 15), 15))

We first did the program described above. The translation of the function name is "DimensionsAndSurfaceArea". Since this is almost the same as the next one, only a bit more elaborate, we will move on to that. The difference between them is that the next script described in the next section also includes the volume of the box in the output. However, we did talk a bit about the output of the first script in the arrangement, trying to compare the different boxes whilst looking at the 3D models that are also described in later sections.

The next task the students was instructed to do was to get Python to print out a list of all possible dimensions, surface area, and volume of the box using integer values for x, y, and z. The prewritten program had this possibility using a combination of the defined functions. To accomplish this the students was instructed to write the following at the bottom of the trinket input textbox:

dimensjonerOverflaterOgVolum(dimensjonSorter(dimensjonsgenerator(1, 15), 15))

dimensjonerOverflaterOgVolum(sorterteDimensjoner)

```
#Funksjon som printer ut alle dimmensjonene, overflatearealet OG volumet som blir matet inn i den.
def dimensjonerOverflaterOgVolum(sorterteDimensjoner):
    for t in sorterteDimensjoner:
        x = t[0]
        y = t[1]
        z = t[2]
        print("Dimensjoner: x=", x, "  y=", y, "   z=", z, "   Overflate=", overflate(x,y,z), "    Volum=", volum(x,y,z))
```

The direct translation of this function is “dimensionsSurfaceareaAndVolume(sortedDimensions)”. This is a function that take one argument consisting of a list of lists of presorted dimensions that need to be generated in a different function or put in manually. It then calls for the surface and volume functions as needed to print this line for line containing the dimensions, surface area, and the volume of the boxes. Below is an example of manually feeding the function two lists with ones and twos as x, y, and z.

```
dimensjonerOverflaterOgVolum([[1,1,1],[2,2,2]])
('Dimensjoner: x=', 1, '  y=', 1, '   z=', 1, '   Overflate=', 6, '    Volum=', 1)
('Dimensjoner: x=', 2, '  y=', 2, '   z=', 2, '   Overflate=', 24, '    Volum=', 8)
```

dimensjonSorter(listeMedDimensjoner, sumKonstant)

```
#Funksjon som kan sortere ut kun de kombinasjonene av dimensjoner som passer med begrensningen på summen av dimensjonene.
def dimensjonSorter(listeMedDimensjoner, sumKonstant):
    usorterteDimensjoner = listeMedDimensjoner
    sorterteDimensjoner = []
    for r in usorterteDimensjoner:
        x = r[0]
        y = r[1]
        z = r[2]
        if (x+y+z) == sumKonstant:
            sorterteDimensjoner.append([x,y,z])
    return sorterteDimensjoner
```

The direct translation of this function is “dimensionSorter(listOfDimensions, sumConstant)”. This function takes two arguments, and it can sort a list of lists with dimensions. The first argument is a list of dimensions that can be generated in a different function or just typed in, although this would be very tedious. The second argument is the sum constant, this is where the function gets the parameter to which it sorts out all other combinations of dimensions that does not have the required sum of x, y, and z. Below is an example of feeding the function a few lists of dimensions, note that this function does not have a built in print function so to get an output it needs to be set up within a print function so that the return gets printed.

```
print(dimensjonSorter([[1,1,1],[2,2,2],[1,1,13],[3,3,9]], 15))
[[1, 1, 13], [3, 3, 9]]
```

First, we can see the two lists of dimensions from the previous example, these do not equal to 15 when they are added together. Second, the two new lists of dimensions consisting of 1,1,13 and 3,3,9 add up to 15 and gets returned to the print function and gets printed.

dimensjonsgenerator(minst, storst)

```
#En funksjon som genererer lister med x,y,z dimensjoner
def dimensjonsgenerator(minst, storst):
    x = range(minst, storst)
    y = range(minst, storst)
    z = range(minst, storst)
    dimensjoner = []
    for n in x:
        for o in y:
            for p in z:
                dimensjoner.append([n,o,p])
    return dimensjoner
```

The direct translation of this function is “dimensiongenerator(minimum, maximum)”. This function takes two arguments, the smallest number, and the largest number, or in other words the smallest dimension and the largest dimension. The function generates a list of lists of dimensions for x, y, and z. Let’s say we insert 1 as the minimum, and 3 as the maximum, it will count up to 3 but not 3. It will then create a list of lists that iterates from 1,1,1, the next list of numbers will be 1,1,2, the next 1,2,1, and so on. The return from this function can be inserted into the dimension sorter and then sorted for the required constrictions. Below is an example of the output that gets returned from the dimension generator when it is set up with 1 and 3, note that the function needs to be put into a print function to get an output.

```
print(dimensjonsgenerator(1, 3))
[[1, 1, 1], [1, 1, 2], [1, 2, 1], [1, 2, 2], [2, 1, 1], [2, 1, 2], [2, 2, 1], [2, 2, 2]]
```

dimensjonerOverflaterOgVolum(dimensjonSorter(dimensjonsgenerator(1, 15), 15))

The output from the complete function that the students typed in was as follows.

```
('Dimensjoner: x=', 1, '  y=', 1, '  z=', 13, '  Overflate=', 54, '  Volum=', 13)
('Dimensjoner: x=', 1, '  y=', 2, '  z=', 12, '  Overflate=', 76, '  Volum=', 24)
('Dimensjoner: x=', 1, '  y=', 3, '  z=', 11, '  Overflate=', 94, '  Volum=', 33)
('Dimensjoner: x=', 1, '  y=', 4, '  z=', 10, '  Overflate=', 108, '  Volum=', 40)
('Dimensjoner: x=', 1, '  y=', 5, '  z=', 9, '  Overflate=', 118, '  Volum=', 45)
('Dimensjoner: x=', 1, '  y=', 6, '  z=', 8, '  Overflate=', 124, '  Volum=', 48)
('Dimensjoner: x=', 1, '  y=', 7, '  z=', 7, '  Overflate=', 126, '  Volum=', 49)
('Dimensjoner: x=', 1, '  y=', 8, '  z=', 6, '  Overflate=', 124, '  Volum=', 48)
('Dimensjoner: x=', 1, '  y=', 9, '  z=', 5, '  Overflate=', 118, '  Volum=', 45)
('Dimensjoner: x=', 1, '  y=', 10, '  z=', 4, '  Overflate=', 108, '  Volum=', 40)
('Dimensjoner: x=', 1, '  y=', 11, '  z=', 3, '  Overflate=', 94, '  Volum=', 33)
('Dimensjoner: x=', 1, '  y=', 12, '  z=', 2, '  Overflate=', 76, '  Volum=', 24)
('Dimensjoner: x=', 1, '  y=', 13, '  z=', 1, '  Overflate=', 54, '  Volum=', 13)
('Dimensjoner: x=', 2, '  y=', 1, '  z=', 12, '  Overflate=', 76, '  Volum=', 24)
('Dimensjoner: x=', 2, '  y=', 2, '  z=', 11, '  Overflate=', 96, '  Volum=', 44)
('Dimensjoner: x=', 2, '  y=', 3, '  z=', 10, '  Overflate=', 112, '  Volum=', 60)
('Dimensjoner: x=', 2, '  y=', 4, '  z=', 9, '  Overflate=', 124, '  Volum=', 72)
('Dimensjoner: x=', 2, '  y=', 5, '  z=', 8, '  Overflate=', 132, '  Volum=', 80)
('Dimensjoner: x=', 2, '  y=', 6, '  z=', 7, '  Overflate=', 136, '  Volum=', 84)
('Dimensjoner: x=', 2, '  y=', 7, '  z=', 6, '  Overflate=', 136, '  Volum=', 84)
('Dimensjoner: x=', 2, '  y=', 8, '  z=', 5, '  Overflate=', 132, '  Volum=', 80)
('Dimensjoner: x=', 2, '  y=', 9, '  z=', 4, '  Overflate=', 124, '  Volum=', 72)
('Dimensjoner: x=', 2, '  y=', 10, '  z=', 3, '  Overflate=', 112, '  Volum=', 60)
('Dimensjoner: x=', 2, '  y=', 11, '  z=', 2, '  Overflate=', 96, '  Volum=', 44)
('Dimensjoner: x=', 2, '  y=', 12, '  z=', 1, '  Overflate=', 76, '  Volum=', 24)
('Dimensjoner: x=', 3, '  y=', 1, '  z=', 11, '  Overflate=', 94, '  Volum=', 33)
('Dimensjoner: x=', 3, '  y=', 2, '  z=', 10, '  Overflate=', 112, '  Volum=', 60)
('Dimensjoner: x=', 3, '  y=', 3, '  z=', 9, '  Overflate=', 126, '  Volum=', 81)
('Dimensjoner: x=', 3, '  y=', 4, '  z=', 8, '  Overflate=', 136, '  Volum=', 96)
('Dimensjoner: x=', 3, '  y=', 5, '  z=', 7, '  Overflate=', 142, '  Volum=', 105)
('Dimensjoner: x=', 3, '  y=', 6, '  z=', 6, '  Overflate=', 144, '  Volum=', 108)
('Dimensjoner: x=', 3, '  y=', 7, '  z=', 5, '  Overflate=', 142, '  Volum=', 105)
('Dimensjoner: x=', 3, '  y=', 8, '  z=', 4, '  Overflate=', 136, '  Volum=', 96)
('Dimensjoner: x=', 3, '  y=', 9, '  z=', 3, '  Overflate=', 126, '  Volum=', 81)
('Dimensjoner: x=', 3, '  y=', 10, '  z=', 2, '  Overflate=', 112, '  Volum=', 60)
('Dimensjoner: x=', 3, '  y=', 11, '  z=', 1, '  Overflate=', 94, '  Volum=', 33)
('Dimensjoner: x=', 4, '  y=', 1, '  z=', 10, '  Overflate=', 108, '  Volum=', 40)
('Dimensjoner: x=', 4, '  y=', 2, '  z=', 9, '  Overflate=', 124, '  Volum=', 72)
('Dimensjoner: x=', 4, '  y=', 3, '  z=', 8, '  Overflate=', 136, '  Volum=', 96)
('Dimensjoner: x=', 4, '  y=', 4, '  z=', 7, '  Overflate=', 144, '  Volum=', 112)
('Dimensjoner: x=', 4, '  y=', 5, '  z=', 6, '  Overflate=', 148, '  Volum=', 120)
('Dimensjoner: x=', 4, '  y=', 6, '  z=', 5, '  Overflate=', 148, '  Volum=', 120)
('Dimensjoner: x=', 4, '  y=', 7, '  z=', 4, '  Overflate=', 144, '  Volum=', 112)
('Dimensjoner: x=', 4, '  y=', 8, '  z=', 3, '  Overflate=', 136, '  Volum=', 96)
('Dimensjoner: x=', 4, '  y=', 9, '  z=', 2, '  Overflate=', 124, '  Volum=', 72)
('Dimensjoner: x=', 4, '  y=', 10, '  z=', 1, '  Overflate=', 108, '  Volum=', 40)
('Dimensjoner: x=', 5, '  y=', 1, '  z=', 9, '  Overflate=', 118, '  Volum=', 45)
('Dimensjoner: x=', 5, '  y=', 2, '  z=', 8, '  Overflate=', 132, '  Volum=', 80)
('Dimensjoner: x=', 5, '  y=', 3, '  z=', 7, '  Overflate=', 142, '  Volum=', 105)
('Dimensjoner: x=', 5, '  y=', 4, '  z=', 6, '  Overflate=', 148, '  Volum=', 120)
('Dimensjoner: x=', 5, '  y=', 5, '  z=', 5, '  Overflate=', 150, '  Volum=', 125)
('Dimensjoner: x=', 5, '  y=', 6, '  z=', 4, '  Overflate=', 148, '  Volum=', 120)
('Dimensjoner: x=', 5, '  y=', 7, '  z=', 3, '  Overflate=', 142, '  Volum=', 105)
('Dimensjoner: x=', 5, '  y=', 8, '  z=', 2, '  Overflate=', 132, '  Volum=', 80)
('Dimensjoner: x=', 5, '  y=', 9, '  z=', 1, '  Overflate=', 118, '  Volum=', 45)
('Dimensjoner: x=', 6, '  y=', 1, '  z=', 8, '  Overflate=', 124, '  Volum=', 48)
('Dimensjoner: x=', 6, '  y=', 2, '  z=', 7, '  Overflate=', 136, '  Volum=', 84)
('Dimensjoner: x=', 6, '  y=', 3, '  z=', 6, '  Overflate=', 144, '  Volum=', 108)
('Dimensjoner: x=', 6, '  y=', 4, '  z=', 5, '  Overflate=', 148, '  Volum=', 120)
('Dimensjoner: x=', 6, '  y=', 5, '  z=', 4, '  Overflate=', 148, '  Volum=', 120)
('Dimensjoner: x=', 6, '  y=', 6, '  z=', 3, '  Overflate=', 144, '  Volum=', 108)
('Dimensjoner: x=', 6, '  y=', 7, '  z=', 2, '  Overflate=', 136, '  Volum=', 84)
('Dimensjoner: x=', 6, '  y=', 8, '  z=', 1, '  Overflate=', 124, '  Volum=', 48)
('Dimensjoner: x=', 7, '  y=', 1, '  z=', 7, '  Overflate=', 126, '  Volum=', 49)
('Dimensjoner: x=', 7, '  y=', 2, '  z=', 6, '  Overflate=', 136, '  Volum=', 84)
('Dimensjoner: x=', 7, '  y=', 3, '  z=', 5, '  Overflate=', 142, '  Volum=', 105)
('Dimensjoner: x=', 7, '  y=', 4, '  z=', 4, '  Overflate=', 144, '  Volum=', 112)
('Dimensjoner: x=', 7, '  y=', 5, '  z=', 3, '  Overflate=', 142, '  Volum=', 105)
('Dimensjoner: x=', 7, '  y=', 6, '  z=', 2, '  Overflate=', 136, '  Volum=', 84)
('Dimensjoner: x=', 7, '  y=', 7, '  z=', 1, '  Overflate=', 126, '  Volum=', 49)
('Dimensjoner: x=', 8, '  y=', 1, '  z=', 6, '  Overflate=', 124, '  Volum=', 48)
('Dimensjoner: x=', 8, '  y=', 2, '  z=', 5, '  Overflate=', 132, '  Volum=', 80)
('Dimensjoner: x=', 8, '  y=', 3, '  z=', 4, '  Overflate=', 136, '  Volum=', 96)
('Dimensjoner: x=', 8, '  y=', 4, '  z=', 3, '  Overflate=', 136, '  Volum=', 96)
('Dimensjoner: x=', 8, '  y=', 5, '  z=', 2, '  Overflate=', 132, '  Volum=', 80)
('Dimensjoner: x=', 8, '  y=', 6, '  z=', 1, '  Overflate=', 124, '  Volum=', 48)
('Dimensjoner: x=', 9, '  y=', 1, '  z=', 5, '  Overflate=', 118, '  Volum=', 45)
('Dimensjoner: x=', 9, '  y=', 2, '  z=', 4, '  Overflate=', 124, '  Volum=', 72)
('Dimensjoner: x=', 9, '  y=', 3, '  z=', 3, '  Overflate=', 126, '  Volum=', 81)
('Dimensjoner: x=', 9, '  y=', 4, '  z=', 2, '  Overflate=', 124, '  Volum=', 72)
('Dimensjoner: x=', 9, '  y=', 5, '  z=', 1, '  Overflate=', 118, '  Volum=', 45)
('Dimensjoner: x=', 10, '  y=', 1, '  z=', 4, '  Overflate=', 108, '  Volum=', 40)
('Dimensjoner: x=', 10, '  y=', 2, '  z=', 3, '  Overflate=', 112, '  Volum=', 60)
('Dimensjoner: x=', 10, '  y=', 3, '  z=', 2, '  Overflate=', 112, '  Volum=', 60)
('Dimensjoner: x=', 10, '  y=', 4, '  z=', 1, '  Overflate=', 108, '  Volum=', 40)
('Dimensjoner: x=', 11, '  y=', 1, '  z=', 3, '  Overflate=', 94, '  Volum=', 33)
('Dimensjoner: x=', 11, '  y=', 2, '  z=', 2, '  Overflate=', 96, '  Volum=', 44)
('Dimensjoner: x=', 11, '  y=', 3, '  z=', 1, '  Overflate=', 94, '  Volum=', 33)
('Dimensjoner: x=', 12, '  y=', 1, '  z=', 2, '  Overflate=', 76, '  Volum=', 24)
('Dimensjoner: x=', 12, '  y=', 2, '  z=', 1, '  Overflate=', 76, '  Volum=', 24)
('Dimensjoner: x=', 13, '  y=', 1, '  z=', 1, '  Overflate=', 54, '  Volum=', 13)
```

