

# INTRODUCTION OF COMPUTERS IN NORWEGIAN SCHOOL MATHEMATICS

Herman Ruge

*Formerly of The Central Institute for Industrial Research, and Forsøks-gymnaset Oslo,  
Daas gate 16, 0259 Oslo; mruge@online.no*

**Abstract:** “Forsøks-gymnaset in Oslo” was an experiment in School Democracy that started classes in automatic data processing in the fall of 1968 as a part of school mathematics in the 11<sup>th</sup> year. We started simple programming in FORTRAN on punched cards, off line. Later we had a teletype terminal with paper tape, changed to the BASIC language, and ran programs online by telephone to a distant computer. We also designed the logic simulator “Benjamin”, with four “And”, “Or”, and “Not” elements, battery and lamps, which could be connected in logical networks to be studied. In 1970, we arranged two two-day seminars for teachers during the Christmas and the summer holidays and presented the whole course. We did this eight times and helped to qualify many teachers. The first Examen Artium in Data was organised in June 1970.

**Key words:** Mathematics, logic, Boole, forsøks-gymnaset, Norway

## 1. INTRODUCTION

In the fall of 1967 Forsøks-gymnaset i Oslo started by young students who wanted to create a democratic and experimenting secondary school. The school lacked a teacher in mathematics, and I applied because I liked their plans. In addition, I had worked with technical cybernetics for ten years, and wanted to try automatic data processing (ADP) as a school subject. In April 1968, we applied for permission to use this in the mathematics curriculum at the humanistic and science classes, arguing: “In ten years, when our students have entered the work force, they will need

this.” The Ministry of Education declined the application, but we wanted to go on, and so the Ministry made a committee.

What should be the contents of this new school subject? In 1968, the students knew almost nothing about computers. So we started with simple historic calculators—the Abacus (counting frame), Pascal’s addition machine (as in the kilometer counter in a car), Leibniz’ multiplier (as in the cash register in the shops), and Hollerith’s punch card machine with electric switch logics, designed for the US Census in 1890. We discussed automata and their programs, like music boxes and gramophones. The traditional chocolate vendor machines also had to ask questions and receive answers: “Is there a coin in the slit? Is it of the correct type? Is there any chocolate in the machine?”

These programs were illustrated with flow-charts. We also discussed artificial languages, as in traffic control, with red, green, and yellow lights. This language has an alphabet with a number of symbols with accepted meaning (semantics) and rules for sequences (syntax). How could we improve this? Another discussion was the use of binary numbers, and other bases.

## 2. PROGRAMMING IN FORTRAN

We planned the data subject as a part of school mathematics. We felt that “Modern mathematics” was a mistake, which made mathematics less relevant. Programming in a high-level language like FORTRAN could refresh the traditional mathematics, enabling the students to see it in a new perspective. For example, a variable is a quantity with a name such as  $x$ ,  $y$ ,  $A$ ,  $A2$  and an address in the computers memory. The number in the memory cell is the current value of the variable. As another example, in the hierarchy of arithmetic such as  $1+2 \times 3 = 7$  or  $(1+2) \times 3 = 9$ , classical school mathematics suggest to dissolve parentheses; however, the parenthesis demands “do me first.”

FORTRAN 1968 was somewhat cumbersome, like the distinction between integers and fractions. Input/output-operations were also intricate. The problems had to be simple. Indicated variables such as  $A(x)$  were considered an upper limit. Our method to illustrate and check a simple FORTRAN program was to draw a square on a paper for each of the variables as pictures of the memory cells, and draw a bigger square for the printout. Students should pretend to be the computer and follow the program strictly. They should write numbers into the memory cells and produce the output according to the program. A new number would wipe out the old.

“Playing the computer” helped the students to program and it was a good method for faultfinding. Correcting pupils’ programs can be a heavy burden for the teacher so we had to teach them to help themselves. Some typical exercises are as follows:

Write a program that

- Prints out a function table for  $y = 2x$  with  $x$  varying from 0 to 5
- Reads in ten numbers and prints out the highest
- Prints out the numbers in a sequence, starting with the lowest
- Reads in the numbers  $A$ ,  $B$ , and  $C$  and prints out the solution to the equation  $Ax^2 + Bx + C = 0$

Few traditional math problems were suited for our course. However, we had to find and make new ones that could justify the use of a big computer while being understandable and interesting.

We found such a type in the stepwise “simulation of processes”. The simplest was the “dog curve” that stated: A dog plays in a field while its owner walks along the  $x$ -axis. Suddenly he calls the dog, which starts to run towards him. The dog is a little foolish; it looks at him each second and runs in his direction for the next second. Compute the curves that the dog and the man will follow! This is a classical problem in differential equations. In a digital computer, we could approximate it by finite differences: Define the direction of the dog towards the man, move the dog one second forward, move the man one second forward, and so on. Other examples were the Volterra’s interaction between rabbits and foxes on a desert island, and the development of a contagious epidemic, or a news item - the logistic curve. The elementary course in FORTRAN, Algol, or BASIC gave the students a new understanding of traditional algebra. The University gave us access to their large computer CDC 3300. A couple of students went there, punched the programs on cards, and received – often the next day – the printouts on large sheets. We named the first textbook *ADB for NFG 1968*.