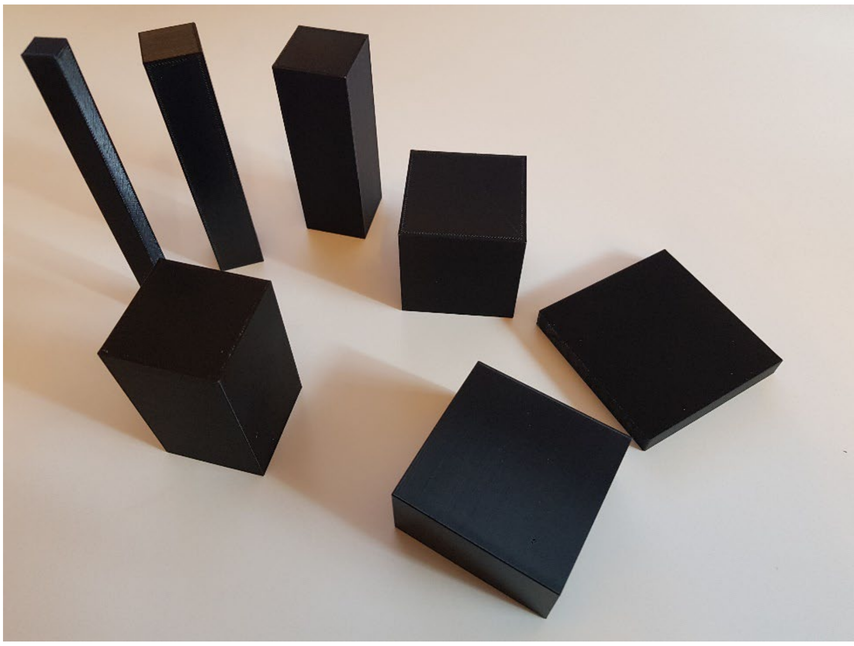
This is of course a long output, and we talked a little bit about that before we moved on to the next part. From here we skipped to the last task on the problem sheet, number 3c. This task adds another constriction to the dimensions, there is still the constant sum of the dimensions, but the x and y needs to be equal to each other as well. To accomplish this a different sorter is needed, this sorter is dubbed oppgave3cSorter, translation being task3cSorter.

oppgave3cSorter(listeMedDimensjoner, sumKonstant)

What this does is the same as the sorter we used earlier, but it also removes all the possibilities where x and y is not equal. The function takes the same arguments as the prior sorter, so “oppgave3cSorter” can just be substituted for “dimensjonSorter” in the textbox. Below is the output the students got.

```
('Dimensjoner: x=', 1, '  y=', 1, '   z=', 13, '  Overflate=', 54, '    Volum=', 13)
('Dimensjoner: x=', 2, '  y=', 2, '   z=', 11, '  Overflate=', 96, '    Volum=', 44)
('Dimensjoner: x=', 3, '  y=', 3, '   z=', 9, '  Overflate=', 126, '    Volum=', 81)
('Dimensjoner: x=', 4, '  y=', 4, '   z=', 7, '  Overflate=', 144, '    Volum=', 112)
('Dimensjoner: x=', 5, '  y=', 5, '   z=', 5, '  Overflate=', 150, '    Volum=', 125)
('Dimensjoner: x=', 6, '  y=', 6, '   z=', 3, '  Overflate=', 144, '    Volum=', 108)
('Dimensjoner: x=', 7, '  y=', 7, '   z=', 1, '  Overflate=', 126, '    Volum=', 49)
```

Now the output is more easily comprehensible, and reason with. At this point the 3D models in the picture below was brought out, and we went through the lines of output from the program and which of the output corresponding to which 3D model. This was the last thing that got done in the arrangement.



3D models corresponding to the output of oppgave3cSorter

The interviews

The next day after the arrangement it was time for the interviews. As previously mentioned not all of the participants was interviewed, but in this section, we are first going to look at the recording and setup of the interviews, a brief overview of the participants, and then the questions we got through and the reasoning behind them.

Recording and setup

The interviews were held in a classroom that was free at the time. The interview was recorded with the same audio setup as the arrangement, a single large diaphragm microphone and studio recorder. The interviews were not filmed. A desk was set up so that the interviewer sat opposite of the interviewee. This made it possible to have a computer with the programs on it so that we could run them and talk about them during the interview. The 3D models of the boxes from the last task on the problem sheet was also on the desk during the interview, and these were used to make connections between the programs output and the geometric shapes.

Participants

As previously mentioned, not all of the participants from the arrangement were interviewed on account of the schoolwide happening. Only three of the six participants were interviewed. The first participant that was interviewed was Alf, the second participant was Christine, and the last participant was Dan. The anonymization of the participants happened during the transcription of the interviews, and the participants were given names according to the alphabetical order. The reason because the names skip a letter after Alf before Christine is because Alf mentioned Bjorn who was not interviewed during the interview.

The participants who were available for interviews were in class in the room next to the classroom the interviews were held in, so there was not too much delay in between the interviews of the different participants. Each interview took around 20 minutes to complete, with the last one being cut a little short because we were out of time.

The last participant Dan that was interviewed had a little bit of programming experience, both with programming classes at school, and an interest in programming on his spare time. He also saw many use cases for programming both in school and in society in general. The first participant Alf had a little bit of programming experience on account of a few classes of programming, and he also said during the interviews that he had observed Bjorn who did do a little bit of programming. He also saw potential for programming being used both in school, and also in construction as replacement for humans doing dangerous work. The second participant to be interviewed Christine had little programming experience, and also said that she was not very interested in programming in general. She could see that programming could relate to mathematics in some ways but expressed that she had not a lot of interest for programming in mathematics either.

The questions and flow of the interviews

The interview guide was made before the arrangement was done, and before the realization that it would not be possible to get through all of the tasks. Therefore, the interview guide contains questions that would be aimed at all of the different subtasks from the arrangement. Just as we skipped quite a lot of the tasks in the arrangement, we also had to skip quite a lot of the questions on the interview guide. Not all of the questions got asked in the exact order of the interview guide, but generally the flow of the interview followed the guide.

The interview and interview guide has three different stages or sections. The first is a small warm up just to get the interview started and the participant comfortable for the interview. This part asks about what the participant thought of the arrangement, the tasks, and using Python in this arrangement. There is not much more to say about this section, but the next sections will be elaborated further. The next section is the background section. This is where we try to establish if the student has any prior experience with programming and what kind of programming, and in what context. The third and last section is about the tasks. Here we try to establish what the student got out of the arrangement and how the student reasons with the geometric concepts and the programs.

The background section of the interview

One of the first questions in the background section that was asked was “Do you have any prior experience with programming?”, this was a generalized question that could uncover if the student had done programming in class or on his or her spare time. The follow up questions to this were attempted to relate to the student’s response, but on the interview guide there was some suggestions for follow up questions that were used. “Did you have programming in an elective course?” and “Have you used programming in math class?”.

Another question that was asked was “Do you have any prior experience with Python?”. Follow up questions was “In what sense?” and if the students did not have any prior experience “Have you heard about it? What have you heard about it?”

Another question related to prior experiences with programming was “Do you have any prior experience or heard of other kinds of programming languages?”. Part of this question got sometimes answered with the first question in the background section. Follow up to this question was naming of a few programming languages that the student might have heard or used, for example: “Scratch? Makecode for microbit? Logo? HTML? Javascript?”. And then the focus was to try to figure out what the student had used this programming language for, or where and how they had heard of it. And what they feel about these languages, especially in relation to Python.

The next part of the background section of the interviews focused on the student’s attitude towards programming. The first question focused on what kind of image or perception the student had of programming. “How would you describe programming? In general? Computer programs, or games? Programmable things? Society? The world?”, the first being the main question, with the follow up questions straight after.

The next question focused on the student’s attitude towards learning programming in general. “How would you describe learning programming in general? Useful? Not useful? Use cases? Interesting? Not interesting?”. These questions are very much tied into the prior questions before it.

The last question in the background section of the interview was about the student’s attitude towards learning programming in mathematics class in school. “How will you describe learning programming in and used in mathematics class in school? Useful? Not Useful? Use cases? Interesting? Not interesting? Positive or negative aspects?”. The goal of the question was to make the student reflect on how they see programming being used in mathematics class.

A large part of the background section of the interview is an extension of the warmup of the interview, to get the student comfortable with the situation and the microphone setup. But also, the student’s prior experience, motivation and attitude towards programming might be relevant towards what can be expected of an outcome of a programming-based arrangement, so it is interesting insight none the less.

The task section of the interview

In this part of the interview the focus is on the tasks from the problem sheet. This part of the interview coincides little with the interview guide, because of the interview guide follows the original plan for the arrangement, but a lot of the tasks was simply not done. The tasks were not performed either as originally intended, the plan for the tasks was for the students work more on their own, but mostly the arrangement was guided. A lot of this part of the interview is a conversation about the geometric significance in relation to the programs output, and fewer clear questions. It is about reflecting on the programs output, the abstraction of the calculations being done in the program, and do a reasoning that is made probable or certain by this output.

One of the first questions asked in relation to the tasks was but quickly moved on from "With the limitation that the sum of the dimensions had to be 15. What happens with the other sides if we change one of the dimensions?". After this question we move on to the last task on the problem sheet, the one that prints out dimensions, surface area, and the volume.

In the last task there was an extra constriction to the boxes. That not only the sum of the sides needed to be 15, but also x, and y needed to be equal. One of the questions in relation to this was "If say, instead of x and y, that the z and y dimensions need to be equal, what would happen? Would it look different?". This question tries to find how the student think about the sides of the box in relation to the program's output.

Another two related questions and also the last questions that was asked was if the student could see a way for the boxes to get even bigger, or even smaller under the constrictions that the boxes had. "Could there exist a combination of dimensions that can make a bigger box than 5 times 5 times 5?". And "Could there exist a combination of dimensions that could make a smaller box than 1 times 1 times 13?". The idea behind these questions is that by looking at the printed list of dimensions, surface area, and volume, it should be possible to make probable if such boxes exist. If looking at the output from the program for the 5 times 5 times 5 box, and the box above and underneath it on the list, it might be possible to argue that this is the best combination of dimensions for the largest box under the set constrictions. And looking at the smallest box at the top of the list, 1 times 1 times 1, it might be possible to argue that if we had a box that had x and y dimensions smaller than 1 then it could become even smaller. These last two questions are where the most part of the reasoning comes in, if whether the student can reason on the programs output and the geometric idea of the sizes of the boxes. The fact that these questions and tasks have very defined rules the reasoning that is most likely to be elicited might be deductive reasoning, although the nature of the output of the program allows for some inductive reasoning on generating of rules in the reasoning, specifically the list of dimensional values representing multiple cases and results.

Data and analysis

The semi structured interviews

The transcripts are in the original language and in full in the appendix, but the parts that are presented here is translated to English. However, the numbering on the segments of the transcripts is kept for ease of transparency. Below the data from the interviews are presented and analyzed in sections based on a specific task or question that the student is asked to reflect on. In these sections the students' reasoning is analyzed based on the student's responses to the questions and the premises the students have available for doing their reasoning. During these sections both deductive reasoning and/or inductive reasoning is analyzed, depending on how it is appropriate. In most sections the most prominent is the deductive reasoning, but in some sections, there are evidence for inductive reasoning happening. How the student reasons are sometimes divided into multiple possible reasonings, in the sense that we can obviously not be completely sure how the student think. There are also sections where the reasonings are analyzed in the form of successive reasonings, that either the reasoning change when reflecting on the task, or some reasoning leads to a different reasoning entirely.

What happens when we change one of the dimensions?

Christine 1

In this part we can see the student Christine and the interviewer talking about what happens when one of the dimensions get changed under the constriction of the sum of the dimensions having to be 15. The question is first asked poorly, and the constriction does not get mentioned at first. Without the constriction in mind the student points out that nothing will happen if you change one of the dimensions. When the constriction is mentioned the student figures that if one of the dimensions change then the other ones have to change as well to make the sum of the dimensions be 15.

173 RE: Ehm, so the first thing we did yesterday, or the first thing we did was to, ehm. We just used a print function, and an easy calculation of surface area with a function. But then we got Python to print out a long, long list. With a bunch of dimensions, and surface areas, and volume. And then of course it was a lot of numbers. So then, eh, everything just got listed, so it starts here with all the x values, all like, x values when they are 1, and y and z get adjusted in a way. And we also get all the, ehm, the surface areas and volumes tied to the dimensions of those boxes. So, what I am thinking about is.
What happens with the height when we change the other dimensions? Like for example this figure here (puts the 3D models on the table) then, so we have an x here, and then an y, and a z in the height. So, if we then, ehm, if the height increases what happens then with x and y? What happens with the other dimensions?
174 Christine: Eh, nothing happens, does it? Or?
175 RE: Ehm because, we put in a dim, we put in a constriction in the code. That did so that, eh, the sum of x, and y, so x plus y, plus z, the height, so all of those three dimensions had to be 15 added together. So, if then, ehm, for example then if eh, x is 1 and y is 1, then z becomes 13. But if these other one's changes what happens then with the height?
176 Christine: Well, then it would have to either get higher or lower to make 15?

So, if we first analyze the reasoning in the first part of this section of the interview, the part where the student thinks that nothing happens if one of the dimensions gets changed. The things we can observe in the transcript is the case and the result. The case (a situation or a parameter) is that one of the dimensions changes, and the result (what happens when the case is what it is) is that nothing happens. Since nothing, at least not anything significant since the student does not mention it, happens when you change one of the dimensions, there is no rule to make the change.

However, when the student was reminded that there was a constriction of the sums of the dimensions, the student applied a rule in her reasoning, and she reasoned a result of changing the dimension. The rule being applied is that if one of the dimensions changes, the other dimensions need to be changed so that the total sum of them becomes 15. Now the student has reasoned a different logic. The case is still; one of the dimensions changes. The rule that was formed is; if one of the dimensions changes, then the others need to change so that the sum of the dimensions are 15. This changes the result; the other dimensions must change to make a sum of 15. This is an example of affirming the antecedent. The antecedent (what makes the consequent true) being that one of the dimensions changes, and this result in the consequent (is true if the antecedent is true) that the other dimensions need to change to make the sum of 15. An interesting observation here is that at first there was no rule or constriction provided through the task or question, in the sense that there was no problem needing solving. There was a mismatch between the expected answer and the answer given because the whole task was not communicated at first, and therefore there was not a reasoning done on the problem.

Dan 1

In this part the student Dan and the interviewer is talking about what would happen if one of the dimensions changes. Dan argues that it is possible to just put on more, you can just make it longer or wider for example. It seems that Dan has not completely understood the idea of constricting the sum of the dimensions here. It is unclear if this is because of lack of understanding of the task, or if he means that this is not a limit to how the program could do the calculations.

278 RE: Ehm. So, ehm, when we had, eh, these here. Like this, what can you say about if we, so we had a constriction, and that constriction was that the total sum of them had to be 15? Of x and y and z, that is like the length, the width, and height.

280 Dan: Even though we could put on more, could make it higher.

281 RE: Yes. If we had put 20 for example as the constriction, then we could have gotten even bigger. Eh, but if we increase the length for example, what has to happen with the width and height then?

282 Dan: Eh, if you increase the length, then it is still, eh, length, like the height and so the same but it will just get longer.

In this part the student Dan is clearly not accepting the constriction of the sums. On line 280 he argues that it is possible to go above the total sum of 15, and on line 282 he suggests if the length increases the height and width can just stay the same, and the box will then only get longer. In this case we could say that there is no rule that constricts the sum, as it was in the beginning when Christine was asked about the same thing. There is no reasoning on whether or not the box is valid with a specific set of values for the dimensions, meaning with how the program is coded for the constrictions on the dimensions. He seems to be more focused on the possibility of changing the code, instead of working within the constrictions of the task and the prewritten code. So, it could be a different reasoning entirely, it could be tied to the idea of the possibility of changing the program, or it could just be a misunderstanding of the sum being a result of the values for x, y, and z instead of a constriction. This would create the following two different scenarios.

In the case of changing the program idea, if the rule is “the sum of the dimensions x, y, and z must be 15 for it to be a valid box”, the antecedent is “x + y + z = 15”, and the consequent is “valid box”. If the case is a set of dimensions that ends up having a larger sum than 15, then the antecedent must be changed so that the consequent remains true. If the dimensions are 4, 6, 8, then the antecedent must be changed to “x + y + z = 18” for the result to be a valid box. This is closely related to the fallacy of affirming the consequent, or at least “conserving the consequent”, although it might be closer to a more inductive approach to the problem. In inductive reasoning rules are concluded based on multiple cases and results. If hypothetically the rule was not set this would allow for changing the program, and also ultimately the task. If it is desired to increase the length of one of the dimensions, the others need not be decreased, the program can be set to accommodate this. The rule is then concluded based on the desired result (valid box) and case (dimensions).

If on the other hand the rule being reasoned is as follows, “x + y + z, equals the sum of the dimensions of the box”, this would mean that Dan has simply misinterpreted, or not understood the task of what happens when one of the dimensions change when the constrictions are in place. The rule being reasoned is not tied to the question being asked, the reasoning is based on a false premise. Instead of thinking the sum of the dimensions as a constriction it is just a number that follows the adding together the values for the dimensions. On line 280 after the constriction was mentioned, Dan says that “… we could put on more, could make it higher.”, this seems to suggest that Dan is not concerned with the sum as a constriction, but rather it is just a trivial calculation after generating some dimensions for the box. This could mean that the idea of the question and task was not understood, and he therefore was more concerned about the general expandability of the box.

What if we changed which two of the three dimensions that have to be equal?

Alf 1

In this part we can see from the first interview the student Alf and the interviewer talking about what happens if not x and y is constricted to be equal but two different dimensions are constricted to be equal. First the student suggests that when some different dimensions have to be equal, one of the dimensions that was replaced could be as long as we want, forgetting the total sum constriction. The student eventually ends up turning one of the boxes in another orientation, and then ending up with a different box with different dimensions being equal and still keeping the total sum constriction.

71 **RE:** Mhm, yes for this was the one that sorted a little bit. And it added an extra constriction as well. If we did.

72 **Alf:** That x and y should be equal.

73 **RE:** Exactly, just that. And then it so that as long as those two are equal. Then, and then there is also the sum, that it should not be more than 15, or it must exactly be 15.
RE: But let's say that if the height and width. For example, that y and z should be equal, what would happen? Would it look different?

74 **Alf:** I would think so.

75 **RE:** mhm

76 **Alf:** Considering that the volume would follow along.

77 **RE:** Mhm. So, if we look at these figures here then. This is the one that is, mm, this is the one that is 6 times 6, I think. And 3 in height. (Shows the 3D-printed figures)
RE: But then we are 6 that way, 6 that way, and so they have to be equal, and the height is 3. But if I say that that one, shall we say, that x there for example, and the height should be equal.

78 **Alf:** Yes.

79 **RE:** What would happen then, would we have gotten the same numbers, or something different?

80 **Alf:** no, I think everything would have, the numbers would have changed considering that that one could be as long as it would then.

81 **RE:** mhm

82 **Alf:** Or it could suddenly, like, be a very long flat one.

83 **RE:** mhm

84 **Alf:** For example, if you put it upwards like that.

85 **RE:** Yes, like this one for example.
86 **Alf:** Yes, like that.
87 **RE:** Eeh, so this would then be, then this would be, yes how would this be now. Then.
88 **Alf:** Isn't that 7, 1, 7, isn't that, right?
89 **RE:** Mhm, yes, that sounds right, 7, 1, and then 7 yes. Then this one, so, then the x and the z would be equal.

In the first part the rule is "the x and y dimensions need to be equal, and the sum of x, y, and z must be 15", the result is whether or not it is a valid box for the task. The consequent a valid box for the task is dependent on the antecedent being true. Afterwards the rule gets changed into "the x and z dimensions need to be equal, and the sum of x, y, and z must be 15". These can be considered as nested syllogisms, or one syllogism with one complex antecedent. If we look at how Alf suggests the box "… could suddenly, like, be a very long flat one." on line 82, might suggest that the reasoning in play here are two nested syllogisms. Although it might have developed into a single syllogism with the complex antecedent, as shown on line 84 and onwards where Alf is describing reorienting the 3D-models, but the reasoning could also just be more processed.

What is meant by a nested syllogism vs a singular syllogism is related to how the reasoning is processed. First, the nested syllogism. The rule "the dimensions x and z must be equal for it to be a valid box", if the case is x = 1 and z = 1, antecedent being "x = z" evaluates to true, which will result in the consequent "valid box". Then there comes a new syllogism after this. The rule being "the sum of the dimensions must be 15 to be a valid box", antecedent being "x + y + z = 15", and consequent being "valid box". Whether or not the result is a valid box is based on two reasonings in succession.

The complex syllogism. This is a syllogism with two antecedents that has to be evaluated to true for it to lead to the consequent of the valid box. The rule here would be "the x and z dimensions need to be equal, and the sum of the dimensions x, y and z must be 15, for it to be a valid box." First the "x = z" antecedent must be processed, next that "the sum of x, y, and z must be 15" antecedent must be processed. The result of this is the consequent "valid box" if the two antecedents are true. Whether or not the result is a valid box is based on one reasoning with a more complex rule set.

When Alf suggest that the box could become a very long flat one, because the y dimension can get as long as it wants, it would mean the box is valid as long as the first constriction is applied. The constriction of the sum is being disregarded or forgotten. This is like jumping the gun after the first reasoning if it is a nested syllogism, or it has not developed into the complex syllogism. Towards the end Alf suggests that one of the 3D-models could be reoriented, so that the dimensions are pointing in a direction that makes x and z equal. The 3D-models are conveniently staying within the constriction of the sum whichever way it is pointing, even if the only constriction that is being reasoned is that x and z is equal. This means that it is unclear if or not Alf is thinking about the sums, and therefore whether or not he does reason both of the nested syllogisms, or a complex syllogism on the matter.

Could there exist a larger box?

Alf 2

Here we can see from the first interview that Alf and the interviewer is talking about if it is possible to make a larger box than the 5 cubed one. The student suggests that it should be possible to get a larger box by taking some of the length off of z and putting it on the x dimension.

The question asked in this section was of both if it is possible to find larger or smaller boxes than the ones we had found, but the student first talked about making a larger box than the 5 cubed, and the interviewer asked specifically about the possibility of a smaller box afterwards.

93 RE: Ehm, and, based on the ones we have here now then. Can we be sure that we have found the combinations of dimensions that in a way actually gives the largest and the smallest? Could there be some numbers here that we don’t see in a way?
94 Alf: Eh, I am not sure.
95 RE: Like, say for example if, if we could have used decimal numbers for example, would it be possible for us to get some other numbers for example?
96 Alf: Yes.
97 RE: Could we get some bigger surface areas and, or larger volume, less volume?
98 Alf: Yeah, I think you could do that.
99 RE: Mhm. Like for example with, like that one which we found out was the largest

was it not?
100 Alf: Yeah, 5 times 5 times 5.
101 RE: That was 5 times 5 times 5. Could we, if we had decimal numbers instead, could we get this bigger in a way? If we are still supposed to, we still have the constriction of that the sum have to be 15.
102 Alf: Ehm. I think that if you put z down to for example 4,3, and then x be 5,7. Then I think it would be bigger.
103 RE: Eeh. Yeah.
104 Alf: Yeah.

On line 102 Alf suggests that if z is reduced to 4,3 and this reduction moved over on to the x dimension, making the x dimension 5,7, the y dimension is kept at 5. Alf then think this would be bigger than the original box of 5 times 5 times 5. It is unclear whether or not this is a reasoned argument based on the understanding of the problem and data, or if it was a guess reasoned based on; since the question was asked, it must have an answer.

In this part the rule that is used in the reasoning could be something like "If the dimensions are changed into a decimal value and optimized, the box will be as big as possible.". The case here is "5 times 5 times 5, is changed into 5,7 times 5 times 4,3", the resolution of decimals is higher than integers and it might make sense that this will get it closer to an optimal size. The x dimension is increased so it will get wider at least, although at the expense of height in the z dimension. With the increased size of x, and y kept at the same value, the base surface area of the box is increased, making it bigger in that regard. The result of this reasoning is that the box is bigger, in a way. This might then be a reasonable logic, but it will still be detached from the original idea in the task; that the size of the box is determined by volume and surface area. The calculation and what size means is missing from the logic. In the rule mentioned above it is only necessary to increase the resolution and "optimize" the values for the box to get bigger, but there seems to be no connection with what makes the size of the box. For example, volume; "x, y, and z multiplied, gives the volume, and given optimized values the box will be as big as possible.".

If on the other hand it was reasoned based on; there must be an answer to this question, this is rather a fallacy of affirming the consequent. The antecedent in this reasoning is “optimized values”, and the consequent is “the biggest box” or at least “bigger box”. Since there must be a “biggest box”, or at least a “bigger box”, then there must be some values for x, y, and z, that increases the size of the box. Since the idea of the problem is not going from making the optimal dimensions of the box based on a total sum, it is merely made probable whether or not there is a bigger box based on the printout of the program. These outputs show that if the dimensions deviate from 5 cubed the size of the box is reduced both in surface area and in volume. Afterwards when Alf is asked if there is a smaller box, more emphasis is made on this printed list. But, in this part of the problem on this level there is no way of generating explicitly the optimal dimensions for the box, any values for x, y, and z based on the consequent (“bigger box”) will be a fallacy of affirming the consequent.

Christine 2

Here we can see the student Christine being asked if she think it is possible to make a bigger box that the one that is 5 times 5 times 5 while still keeping in mind the constriction of the sum being 15. She asks what is meant by the question, if it means a dimension other than x, y, and z. This gets clarified that it means other numbers for the three dimensions, and she says she can’t think of any way of making the box larger.

177	RE: Yes, absolutely, absolutely. So, ehm, let’s see. Ehm. So, if vi then look at the end of, eh, the last part of the task, the last script that we ran, then it became a bit clearer, because it sorted a bit more. Let’s see, eh, there. So, it is those three, or these seven here. So, these are all the ones that I have here then. So, these are the different dimension, and these dimensions are really like this. (Showing the 3D-models). And then we have like the first one here, then we have x is equal to 1, and y is equal to 1, and then z is 13, so that is this one here, 1 times 1 times 13. And that code there had an extra constriction, the constriction was that x and y hade to be equal, and the total sum of all three had to still be 15. So, that is an extra constriction. So, when we then have, let’s see, put in all of these constrictions here, we get all the dimensions that comply with this. What I am wondering is, how sure can we be that we. Or I can ask something else first.

Ehm, which one was it that was the smallest and which one was the largest, can you remember?
178 Christine: What was smallest or largest?
179 RE: So, which one of these models was it that had the least surface area and volume, and which one of them had the largest surface area and volume?
180 Christine: I can't remember completely … Where they not all the same, if they where 15 in total?
181 RE: If we look at the code here. So, this here is then, these are the dimension, so, 1 times 1 times 13, then 2 times 2 times 11, 3 times 3 times 9, and then we get the different surface areas. So, 1 times 1 times 13 has 54 and volume is 13. If we then look at that list. Which one of these are that largest of them?
182 Christine: Ehm. I'm thinking one of those two perhaps.
183 RE: Mhm. So, this is one of them, these are one of the two of the largest. So, which one of these is it then?
184 Christine: I am really unsure, hehe.
185 RE: If we look at this one here then, like this, so this one has 1 times 1, these are then in a way the base surface. So, the base surface, that is x and y. This is 1 times 1, and then it is 13 in height which is z. So, that is the first of these here. So, which one of, is it, which one of these 3D-models here is then the number 2? 2 times 2 times 11, do you think?
186 Christine: Ehm, this one.
187 RE: Mhm, yes. So, this one here, for this is 1, then this one is 2 wide, and 2 wide that way, so 2 times 2, and then it is 11 high, so 2 less than 13. Then we have, so the last one there then, the 1 times 1 times 13, that one has surface area 54 and volume 13. Surface area is in square centimeters, and then we have volume in cubic centimeters, which is centimeters to the power of three. So, then we have the x equals 2, and y equals 2, and z is equal to 11, that has a surface area of 96 and volume 44. We can then see that the volume has increased a bit, and the surface area has increased a bit. So which one of these are then 3 times 3 times 9?
189 Christine: Thinking … This one was 11 high, was it not?
190 RE: That one was 11 high yes.
191 Christine: So, it is that one.
192 RE: mhm, yes. So, then it is, so that one is 2 wide, and that one is 1 wide, then you know okay, then it is, that one is 3 wide and then it is most likely 9 high then.
193 Christine: Mhm.
194 RE: Then we have the next one. So, then it is 4 times 4. What is times 7?
195 Christine: Ehm, that one. (points at the 3D model which is 4 times 4 times 7)
196 RE: Yes, that sound right. That one is then, that is 3 pluss that one there, then yes, that's right. So that is 4 times 4 times 7 (showing with the 3D-models). Ehm, then we have, like from that one, eh, 3 times 3 times 9, which was that one there. That had a surface area of 126 and a volume of 81. And then, the 4 times 4 times 7, had a surface area of 14 and a volume of 112. As we can see both the surface area is rising still, and the surface area and the volume is rising still. So, then we have 5 times 5 times 5. Which one is that?
197 Christine: (Points at the right 3D-model, which is a cube)
198 RE: Yes! So, 5 times 5 times 5, and then we have a surface area of 150 and a volume of 125. And 6 times 6 times 3?
199 Christine: (Points at the right 3D-model)

200 RE: yes! So, then we have, this is 6 that way, 6 that way, and 3 high. And then the surface area is 144, and the volume 108. Now the surface area and the volume is starting to go a bit down, it is decreasing a bit. And then, we have x equals 7, and y equal 7, and z equals 1. Which one must that be?

201 Christine: Eh, wait, what was that it was 7 times 7 times 1?

202 RE: Yes.

203 Christine: Haven't we done all except for that one now? (Points at the last 3D-model)

204 RE: Yes, that is probably right. Well.

205 Christine: Yes (... unclear)

206 RE: Exactly right that. This one must then be 7 that way, 7 that way, and 1 in height.

207 Christine: Yes.

208 RE: That one has a surface area of 126 and a volume of 49. Then it is really starting to, eh, both the surface area and the volume to decrease a bit. But it is still not all the way down to, what we had when we had 1 times 1 times 13, which is that one (points at the 3D-model), then we had a really low surface area and volume. But if we look at, eh, the 5 times 5 times 5. That was that one (points at the 3D-model). That is the one that shows to have the highest number, the highest surface area, and the highest volume. Ehm, if we then look at these dimensions. In relations to that one, 4 times 4 times 7, the one that is over (points at the program), that is that one (puts the 3D-model forth). And then we have the other one which is 6 times 6 times 3, which is this one (puts forth the 3D-model). These are the three here. Then we have that one which got the largest volume and the largest surface area. And then here we have a bit less, and there we have a bit less volume and surface area (points at the 3D-models). But how sure can we be that it is actually the biggest? Does there exist, is it possible to find, or could there be a type of dimension that could have made it so that it would have gotten even bigger?

209 Christine: Dimension? With like volume and that?

210 RE: Yes, so, the dimensions are then x, y, and z, so it is the width, length, and height. So, does it exist a dimension that could make it even bigger than what it is?

211 Christine: So, something different than height, width, and breadth in a way? No, length?

212 RE: Eh, yes, or something different than 5 times 5 times 5.

213 Christine: (Oh, all the numbers? ... a bit unclear)

214 RE: As long as vi have the constriction that the sum needs to be 15.

215 Christine: Ehm, but we have all the variations here, don't we?

216 RE: Ehm, yeah, in a way we could say that we do. But could there exist, could it be a possibility to make, or change those numbers 5 times 5 times 5, so that is 5 5 5, so 5 plus 5 plus 5 which is 15, so that's the constriction we have there. And the volume is 5 times 5 times 5. Is it possible that we could change the numbers, made one larger and make another smaller to get an even larger model or box? With a larger volume, and larger surface area?

217 Christine: Ehm. Not that I can think of at least.

Before the student answers the question, all the boxes from the Python-printout are pointed out and recognized. It is also pointed out how the surface area and the volume changes when the dimensions change. Also, the trend that the largest sizes converge at the regular polyhedron, with the values for the dimensions are equal, are laid out. These are all factors that contribute to the rule to which the student reason with when answering the question at the end.

The rule that these factors contribute to is “x+y+z=15, x=y, and values for x, y, z that creates a box that is larger than 5, 5, 5”. We could think of it as having one antecedent with multiple factors, or multiple antecedents needing to be true for the consequent to be true. The antecedent in this rule is “x+y+z=15”, “x=y”, and to which the values for x, y, and z creating a box. The consequent being that “the box created by the values for x, y, and z results in a bigger box than 5, 5, 5”. During the interview multiple cases get evaluated, for example 4, 4, 7, and 6, 6, 3, the generated boxes over and under 5, 5, 5, both of them resulting in being smaller than the biggest box. It is not unreasonable to think that the student evaluates the trend of the reduction in size going away from 5, 5, 5, and it is unclear if she reasons about the volume by multiplying the three values. However, she does conclude the result that there is no box that fill these requirements and becomes bigger than the 5, 5, 5, or as she puts it “… Not that I can think of at least.”.

This is an example of denying the consequent, in which since she can’t think of a case that result in a box that gets bigger than the biggest box with the set constrictions, then there are no values that fulfill this requirement. This is a valid logic.

This in turn can also be looked at in the form of inductive reasoning. Because multiple cases resulted in boxes that are smaller than the biggest box, there are multiple cases and results present that can be used to generate a new rule. This might be the one that is present at the end of the reasoning on the biggest box and could also have explanation value to the response that she can’t think of values creating a larger box. The rule generated by the reasoning might be “a box with equal values for x, y, and z, creates the box with the largest possible surface area and volume”. Since the trend of box-sizes points to the middle ground being the biggest, it might seem intuitive that all equal values for x, y, and z, gets biggest. But since not all possible values has been tested, nor has it been proven, it has only been shown probable, the student can not be absolutely certain that this is the biggest possible box. Although, this is the nature of inductive reasoning, it is not certain, only probable. But it could seem that because the student does express an uncertainty concerning whether or not there exist a possible set of values that result in a bigger box, she reasons this uncertainty. This uncertainty is interesting to keep in mind in the part of the analysis "Christine 3" where she reasons about the possibility of a smaller box than 1, 1, 13, where she seems to be much more certain in her response to the task.

Dan 2

In this part Dan and the interviewer is talking about making a bigger box than the 5 times 5 times 5. As before Dan disregards the constriction of the sum of the dimensions having to be 15. He suggests making the box higher without changing the other dimensions, and later on makes a combination of two of the generated boxes with the sum being greater than 15.

299 RE: Aha, yes, cool. So, eh. So, we have then found out that, that we have the biggest, that was 5 times 5 times 5. Then the surface area was 150 and the volume was 125. So, how certain can we be that this is in fact the biggest within the constrictions that we have?

300 Dan: Ehm, because those three numbers are 15 in total.

301 RE: Mhm. So they are in line with the constriction. But could we change the dimensions in a way. For example, eh could we have. If we had increased x a little bit, and decreased the others a little bit. Would we then, with a combination of that, could we have gotten, eh …

302 Dan: We could of course have gotten something else then. If we had increased a little bit on one of them, and, yeah, left the others unchanged.