### 3. BENJAMIN

We also wanted to teach the students something about how electronic computers worked. Statements such as “It is not necessary to know how the TV works to watch a TV program!” often criticized this thought. However, process knowledge was an important part of school curricula. Therefore, we designed an elementary electronic computer to illustrate some principles. In the language classes, we used a mathematics book with some classical logics, about statements and syllogisms: “According to Aristotle, a syllogism is a sequence of thoughts where certain things are stated and other things

follow with necessity from the first ones” (implications). Then followed more advanced logic, such as two statements joined together, some set theory, and the use of different notation systems emphasizing binary numbers. It was an engaging book, but difficult to teach. However, each Tuesday I taught much of the same theory to university physics students. The textbook we used there was Brusdal’s *Electronic Computers*. We could use this simple mathematics for very different purposes.

In the 1850s, the Irish mathematician George Boole developed a method for computing logical conclusions – a “calculation of statements” – on rational thinking. The statements could have only two values, True or False, and the operations were Or, And, and Not. This “Boolean algebra” remained for a long time pure theory. However, in 1938 the American student Claude Shannon discovered that one could use it for electric switch networks, as in telephone exchanges and computers. Two switches coupled in a series constituted an And, and in parallel an Or. Boole’s algebra thus became a basic theory for digital computers. Circuit networks of And, Or, and Not made it possible to calculate with numbers and sort data written in binary systems.

In 1969, logical circuits were mass-produced as cheap microelectronics. We used many of them at SI. So we soldered together each of the three circuit types: And, Or, and Not on a A4 sized sheet, four of each type. We could couple these twelve circuits in a network, with separate conductors, contact pegs, switches, and signal lamps. This resulted in a simple electronic computer for school use. We called it a “logical simulator” and named it “Benjamin”. We developed a number of training exercises and started to use Benjamin in the teaching of mathematics and computer science. Some friends had established an electronic workshop and we gave them the production rights. They gave Benjamin an attractive design, with gold plugs on a white sheet in a frame of oak tree. They produced several hundred Benjamin simulators for secondary and vocational schools.

Benjamin realized basic mathematics of different origins, applications, and names: Boolean algebra, Statement calculus or Mathematical logics, Set theory, Binary mathematics. The technical terms used varied accordingly: The functions And and Or were also called “Boolean multiplication and addition”; in logic, they are “conjunction and disjunction” and in set theory they are “cut and union”. The symbols varied accordingly.

In the summer of 1970, we wrote the textbook *Exercises in electronic logics*, with networks for statements like “If you wish, and I have the time, and it is not raining, then we will go for a walk.” These exercises included alarm circuits, Boolean algebra, set theory and binary computing, flip-flop memory, all elements in computers. We designed the book for teaching with laboratory work, with exercises both in the mathematics lessons and humanistic classes and in computer science in the science classes. They also

used it in junior high schools and vocational schools. Some students discovered for the first time how coupling of conductors could make lamps light up. For others, Benjamin served as an introduction to advanced electronics. Anne was a student who later became an electrician at Spigerverket in Oslo. She told me about a machine which should become automated, with conductors leading to a mysterious box. "But suddenly I understood that this was only a large Benjamin, and then everything became clear."

In the fall of 1969, we bought a Teletype teleprinter (which we named "Katja") and a modem. It became a terminal for a Bull computer based in Stockholm. We changed the language from FORTRAN to BASIC. BASIC was a time-sharing computing language developed by the teachers Kemeny and Kurtz in 1964 and was better suited for school use. Now the students could write their programs on paper tape, ring up the computer, and read in the program at high speed. This was a big pedagogical improvement. Students became less dependent on the teacher. Programming and computing increased. However, the price per second was high. We made a new textbook called *Programming Course in Basic: Language elements and program structures*, Oslo 1971. The length of the course was 70 school-hours. The first official school examination in this subject took place in June 1970.

#### **4. COMPUTER SEMINAR FOR TEACHERS**

Around Christmas of 1969, we invited mathematic teachers from all over Norway to a computer seminar 2-3 January 1970 with the same curriculum as for the students, in a compressed version. Jan Wibe from Trondheim and Kjell Bjarne Henden from Sogndal were the first two to register. They were active teachers in their respective regions and contributed much to the further development. Seminarer was a success. It was repeated in June and then every six months, in all eight seminars. Over a hundred teachers participated. Many of them had computer experience, but needed the discussions for the school-oriented subjects. We made the seminars as a "dugnad", a voluntary effort by teachers, students, and participants. They were stimulating for us all and actually started this subject in many schools in Norway.

In March 1970, we presented our plans and experiences to the OECD-conference on Computer Education in Paris, and later to the Inter-Nordic seminar in Helsinki. FGO was the only Nordic school with this kind of education. The French had ambitious plans, and showed a great interest in our ideas.

In Norway, strong local interest prevailed in the gymnasiums, but also a lack of funding. In 1984, fourteen years later, the School Ministry made an all-out effort to introduce computer education in all schools. At that time, over 30,000 students had already acquired a basic data competence at school and had become part of a work force that needed just that knowledge of computers.

## **REFERENCES**

- Ruge, Herman, "Automatisk databehandling i skolen" *Den høgre skolen*, No 3, 1969  
Ruge, Herman, *Øvinger i elektronisk logikk*, Forsøksgymnaset, Oslo 1970  
Ruge, Herman, *Programmeringskurs i Basic*, Forsøksgymnaset, Oslo 1971

### Author's Note

See also: Steinar Thorvaldsen and Jan Wibe, "The First IT-Course in Norwegian Upper Secondary Education", *Proceedings from the HINC1 conference*.