303 RE: Mhm, but could we have gotten it even bigger than what we have got already?

304 Dan: Ehm, that depends really, on what number that you change.

305 RE: Mhm. For if we are going to have the constriction that x and y should be equal. Eh, if we then had for example said, eh, 4,5 times 4,5, plus, eh, or times 6 then. Could it have gotten bigger or would it get smaller? If we compare it with for example the, eh, the, eh, 4 times 4 times 7, eh, and, if we then have, let's see. (finding the 3D-models that corresponds to the boxes that is over and under the printed list). We then have, this is the one that is biggest, and then we have, no which one was that was, that's the one that was directly below …

306 Dan: (paying attention) … Yes …

307 RE: … and that was directly above, was it not. (The three 3D-models is now in front on

the table). So, if we then, eh, so that one there is then, eh, 6 times 6 times 3, so that is that one (showing the connection between the models and the printed list on the computer). And that one is 4 times 4 times 7, so 7 high and 4 in width. So, if we then had if we had changed these dimensions. Is there a middle ground between 5 times 5 times 5, and 4 times 4 times 7? Or between the, eh, 6 times 6 times 3, and 5 times 5 times 5? Is there something in between that could have become bigger?

308 Dan: Ehm, if you combine some of them, then it could be something in between. If for example you take 5 times 5 times 7, then it is a middle-, then it is a mix of both.

309 RE: But could they have become, could it have become even bigger?

310 Dan: It would get taller. Eh, maybe it would have a little bit more volume as well. If it had, since if it would have gotten a little bit higher, then I think, it is two higher so it as high as that one. (Show with the 3D-printed models) And, then it will get the volume it is supposed to have.

In the first part here, the student says that we know it is the biggest box because the sum of the dimensions is 15. It is a bit unclear whether or not he keeps the constriction of x=y in mind, but he does not suggest a combination of values that would indicate that he disregards it, so for the sake of argument we can say he keeps to this. The rule in this first part would then be “x = y, and x + y + z = 15, makes the biggest box”. The antecedent being x and y being equal and the sum of the dimensions. The consequent being the biggest box. It does not seem that he is concerned about the significance of the x, y, and z values being equal. The case is 5, 5, 5 and since these values stay within the set of rules, this results in being the biggest box possible. This is an affirmation of the antecedent, which is a valid logic, although the rule has its weaknesses.

However, the rule present could be a result of inductive reasoning of multiple cases, whereas these cases are the other neighboring boxes with different values for the dimensions, and the results being less surface area and volume. This might also mean that there is another part to the rule which is not clearly communicated, “since the combination of values other than 5, 5, 5 decreases the size of the box, then the biggest box within the constrictions is 5, 5, 5”. Then it would be reasonable to say as Dan points out “… because those three numbers are 15 in total”, because it means that the only way of making the box larger is to increase the total sum of the dimensional values. The largest possible box as made probable by the Python printouts, would then be the 5, 5, 5.

In the last part there seems to be a different rule present, “x=y, and increasing one of the dimensional values leads to a bigger box”. He suggests a case of 5, 5, and 7, as a combination of two of the boxes. This result in a taller box whilst still keeping the base surface the same as the biggest box within the constriction, and therefore making a bigger box. It is a valid logic on its own, although this is not within the original constriction within the task, or the question. The “correct” answer that is probed for in the questioning is that it is not possible within the constrictions to make a bigger box. But if it is reasoned that the only way of making a larger box is to increase one of the dimensions, and therefore making the total sum bigger than the constriction initially allows, then it would mean that it is reasoned that it is not possible to make the box bigger than the 5, 5, 5. The “correct” answer to the task and questioning, is given in the answer to the “incorrect” question.

Could there exist a smaller box?

Alf 3

In this section the student Alf and interviewer is talking about if it is possible to get a smaller box than the 1 times 1 times 13. The student suggests a box that lies between this, and the 2 times 2 times 11 box, to be smaller than the smallest box. The interviewer tries to get the student to reflect on this fact, but the student think that the box must be smaller or the same size.

105 RE: Interesting. So, what about the smallest that we found? The smallest we found, that was?
106 Alf: That one, was it not? (Points at the 1x1x13 model)
107 RE: Yeah, that is right. The 1 times 1 times 13 one there.
108 Alf: Yes.
109 RE: So, could we then do something with those numbers to make it even smaller?
110 Alf: Ehm. If you had. Because you could take for example 12,5 and like 1,5, then I think it would have become a little smaller. I don’t know, I am not completely sure.
111 RE: No. No, but that is interesting. So how, what do you think the reason for this is, what do you think could be the reason for why it would be smaller then?
112 Alf: Because one of them will get smaller, then one of them will be bigger but the other will be like, smaller.
113 RE: Mhm. But then, if we then look at the other one there, that’s the. (Points at the second model)

114 Alf: 2 times 2 times 11.
115 RE: That one is 2 times 2 times 11 yes. So, that is then, that gets a little bigger in the x and y directions, while the z directions get a little smaller.
116 ALf: Yes.
117 RE: So, if we then in a way, then will. Then we see that the surface area gets a little bigger, and the volume get a little bigger. (Points at that part in the script output on the computer) So, in relation to what you pointed out now in regard to the dimensions of the 1 times 1 times 13.
118 Alf: Yes.
119 RE: Because if that one should be 1,5 times 1,5 times 12.
120 Alf: Yes.
121 RE: Then we are starting to get closer to that one, 2 times 2.
122 Alf: Yes.
123 RE: Then it gets, the volume gets little bit bigger. Will then, eh, eh, when x is 1,5, and y is 1,5, and z is 12, will then the volume get bigger, or will it get smaller? Surface area and volume.
124 Alf: I think it will get smaller, or the same size really.
125 RE: Heh, yes.
126 Alf: Yes.

This is very reminding of the Alf 2 section, when it was about a bigger box than the biggest box. It seems that there is a factor of “since it is being asked, there must be an answer”, and it is possible that there is some guesswork going on. None of the calculations has been done by hand, and none of the alternative boxes that have been proposed have been checked specifically during the interview, so this could in part explain why there is some inconsistency in what a bigger box mean. This gets substantiated on line 112 where Alf say that “… one of them will get smaller, then one of them will be bigger but the other will be like, smaller”, he is referring to the dimensions after suggesting that 1,5 times 1,5 times 12 will net a smaller box.

If on the other hand, there is a dimensional reasoning resulting in a smaller box, that might be something like “if the dimensions get smaller, then this will result in a smaller box”. The antecedent being “smaller dimensions”, and the consequent “smaller box”. The problem here being that it does not specify what dimensions needs to get smaller, and since the student does remember to keep the constriction of the dimensional sum of 15, the box might not necessarily become smaller. The case the student proposes is 1,5 times 1,5 times 12, making the largest dimension smaller and putting the rest onto the smaller dimensions. By the logic of affirming the antecedent, making the dimensions smaller, at least one of them, result in a smaller box.

As the interview progresses the box in the printout that is slightly larger is pointed out. Trying to make the student reflect on the fact the box proposed by the student is a middle ground between the smallest box and the box that’s a bit larger. This is setting up the possibility for the student of assessing different cases and results, giving an opportunity for inductive reasoning, and “updating” the rule to gain a new understanding. But it does not seem to have the desired effect. The student keeps to the originally set rule and disregards the trend of the sizes of the boxes when moving length away from the z dimension and putting it on the x and y dimensions.

The reason for this problem of recognizing the pattern of the size trend is very unclear. The student seems to have no problem of recognizing the different boxes in 3D-models and connecting them to the Python printout. It could be because the initial rule is quite strong, or it is held in place stronger because of the uncertainty of what it means for it to be a bigger box. This combined with the notion of “since the question is asked, there must be an answer” and the guesswork that this entails. Might result in the obstruction of an inductive reasoning, and thus make it difficult to recognize the observations of different cases and results, and therefore difficult to reason a different or updating the rule.

Christine 3

In this section we can see the student Christine talking about the possibility of finding a smaller box than the smallest box printed out by the Python program. She asks if it is possible to have smaller numbers than 1 and figures it should get smaller if two of the dimensions is set as less than one and the remainder of 15 put into the last dimension.

218 RE: No, no, absolutely. Ehm, so, if we then go the other way then. So, if we look at the one that has the least, so that is 1 times 1 times 13. So, is there a possibility for it to have an even smaller surface area, even smaller area, no, volume I mean? If we had changed the dimensions? For this is this one that is 1 times 1 times 13. Could we do

something with these numbers, and still have, eh, that it was 15 in total, and get an even smaller surface area, an even smaller volume?
219 Christine: Is it possible to have smaller than 1, or?
220 RE: Yes, that’s possible.
221 Christine: Maybe like 0,5 times 0,5 times 14, is that right?
222 RE: Yeah.
223 Christine: Yes. No, wait, 13,5 is it … Yes, it is 14, yes 14.

This section builds on the prior reasoning on the biggest box, earlier in the interview. During the biggest box section all the boxes that was generated by the Python script was connected to the 3D models, and the student was also made aware of the trend of the sizes of the boxes as the dimensions changed. These can all be seen as multiple different cases and results, setting up for inductive reasoning to generate rules from. In this case as the boxes get smaller towards the smallest box on the printout 1 times 1 times 13, it can be observed that the x and y dimensions get smaller as more and more of the values gets allocated to the z dimension. Whether or not the student reasons on the geometric understanding of how the boxes will change in shape as we keep doing this, she does reason a probable solution. That if you make the x and y dimensions smaller than 1, and allocates the difference to the z dimension, you keep all the dimensional constrictions defined in the task. And still get a box that will presumably be smaller than the smallest box, because it is made probable by the trend of diminishing sizes on the printout.

The rule that was generated by this reasoning might be “as the x and y dimensions get smaller the box will get smaller”. The antecedent being “smaller x and y dimensions” and the consequent being “smaller box”. After the rule has been generated, she could answer the question of the task by deductive reasoning. The case she then proposes is 0,5 times 0,5 times 14, this is an affirmation of the antecedent which is a valid logic.

Dan 3

This is the last part of the last interview, and it was cut a bit short, but it still has a small fragment of interesting data. Right before this interaction fellow students of Dan were starting to leave outside the windows of where we were sitting, and the bell rang right after line 314. In this part Dan was asked if it was possible to make a smaller box than the 1 times 1 times 13, to which Dan suggests setting a lower height for the box. As with the other questions, the constriction of a sum of 15 is disregarded, or at least not commented on. Although it is true the box would get smaller, or not as tall, the constriction is not satisfied, unless the other dimensions get increased.

313 RE: Yes, I will not keep you for much longer. So, that is the smallest as we found out. The 1 times 1 times 13. And this was the one with least surface area. Is it possible to make, is it possible to have some dimensions that could have gotten it even smaller, in other words less volume, less surface area?

314 Dan: Eh, could have had less, less height.

At this point less of the thinking activity is based in the mathematical task, and more on the things happening outside. As he concluded with the bigger box task, that to make a bigger box you need to increase the size, or height as he suggests. In the task of making a smaller box he immediately concludes that the height needs to be reduced. This might be tied into a rule of “if the box gets higher the size increases, if the box gets lower the size decreases”. The antecedent relevant for this task being “lower box”, resulting in the consequent “smaller box”. In the case that the student proposes; that the box had less height, then that would result in a smaller box. However, as with the biggest box task, it is very unclear and possibly unlikely that an inductive reasoning has taken place to evaluate the trend of the box sizes according to their dimensional values.

Results and discussion

There are two things that are important to note. Firstly, all calculations that was worked with in the tasks and the interview, was not explored manually, they were only printed out by the Python script. The focus is on the reasoning on these abstracted outputs. Secondly, the rules that are presented in the analysis that the students reasoned are not necessarily how the students think, it is not possible to read their minds. Also, they are not necessarily mathematically correct, for that more exploration of the mathematical concepts would be needed. It is not possible to know if whether the programming hindered or helped the mathematical reasoning, the output from the program gives an abstracted result of many different possible calculations which would otherwise have had to be done manually. What is explored here is how the students manage to utilize this information.

The analysis shows some very varied findings between the three students. Something that stand out is the pattern recognition on the sizes on the printout from the program. As seem to be quite clear is that Christine manages to spot the patterns of what happens when moving away from the box of 5 times 5 times 5, the size decreases. And with this ends up saying she can't think of a set of values that leads to a bigger box. The same is true for the pattern of decreasing sizes going towards the smallest box on the printout and manages to utilize this to figure that it should be possible to reduce the x and y dimension to less than 1. This pattern seems not to be recognized by Alf, although it clearly shows that he sees the connections with list of boxes and the 3D-models. Since he does not seem to be aware of the patterns, he ends up suggesting a box that is in between the two smallest boxes, and says he thinks that this would make a smaller box. The concerning thing here is that he did not see the problem with this reasoning even after he was made aware of this fact. Dan on the other hand is more difficult to understand. When he suggested that the biggest box is the biggest because the dimensional sum is 15, it seems that he did not understand the task fully. The same goes for; to make the box bigger you can increase the height without changing the other dimensions. This is either because the task was not understood, or it was because he reasoned that as the only solution, meaning he understood the task but answered a different question that the one that was asked.

Christine takes her observations into account when answering the questions related to tasks. While Alf either does not recognize the patterns or disregards them in favor of an already reasoned rule. Dan seems to answer a different question than what is asked, either because he does not understand the task and therefore it is convenient to answer something else, or because he reasons that an answer without a solution is not a viable answer. It shows that when working with abstract data like this it is very important to be able to recognize the patterns and allow those patterns to change the way one think about the problem and task, and through this understanding the task and what is required to find a solution to the problem. These are not terribly different from the results of Hadjerrouit and Hansen's (2022) study on CT, which was presented under prior research, whereas they found that an important factor in successfully being able to utilize CT, it was necessary to have a clear understanding of the mathematical concepts.

Using programming in this way gives a possibility of trying many different dimensions explicitly, like if by hand. This is different than exploring a similar data set using a function graph, which would present be able to represent the same data, although more visually. As pointed out it is an abstraction, meaning the specific calculations does not have to be done for each individual box, there is an input and an output. Preferably the students should make the program themselves so that they understand the task, and the mathematical concepts underneath. As with the results from Hadjerrouit and Hansen's study (2022), showing the importance of an understanding of the underlying concepts both in programming and mathematics. This would also ensure that the constrictions that are present in the program is understood. After the algorithm and the program is made, the number crunching can be done with the computer, and there can be done a reasoning on the box sizes in a more macro perspective. So, for an activity like this to be properly productive for reasoning and learning, multiple things have to be in place. A clear understanding of the task and the problem, the mathematical concepts, and the algorithmic and computational principles.

This has implications for mathematical education in three ways. Firstly, it is very important to work with task understanding, the process of reading and making sure the student understand the task. This can be done by specifically giving students tasks which have a complex nature, so that the students get practice in figuring out what the task is about and how it work before attempting to solve it. Secondly, which is very much tied to the understanding of the task. In the case of students like Dan where there is doubt whether or not the student understands the task, or if the student thinks differently about the process of answering the task. An awareness of the discoveries done by Papert and others related to constructionism, on whether or not the student can be characterized as a bricoleur, where the bricoleur would go drastically different about solving and understanding the task. In the case of these students, it is possible that some guidance towards understanding what is needed when answering or solving mathematical task, especially in the case of standardized school mathematics testing regime. Thirdly, using programming in mathematics education, it is important to note that this is a high-level concept. It is dependent on multiple factors needing to be in place to be properly productive for learning. It might be an activity or a tool that gives the opportunity to explore and develop a deeper understanding of a concept, but it might also be a hindrance or something that has to be learned before learning the thing that is the goal in the first place. This ties into one of the important factors from Olteanu's work (2020), that it is important for the teacher to have a good understanding of both the programming-, and the mathematical concepts.

Conclusions and my own evaluations of the project

This concludes this thesis on students' reasoning. The things that were presented in the introduction has been covered, and what remains now is an evaluation of the project as a whole. There are many things that I am not too happy with about the study, many of which ties into the inexperience of doing research like this. The tasks could have been a lot better planned out, it might have been better to have more open and explorative tasks, and to accomplish this they should perhaps have been much smaller. Another thing that could have gone better is the interviews, the interview guide should have had much more emphasis on gaining more specific insight into how the students think, and with that getting to understand better how the student reasons. There would have been more time and emphasis on this had the interview guide focused less on the students view of programming in general, although it was interesting to hear this as well, it did not contribute as much to the study as originally thought. And on a more practical note, also the time scheduling, it would have been better for the study if greater emphasis on getting the research done quicker to allow for more time for analysis and to finish writing the thesis.

On a positive note, the thesis represents a new insight into how a student might attack a mathematical problem, and on that basis, it gives new knowledge to be able to make better lectures and better mathematical problems for the students to solve. This goes for multiple subjects, but specifically when working with programming in mathematics. It is clearly very important to focus on laying a solid groundwork when utilizing programming in mathematics education. This goes for the programming part, since it is preferable to understand the workings of a program, if however, one would want to use a skeleton program it is preferable to explain what this means, so that it is understood it is used by way of abstraction. And it goes for the mathematical part, it is important to understanding why the program does what it does, because it is needed to understand "what" the output is. And most importantly a powerful toolset among others, of understanding how the students understand a problem or a solution in general by having the experience of doing the analysis on the reasoning in the study.

In my opinion going forward it would be interesting to see a study on the actual specific benefit of programming in mathematics education. Learning programming takes time, of which there is a limited amount of in school. Do the benefits of learning how to program, and the potential learning outcome from working with the concept using programming, outweigh the time it takes to learn programming? I think to be able to answer this there should be done a comparative study between learning a mathematical concept using programming, versus using digital tools like GeoGebra, excel, and so on, versus using non digital ways of learning the same concept. This must of course be a concept that can be learned using all these methods. It should then be possible to compare the differences in competences and proficiency in that concept. Is it more productive spending more time on the actual concept, versus learning the digital tools or programming? This would be informing to which degree programming should be taught in elementary school, is it most important to get an understanding of how programming works using block-programming like Scratch or Makecode designed for school and education, or is there a benefit to learning more general-purpose programming with for example Python?

References

Bryman, A., Sloan, L., Foster, L., & Clark, T. (2021). *Bryman's Social Research Methods* (6 ed.).

Hadjerrouit, S., & Hansen, N.-K. (2022). Undergraduate Mathematics Studens Engaging in Problem-Solving Through Computational Thinking and Programming: A Case Study. In D. Ifenthaler, P. Isaías, & D. G. Sampson (Eds.), *Orchestration of Learning Environments in the Digital World* (pp. 197-214). Springer. https://doi.org/https://doi.org/10.1007/978-3-030-90944-4_11

Kunnskapsdepartementet. (2020). *Læreplan i matematikk 1.-10. trinn*. (MAT01-05). Retrieved from https://data.udir.no/kl06/v201906/laereplaner-lk20/MAT01-05.pdf?lang=nob

Logo Foundation. (2015). *Logo History*. Retrieved 13.05 from https://el.media.mit.edu/logo-foundation/what_is_logo/history.html

Olteanu, C. (2020). Programming, mathematical reasoning and sense-making. *International Journal of Mathematical Education in Science and Technology*, 1-19.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. Basic Books, Inc.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1-11.

Papert, S. A. (2020). *Mindstorms children, computers, and powerful ideas* (3 ed.). Basic Books. (1980)

Reid, D. A., & Knipping, C. (2010). *Proof in mathematics education: Research, learning and teaching*. Brill.

Resnick, M., Ocko, S., & Papert, S. (1988). LEGO, Logo, and design. *Children's Environments Quarterly*, 14-18.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A:*

Mathematical, Physical and Engineering Sciences, *366*(1881), 3717-3725.

Wing, J. M. (2011). Research Notebook: Computational thinking: What and why. *The link*.

Appendix

The consent-form

Vil du delta i forskningsprosjektet?

"Students' mathematical reasoning when solving geometry problems using programming"
"Elevers matematiske resonnement når de løser geometrioppgaver med programmering"

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

Formål
Dette er et forskningsprosjekt tilknyttet en masteroppgave innenfor fagfeltet matematikkdidaktikk. Målet med prosjektet er å studere hvordan programmering kan brukes i matematikkundervisning. I denne omgang sees det på hvordan programmering kan legge til rette for utforsking og resonnering i forbindelse med geometri. For deltakerne vil det ta et par skoletimer, og litt tid i etterkant for et intervju/samtale om aktiviteten.

Det er to forskningsspørsmål som skal analyseres i prosjektet. Det ene er «Hvordan kan programmering brukes for å legge til rette for utforskning i geometri i ungdomskolen?», og det andre er «Hva karakteriserer elevers resonnement når de løser geometrioppgaver med programmering?».

Hvem er ansvarlig for forskningsprosjektet?
Universitetet i Agder er ansvarlig for prosjektet.

Hvorfor får du spørsmål om å delta?
I dette prosjektet har jeg behov for en liten gruppe på 5 elever på niende trinn som vil være med på et opplegg hvor vi skal utforske geometriske sammenhenger med programmering. I dette tilfellet har jeg spurt mattelæreren deres om noen som er litt ekstra matematikkinteresserte, og som gjerne har litt erfaring med programmering fra før, for eksempel gjennom valgfag.

Hva innebærer det for deg å delta?
Hvis du velger å delta i prosjektet, innebærer det at du er med på et opplegg hvor det skal jobbes med geometri med programmering, også i etterkant av opplegget er det et intervju.

- Opplegget kommer til å foregå i skoletiden over to skoletimer. Vi kommer til å være på et grupperom eller liknende på skolen, utenfor resten av klassen. Det kommer til å bli presentert en oppgave i geometri med underoppgaver knyttet til denne, som det da skal utforskes gjennom programmering i Python. Nødvendig programvare, innføring, og veiledning i programmering med Python blir lagt til rette for underveis i opplegget.
- Datainnsamling knyttet til dette opplegget blir gjort ved:
  - Skjermopptak på din egen datamaskin (Windows game bar skjermopptak), dette opptaket tar opp hva som gjøres på skjermen, slik at det er mulig å analysere hvordan programmering blir brukt til oppgavene.
  - Video- og lydopptak av opplegget, dette er i form av kamera og mikrofon som tar opptak mens opplegget holder på, slik at det er mulig å analysere hvordan matematikken blir formulert og kommunisert mellom deltakerne, og meg som veileder under opplegget.
  - Observasjoner som blir notert av meg underveis i opplegget, dette er i form av notater som ikke nødvendigvis kommer med i opptakene, men som kan være viktig for å forstå hvordan deltakerne tenker for å løse oppgavene.
- Etter opplegget vil det være et intervju, dette vil også være i skoletiden. Spørsmålene vil være om sånne ting som: Hva du synes om opplegget? Hva du synes om oppgaven? Hvordan tenkte du når du løste oppgaven? Hva var det som gjorde at du løste oppgaven på den måten?

- Under intervjuet kommer jeg til å ta lydopptak og notater.

Hvis det er ønskelig, kan foreldre ta kontakt for å se intervjuguiden i forkant av intervjuet ved å ta kontakt.

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. Det vil ikke påvirke ditt forhold til skolen/lærere.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger
Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- De som kommer til å ha tilgang til opplysningene er meg og mine veiledere på prosjektet ved Universitet i Agder.
- Alt datamateriale blir lagret sikkert på institusjonens datasystemer slik at uvedkommende ikke får tilgang til dette.

Deltakerne vil ikke kunne gjenkjennes i endelig publikasjon (masteroppgaven).

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?
Opplysningene anonymiseres når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er sommeren 2022. Planlagt dato for prosjektslutt er 15.06.2022. Men i tilfelle det oppstår forsinkelser ved godkjenning og avslutning av prosjektet vil opplysningene kunne bli oppbevart frem til 15.06.2023.

Når prosjektet er godkjent og avsluttet vil alle personopplysninger og opptak bli slettet.

Hva gir oss rett til å behandle personopplysninger om deg?
Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Universitetet i Agder har NSD – Norsk senter for forskningsdata AS 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:

- Universitetet i Agder ved Robert Eidså (student), e-post (robert@eidsa.com), telefon: 41726193. Shaista Kanwal (veileder/prosjektansvarlig), e-post (shaista.kanwal@uia.no), telefon: 38142405.
- Vårt personvernombud Johanne Warberg Lavold, e-post (johanne.lavold@uia.no), telefon: 41212048.

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

- NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 53 21 15 00.

Med vennlig hilsen

Shaista Kanwal	Robert Eidså
(Forsker/veileder)	(Student)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet ”Elevers matematiske resonnement når de løser geometrioppgaver med programmering”, og har fått anledning til å stille spørsmål. Jeg samtykker til:

- ☐ å delta i opplegget om utforsking av geometri med programmering.
- ☐ å delta i intervjuet i etterkant av opplegget.

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

----
(Signert av prosjektdeltaker, dato)

(Signert av forelder/verge, dato)

The sheet of problems from the arrangement

Sammenhengen mellom overflate og volum av et rektangulært prisme

Målet med oppgaven er å studere hvordan overflatearealet endres hvis volumet endres. Programmene kan enten lages underveis eller kopieres og limes inn fra bibliotekfilen som er gjort tilgjengelig)

Overflateareal:

Arealet av en rektangulær overflate er lengde multiplisert med bredde. Men gitt informasjonen bredde, dybde, og høyde. Kan du kalkulere arealet av overflaten til et rektangulært prisme? (en boks)

a) Ved å velge passende informasjon for bredde, dybde, og høyde. Skriv et program som utfører beregningen og oppgir overflaten til prismet.
b) Gitt at **summen** av bredde, dybde, og høyden må holdes konstant. Gitt at den konstante summen av dimensjonene er 15. Hva er de optimale dimensjonene til boksen hvis målet er det største mulige overflatearealet? Forklar.
c) Gitt samme begrensninger som i oppgave b. Finnes det optimale dimensjoner av boksen hvis målet er å ha det minste mulige overflatearealet? Forklar.
d) Skriv et program som tester alle mulige kombinasjoner (innenfor en rimelig oppløsning) av dimensjoner slik at du kan finne ut det største mulige overflatearealet. Er resultatene rimelige?
e) Skriv også et program som finner det minste mulige overflatearealet. Er disse resultatene rimelige?

Volum:

Volumet av et rektangulært prisme (en boks), er bredde, multiplisert med dybde, multiplisert med høyden.

a) Skriv et program som utfører beregning og oppgir volumet til et rektangulært prisme.
b) Gitt at summen av bredden, dybden, og høyden må holdes konstant. Si at denne konstanten er som tidligere 15. Hva er de optimale dimensjonene til boksen for å maksimere volumet? Forklar.
c) Gitt samme begrensninger som i oppgave b. Finnes det optimale dimensjoner av boksen hvis målet er å finne det minste mulige volumet? Forklar?
d) Skriv et program som tester alle mulige kombinasjoner (innenfor en rimelig oppløsning) av dimensjoner slik at du kan finne det største mulige overflatearealet. Er resultatene rimelige? Hvordan er disse resultatene sammenlignet med det største mulige overflatearealet?
e) Gjør det samme for å finne det minste mulige volumet. Er disse resultatene rimelige?

Sammenhengen mellom overflateareal og volum:

Se for deg en boks.

a) Hva skjer med overflatearealet hvis du kutter boksen i to? Forklar.
b) Hva skjer med volumet hvis du kutter boksen i to? Forklar.
c) Hvordan endres overflatearealet og volumet hvis høyden endres? Gitt at bredde og dybde, holdes like, altså bredde=dybde. Høyde kan variere, men summen av bredde, dybde, og høyde må holdes konstant på 15. bredde + dybde + høyde = 15, bredde = dybde.
d) Gitt reglene i oppgave c, skriv et program som finner de optimale dimensjonene for boksen for å maksimere volumet **og** overflatearealet.
e) Gitt reglene i oppgave c, skriv et program som finner de optimale dimensjonene for boksen for å minimere volumet **og** overflatearealet.

The prewritten Python scripts

```
#Funksjon som returnerer det beregnede volumet av en kuboide
def volum(x,y,z):
    volum = x * y * z
    return volum


#Funksjon som returnerer det beregnede overflatearealet av en kuboide
def overflate(x,y,z):
    areal = 2*x*y+2*z*y+2*x*z
    return areal


#En funksjon som genererer lister med x,y,z dimensjoner
def dimensjonsgenerator(minst, storst):
    x = range(minst, storst)
    y = range(minst, storst)
    z = range(minst, storst)
    dimensjoner = []
    for n in x:
        for o in y:
            for p in z:
                dimensjoner.append([n,o,p])
    return dimensjoner


#Funksjon som kan sortere ut kun de kombinasjonene av dimensjoner som passer med begrensningen på summen av dimensjonene.
def dimensjonSorter(listeMedDimensjoner, sumKonstant):
    usorterteDimensjoner = listeMedDimensjoner
    sorterteDimensjoner = []
    for r in usorterteDimensjoner:
        x = r[0]
        y = r[1]
        z = r[2]
        if (x+y+z) == sumKonstant:
            sorterteDimensjoner.append([x,y,z])
    return sorterteDimensjoner


#Funksjon som printer ut alle dimmensjonene og volumet som blir matet inn i den.
def dimensjonerOgVolum(sorterteDimensjoner):
    for q in sorterteDimensjoner:
        x = q[0]
        y = q[1]
        z = q[2]
        print("Dimensjoner: x=", x, "  y=", y, "  z=", z, "  Volum=", volum(x,y,z))


#Funksjon som printer ut alle dimmensjonene og overflatearealet som blir matet inn i den.
def dimensjonerOgOverflate(sorterteDimensjoner):
    for s in sorterteDimensjoner:
        x = s[0]
        y = s[1]
        z = s[2]
        print("Dimensjoner: x=", x, "  y=", y, "  z=", z, "  Overflate=", overflate(x,y,z))


#Funksjon som printer ut alle dimmensjonene, overflatearealet OG volumet som blir matet inn i den.
def dimensjonerOverflaterOgVolum(sorterteDimensjoner):
    for t in sorterteDimensjoner:
        x = t[0]
        y = t[1]
        z = t[2]
        print("Dimensjoner: x=", x, "  y=", y, "  z=", z, "  Overflate=", overflate(x,y,z), "   Volum=", volum(x,y,z))


#En alternativ sorteringsfunksjon som også legger til begrensningen om at x og y må være like store.
def oppgave3cSorter(listeMedDimensjoner, sumKonstant):
    usorterteDimensjoner = listeMedDimensjoner
    sorterteDimensjoner = []
    for r in usorterteDimensjoner:
        x = r[0]
        y = r[1]
        z = r[2]
        if (x+y+z) == sumKonstant and x == y:
            sorterteDimensjoner.append([x,y,z])
    return sorterteDimensjoner
```

Intervjuguide

I etterkant av opplegget med geometri og programmering

I kjernen handler det om å avdekke oppfatninger, forståelser, og meninger tilknyttet programmering og geometri, samt resonnementer knyttet til geometrioppgavene i opplegget.

Oppstart

Generelle korte spørsmål (målet er i størst grad å bli vant til opptakeren i rommet og settingen)

- Hva syns du om opplegget?
- Hva syns du om oppgavene?
- Hva syns du om å bruke Python de timene?

Bakgrunn

Programmering:

- Har du vært borti programmering før?
  - Hvordan da?
  - Valgfag i programmering?
  - Programmering i matematikkfaget?
- Har du vært borti Python før?
  - I hvilken grad?
  - Hørt om det? Hva har du hørt om det?
- Har du vært borti eller hørt om andre programmeringsspråk?
  - For eksempel: Scratch? Makecode for micro:bit? Logo? HTML? Javascript?
  - Når har du brukt de?/hørt om de?
  - Hva har du brukt de til?
  - Hva synes du om disse programmeringsspråkene?
  - Hva synes du om det i forhold til Python?
- Hvordan vil du beskrive programmering eller koding?
  - Generelt?
  - I forbindelse med dataprogrammer, spill?
  - Dingser?
  - Samfunnet?
  - Verden?
- Hvordan vil du beskrive det å lære programmering generelt?
  - Nyttig? Unyttig?
  - Bruksområder?
  - Interessant? Uinteressant?
- Hvordan vil du beskrive det å lære programmering i og tilknyttet matematikk på skolen?
  - Nyttig? Unyttig?
  - Bruksområder?
  - Interessant? Uinteressant?
  - Positive sider? Negative sider?

Oppgavene

Det første vi gjorde var å bare teste ut python-print, hvor vi bare tastet inn et sett med dimensjoner.

overflate

Deretter brukte vi noen funksjoner for å printe ut alle dimensjonene som blir 15 til sammen. Og som viser overflaten av det.

- I oppgaven så settes det opp begrensninger på hvordan lengde og bredde er i forhold til hverandre, mens høyden derimot; kan variere fritt så lenge de totale lengdene er likt. Hvordan påvirker dette overflatearealet til boksen?
    - Hvis høyden øker, hva skjer med lengden og bredden?
    - Hva skjer med overflatearealet?
- Hva hvis begrensningen hadde vært på lengde og høyde for eksempel, og bredden kunne variere fritt. Hvordan ville det påvirket resultatet?

Overflate og volum

- I oppgaven så settes det opp begrensninger på hvordan lengde og bredde er i forhold til hverandre, mens høyden derimot; kan variere fritt så lenge de totale lengdene er likt. Hvordan påvirker dette volumet til boksen?
    - Hvis høyden øker, hva skjer med lengden og bredden?
    - Hva skjer med volumet?
- Hva hvis begrensningen hadde vært på lengde og høyde for eksempel, og bredden kunne variere fritt. Hvordan ville det påvirket resultatet?

Overflate og volum med enda en begrensning

Deretter til slutt så satt vi inn en ekstra begrensning med at x og y måtte være like lange.

    - Kan du si noe om hvor sikker du er på at du har funnet de beste dimensjonene for å maksimere og minimere volumet?

Overflateareal

- Da du først leste oppgaven, hva tenkte du?
  - Løsningsstrategier?
  - Planlegge?
  - Bare prøve noe?
- Kan du utdype/forklare svaret du gav i oppgaven?
- I oppgaven så settes det opp begrensninger på hvordan lengde og bredde er i forhold til hverandre, mens høyden derimot; kan variere fritt så lenge de totale lengdene er likt. Hvordan påvirker dette overflatearealet til boksen?
  - Hvis høyden øker, hva skjer med lengden og bredden?
  - Hva skjer med overflatearealet?
- Hva hvis begrensningen hadde vært på lengde og høyde for eksempel, og bredden kunne variere fritt. Hvordan ville det påvirket resultatet?
- Hvordan gikk du frem for å lage programmet?
- Programmet som du lagde testet for «et antall» verdier av de tre dimensjonene. Hvorfor valgte du de verdiene?
  - Kan du si noe om hvor sikker du er på at du har funnet de beste dimensjonene for å maksimere og minimere overflatearealet?

Volum

- Da du først leste oppgaven, hva tenkte du?
  - Løsningsstrategier?
  - Planlegge?
  - Bare prøve noe?
- Kan du utdype/forklare svaret du gav i oppgaven?
- I oppgaven så settes det opp begrensninger på hvordan lengde og bredde er i forhold til hverandre, mens høyden derimot; kan variere fritt så lenge de totale lengdene er likt. Hvordan påvirker dette volumet til boksen?
  - Hvis høyden øker, hva skjer med lengden og bredden?
  - Hva skjer med volumet?
- Hva hvis begrensningen hadde vært på lengde og høyde for eksempel, og bredden kunne variere fritt. Hvordan ville det påvirket resultatet?
- Hvordan gikk du frem for å lage programmet?
- Programmet som du lagde testet for «et antall» verdier av de tre dimensjonene. Hvorfor valgte du de verdiene?
  - Kan du si noe om hvor sikker du er på at du har funnet de beste dimensjonene for å maksimere og minimere volumet?

Sammenhengen mellom overflateareal og volum

- Da du først leste oppgaven, hva tenkte du?
  - Løsningsstrategier?
  - Planlegge?
  - Tegne?
- Kan du utdype/forklare svaret du gav i oppgaven?
- Hvordan gikk du frem for å lage programmet?

- Programmet som du lagde testet for «et antall» verdier av de tre dimensjonene. Hvorfor valgte du de verdiene?
    - Kan du si noe om hvor sikker du er på at du har funnet de beste dimensjonene for å maksimere og minimere volumet?
-

Transcription of the interviews

	Timespan	Content
1	0:59,2 - 1:19,6	RE: Ah, så. Ehmmm. Så det er liksom en liten sånn en debrief om opplegget i går.
2	1:19,5 - 1:19,8	Alf: Ja.
3	1:19,8 - 1:25,4	RE: Det var litt sånn jeg tenkte på. Så hva, hva syntes du om opplegget?
4	1:25,3 - 1:27,6	Alf: Nei, det var jo, eh, gøyere enn vanlig matte.
5	1:27,5 - 1:29,3	RE: Ja.
6	1:29,3 - 1:32,9	Alf: Sånn liksom gjør ting, også få andre ting ut av det liksom.
7	1:32,8 - 1:33,8	RE: Ja, absolutt.
8	1:33,8 - 1:34,2	Alf: Ja.
9	1:34,2 - 1:40,3	RE: Ja, jaja. eheh. Det, men de oppgavene, nå gikk vi liksom fort igjennom de oppgavene sånn sett.
10	1:40,2 - 1:41,0	Alf: ja.
11	1:41,0 - 1:41,8	RE: Hva syntes du om de sånn ..?
12	1:41,7 - 1:44,1	Alf: Nei, jeg syntes det var ikke noe vanskelig akkurat.
13	1:44,1 - 1:44,5	RE: Nei?
14	1:44,4 - 1:44,5	Alf: Altså det var sånn, greie oppgaver.
15	1:46,8 - 1:51,3	RE: Eheh, javisst. Eeeh, har du brukt Python før, eller et eller annet sånn?
16	1:51,3 - 1:54,2	Alf: Nei, eller jeg har sett på det for [Bjorn] driver jo med det.
17	1:54,2 - 1:54,9	RE: Åja.
18	1:54,9 - 1:56,9	Alf: Så jeg har sett på han når han gjør det.
19	1:56,9 - 1:59,1	RE: Eheh. Ja, for, dere har ikke brukt det i timene eller ;?
20	1:59,0 - 1:59,9	Alf: Nei ...
21	1:59,8 - 2:04,9	RE: ; matten sånn ellers, nei? Nei, nei visst. RE: Eh, men har du vært bort noe ann type programmering før?
22	2:04,9 - 2:07,9	Alf: Eh. Jeg hadde et halvt år med programmering på skolen.

23	2:07,8 - 2:08,4	RE: Åja, som valgfag da?
24	2:08,4 - 2:08,9	Alf: Ja.
25	2:08,8 - 2:11,0	RE: Ja. Å slags programmering gjorde dere da?
26	2:11,0 - 2:14,0	Alf: Nei, da skulle vi bare lære å flytte forskjellige brikker og sånn.
27	2:13,9 - 2:16,9	RE: Ja, altså da, sånn som Scratch da?
28	2:16,8 - 2:17,1	Alf: Ja, egentlig.
29	2:17,1 - 2:27,5	RE: Javisst ja, med det er ikke så dumt. Ehm. Skal vi se. Ehm ... Har du vært borti, har du vært bort Micro:bit for eksempel?
30	2:27,5 - 2:28,1	Alf: Nei ..
31	2:28,1 - 2:31,1	RE: Nei. Eller noe HTML eller noe?
32	2:31,0 - 2:31,9	Alf: ...
33	2:31,8 - 2:52,0	RE: Nei. Ehm. Hvordan ville du, hvis du skulle beskrevet programmering. Litt sånn generelt, liksom hvordan, hva er dine oppfatninger av hva programmering er for noe?
34	2:52,0 - 2:58,7	Alf: Eh, at du skriver koder som gjør ting for deg liksom.
35	2:58,6 - 2:59,2	RE: Eheh.
36	2:59,2 - 3:01,1	Alf: Ja.
37	3:01,0 - 3:04,2	RE: Javisst, ja så det er liksom programmere dataen til å utføre oppgaver på en måte.
38	3:04,2 - 3:07,5	Alf: Ja.
39	3:07,5 - 3:11,6	RE: Ehm. Hvordan vil du beskrive det å lære programmering? Vil du si det er nyttig, eller unyttig?
40	3:11,6 - 3:15,4	Alf: Nei, jeg vil jo si det er nyttig med tanke på at verden går inn i en fase der alt er på pc.
41	3:15,4 - 3:16,1	RE: Eheh.
42	3:16,0 - 3:17,3	Alf: Det er vel lurt å kunne programmering.
43	3:17,3 - 3:21,4	RE: Javisst, absolutt. Kan du tenke deg noen sånn bruksområder?
44	3:21,4 - 3:29,3	Alf: Eh, det kan jo være sånn visst, istedenfor å sende folk opp i kraner sånn på byggeplasser, at du programmerer kranene også gjør

		de alt elektrisk.
45	3:29,2 - 3:54,1	RE: Eheh. Javisst, jaja. Det er jo kjempe fornuftig, interessant bruksområde. RE: Ehm. Hvordan vil du se programmering tilknyttet matematikk? Ser du liksom, kan du se en nytte effekt der på en måte sånn?
46	3:54,1 - 4:00,0	Alf: Ja, sånn som det vi dreiv med å finne ut, sånn overflate og volum og sånn.
47	3:59,9 - 4:00,4	RE: Mhm.
48	4:00,3 - 4:06,6	Alf: Det er jo. Det kan være ganske nyttig, det er jo en ganske kjapp måte å regne ut opp til flere regnestykker hvis du gjør det riktig.
49	4:06,6 - 4:30,4	RE: Eheh. Ja, absolutt. Det er interessant. RE: Ehm. Så... Det var litt sånn generelt da. Så tenkte vi kunne prate litt sånn konkret om de oppgavene vi gjord da. RE: Det første vi gjorde, skal vi se. RE: Det aller første vi gjorde var å teste ut en print funksjon da.
50	4:30,1 - 4:39,6	RE: Eh, så, men, så gjorde vi jo den, hvor vi fikk printet ut alle de. Eller først tok vi vel først bare overflatene.
51	4:30,4 - 4:30,5	Alf: Ja.
52	4:39,6 - 4:39,7	Alf: Ja.
53	4:39,6 - 4:42,7	RE: Ehm, men så printet vi jo ut så vi fikk ut alle.
54	4:42,6 - 4:43,7	Alf: Ja.
55	4:43,7 - 5:03,0	RE: Eh. Da tenker jeg at vi tar. (utydelig) Altså nå satt vi jo opp noen begrensninger, at det er. At summen av høyden, eller bredden, høyden og lengden. Det skulle være 15, med de begrensningene da.
56	5:03,0 - 5:04,0	Alf: Mhm,
57	5:04,0 - 5:10,0	RE: Så, hvis da høyden øker, hva skjer da med bredden og lengden?
58	5:09,9 - 5:13,2	Alf: M, ja.
59	5:13,2 - 5:28,8	RE: Mhm. Så da, er det jo sånn atte. Så hva skjer da med da med overflatearealet, når vi da endrer på de dimensjonene der?
60	5:28,8 - 5:34,4	Alf: Ja. De kan bli større og mindre. Hvis du. Det kommer ann på hvordan du setter det.
61	5:34,4 - 5:36,4	RE: Eheh. Mhm.

62	5:36,4 - 5:37,8	Alf: Mhm.
63	5:37,8 - 5:45,1	RE: Ehm. Hva hvis begrensningen hadde vært på, istedenfor på høyden, eller da på, begrensningen var jo på.
64	5:45,0 - 5:45,5	Alf: 15?
65	5:45,4 - 6:28,7	RE: På 15 ja. Eh. (Avbrytelse med noen som kom inn i rommet. Lang tid med summing, for å finne ut hvor intervjuet går videre.)
66	6:28,7 - 6:34,0	RE: Der. Vi gjorde jo dette her. Vi hadde fått. (kjører skriptet på en datamaskin under intervjuet.) Vi printet ut alle disse, det blir jo veldig mange tall.
67	6:33,9 - 6:34,2	Alf: Ja.
68	6:35,0 - 7:05,8	RE: Men så lagde vi en. skal vi se, skal bare kjøre det skripted der på nytt, skal vi se. Men vi lagde jo en sånn sak at, skal vi se, oi, ... (knoter med å få skriptet til å kjøre skikkelig.)
69	7:05,7 - 7:07,6	RE: Så dette her er jo da.
70	7:07,5 - 7:11,2	Alf: Den sorterte.
71	7:11,1 - 7:22,3	RE: Mhm, ja for det var den som sorterte littegranne. Og den satt jo på en ekstra begrensning da. Om at vi hadde.
72	7:22,3 - 7:24,1	Alf: x og y skal være like.
73	7:24,1 - 7:43,2	RE: Nettopp, akkurat. Og da er det jo sånn da at så lenge de to er like. Så, så, så er det jo da også en sum da, at det skal ikke være mer enn 15, eller det skal være akkurat 15 da. RE: Men si da hvis, hvis høyden og bredden. for eksempel da y og z skulle vært like, hva ville da skjedd? Ville det sett anderledes ut?
74	7:43,1 - 7:44,2	Alf: Jeg ville jo tro det.
75	7:44,2 - 7:44,9	RE: mhm
76	7:44,9 - 7:47,2	Alf: Med tanke på at volumet ville jo beveget seg alt.
77	7:47,2 - 8:15,0	RE: Mhm. Så hvis vi ser da på disse her figurene her da. Dette her er jo den som er, mm, dette er vel den som er 6 gange 6 tror jeg. Også 3 i høyden. (viser 3D-.printede figurer) RE: Men så da er vi, 6 den veien og 6 den veien, og, så de må være like, og høyden er 3. Men hvis da jeg sier da at den, skal vi si, den x'en der da for eksempel. og høyden skulle vært like.

78	8:14,9 - 8:15,2	Alf: Ja.
79	8:15,1 - 8:21,0	RE: Hva ville skjedd da, ville vi fått de samme tallene egentlig, eller ville vi fått noe anderledes?
80	8:20,9 - 8:26,5	Alf: Nei, jeg tror alt hadde, tallene hadde endret seg med tanke på at den kan være like lang som den vil da.
81	8:26,5 - 8:28,4	RE: mhm
82	8:28,3 - 8:32,7	Alf: Eller det kunne plutselig vært en veldig lang flat en liksom.
83	8:32,7 - 8:33,9	RE: mhm
84	8:33,9 - 8:36,5	Alf: Sånn hvis du hadde satt den oppover sånn.
85	8:36,4 - 8:39,7	RE: Ja, sånn som denne her rett og slett.
86	8:39,7 - 8:40,3	Alf: Ja sånn.
87	8:40,3 - 8:47,8	RE: Eeh, så da vil det jo bli, da vil dette blir, ja hvordan vil dette blir da. Så da er jo.
88	8:47,8 - 8:53,9	Alf: Er ikke den 7, 1, 7, blir det ikke sånn?.
89	8:53,9 - 9:02,4	RE: Mhm, ja visst ja, 7, 1, også 7 ja. Så da liksom den, så da vil jo x'en og z'en være like.
90	9:01,9 - 9:28,7	RE: Ja, absolutt, spennende interessant. Så da kan vi liksom, vi kan egentlig snu på de og da får vi de samme. Hvis jeg forstår riktig? (Sjekker om jeg forstår informanten riktig) RE: Ehm, skal vi se. Når vi da ser på disse her, for disse her har jo da en, eh, vi har jo da printet ut en god del dimensjoner, eller noen få dimensjoner, dette er jo bare de, skal vi se, seks stykker, syv stykker, syv stykker ja,
91	9:02,3 - 9:02,4	Alf: ja.
92	9:28,7 - 9:28,8	Alf: Ja.
93	9:28,7 - 9:50,6	RE: Ehm, og, basert på de som vi har her da. Kan vi være sikre på at vi har funnet de kombinasjonene av dimensjoner som på en måte faktisk gir den minste og den største? Kan det være noen tall som vi ikke ser her på en måte?
94	9:50,5 - 9:53,7	Alf: Eh, jeg vet ikke helt.
95	9:53,7 - 10:04,7	RE: Sånn for eksempel hvist, hvist vi kunne ha brukt kommatall for eksempel, ville vi kunne fått noen andre tall da for eksempel?

96	10:04,7 - 10:05,1	Alf: Ja.
97	10:05,1 - 10:08,9	RE: Ville vi kunne fått noen større overflater og, eller større volum, mindre volum?
98	10:08,8 - 10:13,1	Alf: Ja, jeg vil jo mene at du kan det.
99	10:13,1 - 10:19,2	RE: Mhm. Sånn for eksempel da med, sånn som den som vi fant ut var størst var den ikke det.
100	10:19,1 - 10:21,0	Alf: Jo, 5 gange 5 gange 5.
101	10:21,0 - 10:43,9	RE: Det var 5 gange 5 gange 5. Hadde vi kunne, hvis vi hadde hatt desimaltall hadde vi kunne fått den større? på noe hvis? Hvis vi fremdeles skal ha, vi har fremdeles den begrensningen på at summen skal være 15.
102	10:43,9 - 10:58,0	Alf: Ehm. Jeg mener jo hvis du hadde tatt z ned til for eksempel, 4,3 også, x vært 5,7 da. Så tror jeg han hadde blitt større.
103	10:58,0 - 10:58,5	RE: Eeh. Ja.
104	10:58,5 - 10:58,8	Alf: Ja.
105	10:58,7 - 11:05,8	RE: Interessant. Så men hva med da med den minste som vi fant? Den minste vi fant ut, det var jo da?
106	11:05,8 - 11:08,0	Alf: Den der var det ikke det?. (peker på 1x1x13 figur)
107	11:07,9 - 11:09,9	RE: Ja, stemmer. Den 1 gange 1 gange 13 der.
108	11:09,9 - 11:10,4	Alf: Ja.
109	11:10,3 - 11:16,7	RE: Så, hadde vi da kunne gjort noe med de tallene for å gjøre den enda mindre?
110	11:16,7 - 11:28,2	Alf: Ehm. Hvis du hadde. For du kunne jo tatt for eksempel 12,5 også 1,5 liksom, så tror jeg den hadde blitt mindre. Jeg vet ikke, jeg er ikke helt sikker.
111	11:28,1 - 11:39,1	RE: Nei. Nei, men det er spennende, interessant. Så hvordan, hvordan tenker du at det, hva tenker du kan være grunnen til hvorfor det vil kunne bli mindre da?
112	11:39,0 - 11:44,6	Alf: Med tanke på at en av de blir mindre så vil jo, den ene bli større men den andre blir mindre liksom.
113	11:44,6 - 11:50,9	RE: Mhm. Men da, hvis vi da ser på den andre der, det er jo den. (peker på den andre figuren)

114	11:50,8 - 11:53,3	ALf: 2 gange 2 gange 11.
115	11:53,2 - 12:06,5	RE: Den er 2 gange 2 gange 11 ja. Så den er jo da, så den er jo, den blir jo litt større i x og y retning, mens z retningen, den blir jo litt mindre da.
116	12:06,5 - 12:08,2	ALf: Ja.
117	12:08,1 - 12:29,0	RE: Så, hvis en da på en måte, så , vil da. Da ser vi at overflaten blir litt større, og volumet blir litt større (peker på den delen i printen i skriptet på dataen). Så i forhold til i det du nevnte nå på, på med, dimensjonene på den 1 gange 1 gange 13.
118	12:29,0 - 12:30,0	Alf: Ja.
119	12:29,9 - 12:36,9	RE: For hvis den skulle bli 1,5 gange 1,5 gange 12.
120	12:36,9 - 12:37,6	Alf: Ja.
121	12:37,6 - 12:48,4	RE: Så begynner vi jo å nærme oss littegranne den, 2 ganger 2.
122	12:43,0 - 12:44,4	Alf: Ja.
123	12:44,4 - 13:03,6	RE: Da blir jo da, da blir jo volumet littegranne større. Vil da, eh, eh, når x er 1,5, og y er 1.5, og z er 12, vil da volumet bli større eller vil det blir mindre? Omkrets og volum.
124	13:03,5 - 13:10,2	Alf: Jeg vil jo tro at det blir mindre eller like stort egentlig.
125	13:10,1 - 13:12,3	RE: Heh, ja.
126	13:12,3 - 13:13,5	Alf: ja.
127	13:13,4 - 13:33,8	RE: Interessant, interessant, spennende. RE: Skal vi se, (blar i intervjuguiden). Ehm, skal vi se. Det var vel egentlig. egentlig stort sett det. RE: Hva, eh. Har du noe sånn kommentar til sånn opplegget generelt sett, til programmering og sånn?
128	13:33,8 - 13:37,6	Alf: Ja, jeg syntes det var et ganske bra opplegg og prøve ut og sånn.
129	13:37,6 - 13:38,6	RE: Ehe.
130	13:38,6 - 13:39,8	Alf: Jeg kunne godt hatt det i matte liksom.
131	13:39,8 - 13:50,4	RE: Ja, for det kommer jo litt sånn plops inn. Så er det jo bare 50 minutter nesten også. Så går jo. Jeg har jo skrevet ferdig en del kode som på en måte bare.
132	13:50,3 - 13:56,0	Alf: Jeg ville jo tro at hvis du hadde hatt dette i en vanlig mattetime,

		så hadde du lært den, å skrive de kodene før alt dette skjer liksom.
133	13:55,9 - 13:57,0	RE: ehe.
134	13:56,9 - 13:57,0	Alf: Ja.
135	13:57,0 - 13:58,8	RE: Det er litt sånn at man bruker litt lenger tid på det da.
136	13:58,8 - 13:59,0	Alf: Ja.
137	13:58,9 - 14:01,7	RE: Ehe, nei men det er spennende. Du må ha tusen takk.
138	14:01,7 - 14:02,0	Alf: Ja.
139	14:02,0 - 16:50,9	RE: Ehe. (Det første intervjuet er ferdig, må velge hvilken informant jeg skal ta, siden det er begrenset hvor mye tid jeg har, rekker ikke alle, samtidig som at ikke alle er tilgjengelige for intervju)
140	16:50,9 - 16:57,4	Det neste inervjuet starter. RE: Så hva syntes du om opplegget? sånn ...
141	16:57,4 - 17:01,0	Christine: Eh, det var ganske så ok. Det var litt sånn forvirrende men.
142	17:01,0 - 17:05,0	RE: Ja, hehe. For har du vært borti noe Python før eller?
143	17:05,0 - 17:07,4	Christine: Eh, under?
144	17:07,3 - 17:10,5	RE: Har du vært borti litt Python før, sånn, for det programmeringspråket som vi brukte det heter Python.
145	17:10,4 - 17:11,5	Christine: Nei.
146	17:11,4 - 17:13,5	RE: Nei. Har du vært borti noe programmering før?
147	17:13,5 - 17:15,0	Christine: Jeg tror ikke det.
148	17:15,0 - 17:17,0	RE: Nei. Ikke vært noe valgfag eller noe sånt?
149	17:17,0 - 17:17,5	Christine: Nei.
150	17:17,5 - 17:21,9	RE: Nei, neinei. Og ikke hatt noe i matematikk heller sånn på åttende eller niende eller?
151	17:21,9 - 17:25,3	Christine: Ikke som jeg kan huske i hvertfall. Men det kan jo hende jeg har vært syk eller noe sånn.
152	17:25,3 - 17:37,5	RE: Ja, men det er sånn. Ehm, skal vi se. Du har ikke vært borti noe sånn som scratch for eksempel, makekode? Det er sånn blokker, det er jo litt programmering det også.
153	17:37,4 - 17:47,9	Christine: Jeg har vært innom en ting, men det er veldig lenge siden

		da. Men det er sånn derre hvor jeg lager et eget spill på en måte.
154	17:47,8 - 17:48,9	RE: Ja.
155	17:48,9 - 17:53,6	Christine: Det er jo litt som programmering tror jeg. Men jeg tror ikke jeg gjorde noe særlig annet enn å plassere ?[blokker]?
156	17:53,5 - 17:59,0	RE: Nei, nei visst. Eh, kan du huske hva det heter for noe?
157	17:59,0 - 18:01,4	Christine: Eh, Roblox studio.
158	18:01,4 - 18:06,9	RE: Åja, ja for det er vel litt sånn derre. For det er på telefonen det er det ikke det?
159	18:06,9 - 18:11,4	Christine: Nei, det er på pc. Eller det jo sikkert på mobil også, men det er mest på pc.
160	18:11,3 - 18:27,9	RE: Ja, ja men det er det. For det er jo litt sånn programmering aktig ja. Stemmer, stemmer, det er jo litt kult da. Skal vi se. Skal vi se... RE: Ehm, sånn ut av... programmering. Hva er ditt bilde av programmering? Hva er programmering for noe?
161	18:27,8 - 18:32,6	Christine: Ehm, eh, Tall og bokstaver og ord og sånn som får ting til å gjøre ting, hehe. Bevege seg, eller.
162	18:32,5 - 18:43,8	RE: Ehe, ja visst, absolutt, spennende. Ehm ... RE: Hvordan vil du beskrive det å lære programmering sånn helt sånn generelt? Er det nyttig eller er det unyttig?
163	18:43,8 - 18:52,6	Christine: Kommer veldig ann på. Eller hvis du vil jobbe med noe av det så er det vel kanskje litt nyttig eller. Om du er interessert.
164	18:53,3 - 19:07,2	RE: Har du noe sånn. Kan du tenke deg noen bruksområder for, sånn hvor man kanskje ikke, for eksempel hvis man ikke jobbe med det kan du tenke deg noen bruksområder for å kunne bruke programmering?
165	19:07,1 - 19:10,1	Christine: Kommer ikke på noe.
166	19:10,1 - 19:14,6	RE: Nei. Ehm. Synes du det er interessant? Eller er det litt sånn uinteressant? litt sånn.
167	19:14,6 - 19:18,2	Christine: Jeg er generelt ikke så veldig interessert i det egentlig.
168	19:18,1 - 19:31,6	RE: Nei. Ehm. Så hvordan tenker du det å lære programmering knyttet til matematikk i skolen? Sånn, er det nyttig eller unyttig eller en sånn unødvendig tungvinn ting, sånn ekstra ting å lære?
169	19:31,5 - 19:42,8	Christine: Jeg vet ikke helt. Ehm, det er jo litt med matte og gjøre. For

		eksempel omkrets av gjenstander, og hvor store de er, og volum og sånn.
170	19:42,7 - 20:04,6	RE: Ja, absolutt. Ehm, synes du det er interessant eller uinteressant for eksempel hvis en skulle hatt litt matte inni, eller litt programmering inn i matematikken for eksempel. Sånn i timene, sånn noen uker med programmering. Er det interessant eller, er det litt sånn, sånn uinteressant?
171	20:04,6 - 20:09,0	Christine: Jeg synes i hvertfall sånn, som person som meg, at det er litt uinteressant men, ja.
172	20:08,9 - 20:20,9	RE: Ja, men det, er jo absolutt en ærlig sak. Ehm, så jeg tenkte vi skulle se litte grann på de... (logger inn igjen på laptopen som har stått en stund og logget av. For å få frem skriptene som vi hadde jobbet med i opplegget)
173	20:20,8 - 22:02,6	RE: Ehm, så det vi gjorde først i går, eller det aller første vi gjorde var jo å, ehm. Da brukte vi jo bare en sånn print funksjon, også bare en enkel sånn beregning av omkrets og sånn, med en funksjon. Men så fikk vi Python til å printe ut en sånn en lang lang liste. Med en hel haug med dimensjoner og overflater og volum da. Og da var det jo veldig mange tall. Sånn, eh, alt ble jo på en måte bare listet opp, så det begynner jo her med alle x-verdiene og alle liksom alle x-verdiene når de er 1 og y og z blir på en måte justert på en måte. Også får man opp liksom, ehm, alle liksom overflatene og volumene som er knyttet til de dimensjonene av boksene da. Ehm. Så det som jeg tenke littegranne på. Hva skjer med høyden når vi endrer på de andre dimensjonene? sånn som for eksempel på denne figuren (setter frem en av 3D figurene) her da, så har vi en x her også har vi en y der og en z i høyden. Så, hvis da, ehm, ehm, ehm, hvis da høyden øker hva skjer da med x og y? Hva skjer med dimensjonene ellers?
174	22:02,5 - 22:06,8	Christine: Eh, det skjer ikke noe gjør det det? eller?
175	22:06,6 - 22:37,0	RE: Ehm fordi, vi la inn en dim, vi la inn en begrensning i koden. Som gjorde sånn at eh, summen av x, og y, så x pluss y, pluss z, altså høyden, så alle de tre dimensjonene måtte bli 15 til sammen. Så hvis da, ehm, for eksempel da hvis eh, x er 1 og y er 1, da blir z 13. Men hvis disse andre endrer seg da hva skjer da med høyden?
176	22:36,9 - 22:45,1	Christine: Da må vel den bli enten høyere eller lavere for å kunne lage 15?
177	22:45,0 - 24:26,7	RE: Ja, absolutt, absolutt. Så, ehm, skal vi se. ehm. Så hvis vi da ser på slutten av, eh, det siste opplegget, det siste koden som vi kjørte, da ble det litt mer sånn oversiktlig, for den sorterte ut litt mer. Skal vi se,

		eh, der. Så det er de tre, eller de syv her da. Så det er jo alle disse som jeg har her da. Så det er sånn de forskjellige dimensjonene, og disse dimensjonene er egentlig sånn som dette da. (viser 3D figurene). Og da har vi da liksom den første her, så har vi x er lik 1 og y er lik 1, og da er z 13, så det er liksom denne her da, altså 1 gange 1 også 13. Og den koden der hadde enda en ekstra begrensning, den begrensningen var at x og y den måtte være lik, så de må være like, også den totale summen av alle de tre det ble fremdeles femten da. Så det er en ekstra begrensning. Så når vi da har, skal vi se, lagt inn disse begrensningene her da, så får vi opp eh, de dimensjonene som stemmer for det. Det som jeg lurer litte granne på hvor sikker kan vi være på at vi da har. Eller jeg kan spørre om noe anne først da. Ehm. Hvilken er det da som ble minst og hvilken av de ble størst, eh kan du huske det?
178	24:26,6 - 24:27,8	Christine: Hva som ble minst og størst?
179	24:27,8 - 24:36,4	RE: Så hvilken av disse figurene var det som hadde minste overflaten og volumen, og hvilken av de hadde det største overflaten og volumet?
180	24:36,3 - 24:46,2	Christine: Det husker jeg ikke helt ... Var de ikke alle så like da eller, hvis de var 15 til sammen?
181	24:46,2 - 25:25,0	RE: Så hvis vi ser på koden her. Så dette her er da, det er de dimensjonene, så 1 gange 1 gange 13, så 2 gange 2 gange 11, 3 gange 3 gange 9, og da får vi forskjellige overflater da, så 1 gange 1 gange 13 den har 54 og volumet er 13. Så hvis vi da ser på den listen her da. Hvilken av disse er det da som er den største av de?
182	25:24,9 - 25:30,9	Christine: Ehm. Jeg tenker jo en av de to kanskje.
183	25:30,9 - 25:45,5	RE: Mhm. Så dette er en av de, dette er en av de to av de største. Så hvilken av disse, eh, er det da?
184	25:45,4 - 25:51,8	Christine: Altså, jeg er veldig usikker, hehe.
185	25:51,8 - 26:27,1	RE: Så hvis vi da ser på denne her da, sånn, så denne her har jo da 1 gange 1, så det er jo på en måte grunnflaten. Så grunnflaten, det er x og y. Så det er 1 gange 1, også er den 13 høy som er da z da. så det er den første av de der. Så hvilken av da, er det, hvilken av disse figurene her er det da som er den nummer 2? 2 ganger 2 ganger 11, tenker du?
186	26:27,1 - 26:29,7	Christine: Ehm, denne her.
187	26:29,6 - 27:32,3	RE: mhm, yess. Så, denne her, for denn her er jo 1, så da er den 2 bred da, og 2 bred den veien, så 2 gange 2, også er den er jo 11 høy

		da, så 2 mindre enn 13. Så da har vi da, så da har vi, så den minste der da, den 1 gange 1 gange 13, den har da overflate 54 og volum 13. Så overflate er kvadratcentimeter, også har vi volumet som er kubikkcentimeter, så det er sånn centimeter i tredje. Så, så har vi den x er lik 2, og y er lik 2, og z er lik 11, den har overflate 96 og volum 44. Så da ser vi at volumet har økt litt og overflaten har økt litte granne. Så hvilken av disse her er da 3 gange 3 gange 9?
188	27:32,2 - 27:48,8	Christine:
189	27:33,6 - 27:39,5	Christine: Tenker ... Denne var 11, høy var den ikke det?
190	27:39,4 - 27:39,8	RE: Den var 11 høy, ja.
191	27:39,7 - 27:41,8	Christine: Så det er den.
192	27:41,8 - 27:50,0	RE: Mhm, ja. så da er jo, så den er jo 2 bred, og den er 1 bred, så da vet du okey, da er det, den er 3 bred også er den jo da antakeligvis da 9 høy da.
193	27:49,9 - 27:50,4	Christine: Mhm.
194	27:50,4 - 28:00,5	RE: Så da har vi den neste. Så da er det 4 gange 4. Hva er gange 7?
195	28:00,5 - 28:05,5	Christine: Ehm, den. (peker på figuren som er 4 ganger 4 ganger 7)
196	28:05,4 - 28:54,9	RE: Ja, det kan nok høres veldig riktig ut. Så den blir jo da, det blir 3 der pluss den der eneren der, så da ja jo, det blir riktig det. Så den er 4 ganger 4 ganger 7. (viser med 3D figurene). Ehm, så da ser vi at det har, sånn i fra den, eh, 3 gange 3, gange 9, som var den saken der. Den hadde jo en overflate på 126 og et volum på 81. også hadde den, den 4 gange 4 gange 7, den hadde en overflate på 144 og et volum på 112. Så ser, eh, både overflaten den stiger enda, og overflaten eller volumet stiger enda. Sånn. Så har vi da 5 gange 5 gange 5. Hvilken av de er det?
197	28:54,8 - 28:56,3	Christine: (Peker på den riktige 3D figuren som er en kube.)
198	28:56,2 - 29:11,8	RE: Jess! Så 5 gange 5 gange 5, og da har vi en overflate på 150 og volum på 125. Og 6 gange 6 gange 3?
199	29:11,8 - 29:13,1	Christine: (Peker på riktig 3D figur)
200	29:13,0 - 29:43,4	RE: Jess! Så da har vi, den er jo da 6 den veien, 6 den veien, og 3 høy. Og da er overflaten på 144, og volumet 108. Så da begynner jo overflaten og volumet å gå litt ned, det synker litt. Sånn, også da har vi da x er lik 7 og y er lik 7 og z er lik 1. Så hvilken av de må det være?
201	29:43,4 - 29:48,0	Christine: Eh, vent hva var det, det var 7 gange 7 gange 1?

202	29:47,9 - 29:48,0	RE: Ja.
203	29:48,0 - 29:51,5	Christine: Har vi ikke tatt alle bortsett fra den nå? (Peker på den siste 3D figuren)
204	29:51,4 - 29:54,7	RE: Jo, det stemmer nok. Vel.
205	29:54,7 - 29:55,4	Christine: Ja (... utydelig)
206	29:55,4 - 30:00,2	RE: Helt riktig det. Denne her må jo da være 7 den veien, og 7 den veien, og 1 i høyden da.
207	29:59,8 - 30:00,2	Christine: Ja.
208	29:59,9 - 31:27,0	RE: Så den har jo da en overflate på 126 og et volum på 49. Så da begynner jo virkelig, eh, både overflaten og volumet også synke littegranne. Men det er fremdeles ikke helt nede på, altså sånn som når vi hadde 1 gange 1 gange 13, som er den da (peker på figuren), da hadde vi et veldig lavt overflate og et veldig lavt volum. Men hvis vi ser på, eh, den 5 gange 5 gange 5. Så det var jo den (peker på figuren). Det er jo den som viser at den har det høyeste tallet, så vi hadde det høyeste overflaten, og vi har det høyeste volumet. Ehm, hvis vi da ser på disse dimensjonene ellers da. I forhold til den som på en måte, både går 4 gange 4 gange 7, den som er over (peker på programmet), det er jo da den der (setter frem figuren). Også har vi den andre som er 6 gange 6 gange 3, som er den da(setter frem figuren). Så det er disse, disse her da, disse tre her. Så her har vi den som har størst volum og størst overflate. Også her har vi litt mindre, og der har vi litt mindre volum og overflate (peker på figurene). Men hvor sikker kan vi være for at den faktisk er størst? Finnes det en, går det an også finne, eller kunne det vært en type dimensjon som kunne gjort at den hadde blitt enda litt større?
209	31:27,0 - 31:31,7	Christine: Dimensjon? Det med liksom volum og sånn?
210	31:31,6 - 31:47,2	RE: Ja, altså dimensjonene det er da, den x og y og z, så det er liksom bredden og lengden og høyden og sånn. Så finnes det en dimensjon som kunne gjort den enda større en det den er?
211	31:47,1 - 31:51,1	Christine: Så noe annet enn høyde bredde og vidde på en måte? Nei, lengde?
212	31:51,0 - 31:54,0	RE: Eh, ja, eller noe annet enn 5 gange 5 gange 5.
213	31:53,9 - 31:54,8	Christine: (Å, på alle tall? ... litt utydelig)
214	31:54,7 - 31:58,9	RE: Så lenge vi da har den begrensningen om at summen skal være 15.

215	31:58,8 - 32:03,0	Christine: Ehm, men vi har vel alle variasjonene her, har vi ikke det?
216	32:03,0 - 32:36,6	RE: Ehe, i utgangspunktet så har vi det. Men finnes det da, kunne det vært noe mulighet for å lage, eller for å endre de tallene 5 gange 5 gange 5, eh, så det er jo da 5 5 5, så 5 pluss 5 pluss 5 det er 15, så det er jo den begrensningen som vi har der. Også, men så, så er jo volumet det er jo 5 gange 5 gange 5. Er det noe sjangs for vi, kunne vi ha byttet rundt på de tallene, altså, liksom økt et tall og minket et annet tall for å få et enda større, en enda større figur? Med større volum, større overflate?
217	32:36,6 - 32:47,1	Christine: Ehm. Ikke som jeg kan komme på i hvertfall.
218	32:47,0 - 33:27,6	RE: Nei, nei, absolutt. Ehm, så, hvis vi da går den andre veien da. Så hvis vi da ser på den som har minst, så det er den 1 gange 1, eh, gange, gange 13 da. Så, er det noe mulighet for at vi kunne fått et enda lavere overflate, enda lavere areal, nei volum mener jeg? Hvis vi hadde endret på dimensjonene? For dette her er jo da denne her som er 1 gange 1 gange 13. Hadde vi kunne gjort noe med de tallene, og fremdeles hatt, eh, at det var 15 totalt, og fått et enda mindre overflate, et enda mindre volum?
219	33:27,6 - 33:31,6	Christine: Går det an å ha mindre enn 1 da, eller?
220	33:31,5 - 33:33,0	RE: Ja, det går jo det.
221	33:33,0 - 33:40,4	Christine: Kanskje sånn 0,5 gange 0,5 gange 14, blir det det?
222	33:40,3 - 33:40,9	RE: Ja.
223	33:40,9 - 33:48,7	Christine: Ja. Nei, vent, 13,5 blir det... Jo, det blir 14, jo 14.
224	33:48,6 - 34:09,4	RE: Jo, absolutt. Men da vil det kunne bli enda littegranne mindre. Som du tenker. Mhm, spennende, spennende, veldig interessant. Ehm, har du noen sånn siste, det var i grunn så og si det jeg hadde egentlig. Har du noen sånn noen andre kommentarer til programmering generelt sett eller? Eller til, hva syntes du om opplegget? Hva syntes du om geometri og programmering?
225	34:09,3 - 34:13,0	Christine: Nei, jeg vet ikke helt.
226	34:12,9 - 34:15,5	RE: Noen sånn tanker eller noe?
227	34:15,5 - 34:16,7	Christine: Eh, nei.
228	34:16,6 - 34:37,9	RE: Nei, da må du iallefall ha tusen takk. Tusen takk for hjelpen, og det er kjempebra, da skal jeg ikke holde deg noe særlig mer.

229	34:37,9 - 36:00,6	Overgang til neste intervju ...
230	36:00,6 - 36:06,9	RE: Ehm, så hva synes du om opplegget sånn generelt?
231	36:06,9 - 36:08,5	Dan: Ehm, det var greit, hmhm.
232	36:08,5 - 36:16,1	RE: Greit? ja. Eh, hva synes du, eh, liksom å, python, har du vært bort Python før eller?
233	36:16,1 - 36:17,2	Dan: Litte granne.
234	36:17,2 - 36:19,2	RE: Ja, sånn på skolen eller?
235	36:19,1 - 36:20,5	Dan: Mest hjemme på fritiden.
236	36:20,5 - 36:23,6	RE: Ja, så du er litt innteressert i programmering da? og?
237	36:23,5 - 36:23,7	Dan: Ja.
238	36:23,6 - 36:27,8	RE: Ja. Eh, sånn, eh. Hva slags ting har du programmert, sånn type?
239	36:27,7 - 36:38,7	Dan: eh, det er ikke så mye programmering jeg driver med i Python, jeg driver mer på med bare. Prøver bare å se på ting i Python mere, enn å prøve å finne ut av ting, enn å... ja.
240	36:38,7 - 36:52,1	RE: Ja altså liksom på en måte finne kode på nettet og prøve å finne ut av hvordan de fungerer på en måte. Ja, spennende. Ehm, har du hatt noe valgfag i programmering, eller noe sånn?
241	36:52,0 - 36:56,2	Dan: Ja, det er det jeg har i valgfag.
242	36:56,2 - 37:00,5	RE: Åja, har du det nå? Åja. Ja, hva gjør dere der for noe?
243	37:00,5 - 37:06,8	Dan: Vi driver på med Javascript, og HTML og sånne ting nå, akkurat nå.
244	37:06,8 - 37:14,4	RE: Ja. Javisst. Ja for det blir jo litt sånn, da har du jo, da er du jo litt sånn vant til den tekstbaserte kodingen og sånne ting også da. Eh, har du vært borti sånn Scratch og sånne ting også da?
245	37:19,3 - 37:23,3	RE: Ja. Ehm, MakeCode? Har du hørt om det?
246	37:19,5 - 37:19,6	Dan: Ja.
247	37:23,3 - 37:24,0	Dan: Ehm, ikke ...
248	37:23,9 - 37:26,1	RE: Det er sånn til Microbit og sånn ting.
249	37:26,0 - 37:29,6	Dan: Ehm, tror vi var borti det med MicroBit ja.
250	37:29,6 - 37:34,5	RE: Mhm, ja. Du har ikke hørt om Logo for eksempel?

251	37:34,5 - 37:34,7	Dan: (nikker nei.)
252	37:34,6 - 37:48,6	RE: Nei, også HTML og Javascript nevnte vi. Ehm, skal vi se. Ehm, så hvordan syntes du det er, liksom Javascript og HTML, hvordan synes du de er i forhold til Python?
253	37:48,5 - 37:56,5	Dan: Python er mye mer komplisert å lære, enn det Javascript og det.
254	37:56,5 - 37:59,0	RE: Ja, du synes Javascript er lettere?
255	37:59,0 - 37:59,7	Dan: Mhm, ja.
256	37:59,7 - 38:12,4	RE: Absolutt, absolutt. Hvis du skal beskrive programmering, sånn helt sånn generelt, liksom sånn bilde av hva er programmering?
257	38:12,3 - 38:17,6	Dan: Programmering generelt? Ehm, mest sånn at du koder nettsider eller spill eller noe.
258	38:17,5 - 38:24,8	RE: Ja, absolutt. Sånn, eh, i forbindelse med dataprogrammer, og spill og sånn hvordan brukes det der på en måte?
259	38:24,8 - 38:32,1	Dan: Eh, det brukes for eksempel sånn som Windows er jo kodet. Egentlig all teknologi som vi bruker er kodet.
260	38:32,1 - 38:42,6	RE: Ja, absolutt. Det er jo linjene med koder som er overalt. Ehm, hvordan vil du beskrive det å lære programmering sånn generelt? Er det nyttig? Er det unyttig?
261	38:42,5 - 39:24,4	Dan: Det er nyttig hvis du skal drive med en jobb som for eksempel du bruker det i. Hvis du for eksempel skal jobbe med datamaskiner så må du gå inn i selve programvaren og prøve å finne ut av det. Da er det jo nyttig å bruke, og ha lært seg det. Generelt så hvis det er noe du holder på med på fritiden, ja. Men hvis ikke det er noe du holder på med på fritiden så er det, ja, så er det ikke det du vil lære hvis du ikke driver med det på fritiden.
262	39:12,1 - 39:22,4	RE: Nei, absolutt. Nei, det er et godt poeng. Ehm, kan du tenke deg noen bruksområder? Sånn til, det er kanskje litt det du pratet om.
263	39:22,4 - 39:50,1	Dan: Eh, programmering kan jo bli brukt til mye forskjellig. For eksempel hvis du vil kode en nettside for eksempel, da kan du jo bruke det. Men det er jo programmer du også kan bruke, som er mye lettere å bruke til å lage nettsider, men hvis du bare vil sitte å kode for å lage nettside, istedenfor å bruke de nettside programmene. Så kan du jo designe det mer som du vil da, hvis du koder den.
264	39:50,1 - 39:52,9	RE: Ja, absolutt, du har mere ..?

265	39:52,9 - 39:54,7	Dan: Du har mer kontroll over nett...
266	39:54,6 - 40:09,7	RE: Absolutt. Det er et godt poeng. Eh, hvordan vil du beskrive det å lære programmering, sånn tilknyttet matematikk i skolen? Er det praktisk, eller er det nyttig, unyttig?
267	40:09,6 - 40:20,2	Dan: Nei, siden hvis du lærer litt programmering fra før så er det jo nyttig å kunne bruke det til å lære matte også. For da lærer du jo to ting på samme tid.
268	40:20,2 - 40:27,6	RE: Ja, absolutt, absolutt. Ehm, kan du tenke deg noen bruksområder i matematikk?
269	40:27,5 - 40:42,8	Dan: For eksempel som å lage en kalkulator. Eller sånn i allefall sånn bruke det som en kalkulator ting. Kan for eksempel kode en Pytagoras kalkulator eller noe sånn ting.
270	40:42,8 - 40:51,3	RE: Javisst, til sånne typer formler eller sånne ting? Synes du det er interessant eller uinteressant?
271	40:51,2 - 40:53,4	Dan: Det er interessant med koding.
272	40:53,4 - 40:56,0	RE: Ja, sånn i matematikken også sånn i?
273	40:56,0 - 40:57,1	Dan: Ja.
274	40:57,0 - 41:09,1	RE: Ja, eheh. Ehm, kan du tenke deg noen sånne positive sider med programmering i matematikken? Og noen negative sider?
275	41:09,0 - 41:34,3	Dan: Eh, at du må lære deg det før du begynner med det. Det er jo negativt at du må prøve å lære deg noe nytt før du skal begynne å lære det du skulle. For eksempel hvis du skulle lære Pytagoras via programmering, så kan du, så må du lære deg programmering før du begynner med pytagorasen. Det er en ulempe da, siden da bruker du lenger tid på det.
276	41:34,2 - 42:15,9	RE: ehe, ja absolutt, interessant, interessant. Ehm, skal vi se. Så det aller første vi gjorde hvis vi kan prate litt om det opplegget vi gjorde i går. (Starter opp datamaskinen igjen og gjør klart på skjermen) ... Så det aller første vi gjorde det var jo bare å printe ut et eksempel på liksom areal, eller volum tror jeg vi gjorde. Også lagde vi etterhvert sånn at vi fikk printet ut en hel liste med alle disse forskjellige dimensjonene da.
277	42:15,8 - 42:22,6	Dan: Ja, og tall og sånn. Masse tall, overflate, dimensjoner, og volum.
278	42:20,6 - 42:40,1	RE: Ehm. Så, ehm, når vi da hadde, eh, disse her da. Sånn, hva kan du si om hvis vi da, så vi hadde jo en begrensning, og den begrensningen var jo det at liksom den totale summen av de skulle jo bli 15. Av x og

		y og z, som da er liksom lengden og bredden og høyden da.
279	42:22,8 - 42:22,9	
280	42:40,0 - 42:44,0	Dan: Selv om vi kan jo sette på mer, kan gjøre det høyere.
281	42:43,9 - 43:01,6	RE: Ja. Hvis man hadde satt 20 for eksempel som begrensning, så hadde man kunne fått enda større. Eh, men hvis man da øker på lengden da for eksempel, hva må da skje med bredden og høyden da?
282	43:01,6 - 43:13,4	Dan: Eh, hvis du øker lengden, så blir den jo fortsatt, eh, lengde, sånn derre høyden og det det samme men det blir bare lenger.
283	43:13,4 - 44:12,6	RE: Ehe. Så, skal vi se, må bare følge med på tiden. Ok, vi har sånn 5 minutt. Ehm, skal vi se. Så det andre vi gjorde etter vi hadde, eller det siste vi gjorde før klokken ringte. Så lagde vi den listen hvor vi hadde dimensjonene x, y, og z. Også, men da var, en ekstra begrensning, vi hadde fremdeles den begrensningen om at totalsummen av x, y, og z, den skulle være 15. Men, x, og y skulle også være like da. Så da står det jo 1 gange 1 er 13, og 2 gange 2 er 11. Så alle de er jo printet ut der da, så det er disse her (Peker på programmet på datamaskinen). Så hvilken av disse figurene her er det som korresponderer til de dimensjonene?
284	44:12,5 - 44:18,0	Dan: Det kommer helt ann på hvilken av de du skal ha.
285	44:17,9 - 44:23,5	RE: Ehe, men hvis vi skulle begynt på toppen. 1 gange 1 gange 13, hvilken av disse er 1 gange 1 gange 13?
286	44:23,4 - 44:23,7	Dan: Den (peker ut riktig figur)
287	44:23,7 - 44:26,0	RE: Den. Og 2 gange 2 gange 11?
288	44:25,9 - 44:27,4	Dan: Den (peker ut riktig figur)
289	44:27,3 - 44:29,4	RE: Yes. Og 3 gange 3 gange 9?
290	44:29,4 - 44:30,0	Dan: Den (peker ut riktig figur)
291	44:29,9 - 44:32,9	RE: Aha. Og 4 gange 4 gange 7?
292	44:32,9 - 44:37,5	Dan: 4 gange 4 gange 7, eh, den, der. (peker ut riktig figur)
293	44:37,4 - 44:41,1	RE: Aha, aha, ja. Og 5 gange 5 gange 5?
294	44:41,1 - 44:42,1	Dan: Den. (peker ut riktig figur)
295	44:42,0 - 44:44,7	RE: Aha. Også 6 gange 6 gange 3?
296	44:44,7 - 44:45,8	Dan: Den. (peker ut riktig figur)

297	44:45,7 - 44:49,1	RE: Aha. Også 7 gange 7 gange 1?
298	44:49,1 - 44:49,9	Dan: Den (peker ut riktig figur)
299	44:49,9 - 45:15,4	RE: Aha, ja, kult. Så, eh. Så har vi jo da funnet ut at, så vi har den største, den fant vi ut var 5 gange 5 gange 5. Så da var jo overflaten 150 og volumet var 125. Så, hvor sikker kan vi være på det faktisk er den største innenfor de begrensningene som vi har da?
300	45:15,3 - 45:20,1	Dan: Ehm, fordi de tre tallene blir jo 15 til sammen.
301	45:20,0 - 45:34,9	RE: Mhm. Så de holder den begrensningen der. Men hadde vi kunne endret på noen av de dimensjonene. For eksempel, eh kunne vi ha. Hvis vi hadde økt x litte granne, og satt de andre ned littegranne. Ville vi da, med en kombinasjon av det kunne vi ha fått, eh ...
302	45:34,9 - 45:41,3	Dan: Vi kunne jo ha fått noe annet da. Hvis vi hadde økt litt på den ene, og, ja, latt de andre stå.
303	45:41,3 - 45:44,3	RE: Mhm, men kunne vi ha fått det enda større enn det vi har fått?
304	45:44,2 - 45:49,4	Dan: Ehm, det må du egentlig se på om du, hvilke tall du endrer på.
305	45:49,3 - 46:23,5	RE: Mhm. For hvis vi skal ha den begrensningen, så hvis x og y skal være like. Eh, hvis vi da hadde for eksempel satt, eh, 4,5 gange 4,5, pluss, eh, eller gange 6 da. Ville det kunne blitt større eller ville det blitt mindre? Hvis vi sammenligner det med for eksempel da, eh, den, eh, 4 gange 4 gange 7, eh, og, så hvis vi da har, skal vi se. (finner frem figurene som hører til de som er over og under på listen). Vi har jo da, det er den som er størst, også har vi, nei hvilken var det som var, det var den som var et hakk under...
306	46:23,4 - 46:24,4	Dan: (følger med) ...ja...
307	46:24,4 - 47:05,4	RE: ... og den var et hakk over, var det ikke det. (de tre figurene står nå forran på bordet) Så, hvis vi da, eh, så den der er jo da, eh, 6 gange 6 gange 3, så det er den (knytter figurene til skjermen). Og den er 4 gange 4 gange 7, så 7 høy også 4 i bredden. Så, hvis da hadde, hvis vi da hadde endret på dimensjonene. Er det et mellompunkt mellom 5 gange 5 gange 5, og 4 gange 4 gange 7? Eller mellom da, eh, 6 gange 6 gange 3, og 5 gange 5 gange 5? Er det en mellomting mellom de, som kunne blitt større?
308	47:05,4 - 47:14,8	Dan: Ehm, det hvis du kombinere litt av de, så kan de jo være en mellomting. Hvis du tar for eksempel 5 gange 5 gange 7, da er det en mell, da er det jo en blanding av begge.
309	47:14,8 - 47:19,1	RE: Men hadde de kunne blitt, hadde den kun blitt enda større?

310	47:19,0 - 47:43,7	Dan: Det hadde blitt høyere. Eh, kanskje hadde hatt litt mer volum også. Hvis den hadde, siden hvis den der hadde blitt et hakk høyere, siden da tror, den er to hakk høyere så er den jo på høyde med den. (viser med figurene). Og, da får den jo det volumet som det skal.
311	47:43,7 - 47:49,4	RE: Mhm, ok. Skal vi se, to minutt igjen så ringer det.
312	47:49,4 - 47:52,3	Dan: Se det, nå begynner folk og gå.
313	47:52,2 - 48:07,5	RE: Ja, skal ikke holde deg så veldig mye lenger. Så, det er jo den minste som vi fant ut. Den 1 gange 1 gange 13. Og det var jo den minste overflaten da. Hadde vi kunne laget, kunne vi hatt noen dimensjoner som kunne fått den enda mindre, altså mindre volum, mindre overflate?
314	48:07,4 - 48:10,5	Dan: Eh, hatt mindre, lavere høyde.
315	48:10,5 - 48:50,9	RE: Mhm, lavere høyde så da litt lavere der, men ville den da kunne, ville da overflaten da blitt større eller blir den mindre? (klokka ringer). Nei det var visst det vi rakk. Men, tusen takk, det var helt topp. (Ferdig, gikk tomt for tid)