99A014I

Project Number: DZD 9902 -51

99A014I

"From Abacus to Microchip – How a Computer Functions.": A Workshop for a German Children's Museum

An Interactive Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Nila E. Almstrom [illegible]

Nila E. Almstrom And Malcolm H. Beaulieu

Date: October 14, 1999

Approved:

David Dollenmayer

Professor David B. Dollenmayer, Major Advisor

1. Computers
2. Workshop
3. Germany

Table of Contents

Abstract

The Kinder-Akademie Fulda, a children's museum in Germany, invited the project team to teach a workshop on computers for 9 to 13-year-olds. We entitled the workshop "From Abacus to Microchip, How a Computer Functions". The topics covered were historical number systems and computing machines, binary numbers, the history of computer programming and data storage, how a printer operates, flowcharts, and how to write a simple program in C++. The workshop was entirely planned and taught in German by the project team.

I. Introduction

During E-Term 1999 two WPI students (Nila Almstrom and Malcolm Beaulieu) were requested to teach one in a series of workshops at a Children's Museum in Fulda, Germany. The theme of these workshops was numbers. In order to remain close to this theme and at the same time bring something new to the museum, it was decided to base our workshop on computers and how they function. Computers are entirely based on numbers and a great strength at WPI. Some background research was performed and a workshop then preliminarily planned. Once we arrived in Germany, a translation was done of our intended workshop. A presentation was given to the staff members of the museum in order that they would be able to have an idea of what we planned to do. Not only was this achieved but the presentation also sparked interest in our workshop and some of the staff wished to attend. Computers are not as widely used in Germany and knowledge of them at the Kinder-Akadamie Fulda (KAF) was minimal. This presented an opportunity for them to learn how the machine actually functioned. The period between the presentation and the workshop was spent in much preparation and research. A museum called the Konrad Zuse Museum was visited in order to get a better understanding of some of the German history of computers. German literature was also read in order to get more ideas of possible activities for the children in this workshop to do. The greatest advantage that was available to us was the ability to sit in on other workshops. This provided insight not only into how, and how much, information other staff was presenting, but also, how well the children were learning or even enjoying the material. With this firsthand knowledge, the workshop was then refined to better fit in with the others that were being taught and what reactions could be expected from the children. The final arrangement of our workshop was two 2.5 hour sections per day for three days. During the first day, everyone got to know each other. Time was spent going over historical numbering systems and which led into the binary number system, which is what computers use. A short game was played to enforce the operation of the various numbering systems. The second segment was then used to briefly explain some history of computers and data storage then begin construction of a "clothespin computer," or simple punch card reader. The second day entailed finishing the "clothespin computer" and showing various devices which people have used to make math easier, beginning with the abacus and leading up to the microchip. Microscopes were then used to view the minute details of microchips. They were not only seen in the little black ceramic cases that we are so accustomed to, but in the wafer form of which they are made. We spent the afternoon at a local computer laboratory and a very simple introduction to C++ Programming. The third and final day was spent explaining how printers and computers communicated and how flowcharts worked. The afternoon was spent at the computer lab where the children received an outline of what a desired program should do and then wrote the necessary code to do so. After returning back to the museum, the students filled out a questionnaire and the workshop was then completed.

II. Statement of task

Our task was to design, plan, and teach a workshop for children on how computers function. This workshop was one of eleven workshops on the theme of numbers presented in the summer of 1999 by the Kinder-Akademie Fulda (KAF) in Germany. Thirteen children between the ages of nine and thirteen attended three all-day sessions. We were asked to make the workshop as activity-oriented as possible and to have the children build or make something that could be displayed in the museum. The major goals of our workshop were the following:

- Give the children some background information about computer history and the technology leading up to modern computers and programming.
- Teach the binary code and why a computer uses it, how it works, and how to translate it into the decimal system.
- Build gadgets and organize activities that simulate the way a computer works, now or historically.
- See what is inside the magic box that makes it function.
- Involve them intensively in computer programming so that they will see the sort of work involved, try it out, and go away with a better understanding of how people use computers. They will see that simply using a computer program is much different from writing one. Our goal was not to create a deep understanding, but rather to introduce them and possibly get them interested in an aspect of computers that they may not have known about.

III. Background research

As preparation for our workshop, we reviewed as much literature on computers as possible before leaving for Germany. We looked at many children's books as well as encyclopedias and technical magazines to find ideas on how to present technical computer information to young children. We consulted with Edmund Hazzard, an educational researcher at Tufts University, who was well versed in the use of a computer program called LOGO, which was designed to introduce young children to computer programming. Even though we decided not to use LOGO as a programming language for our workshop, we learned about presenting the concepts of programming to children and used this information in developing our own way to teach some very elementary C++. We found activities and ideas for what to teach in the children's books. Here we will show how our ideas and ones we found in books developed into components of our workshop.

The literature review focused mainly on the history of numbers, computers and computing machines.

Human societies have had various ways of writing down numbers. The Romans and the Greeks used letters to write numbers. The Greeks used the letter alpha to represent one and the second letter to represent the number two and so on. The Romans used the numeral system that is still in use today that uses letters to represent numbers.

Today we use the base ten system, but not all cultures did. The ancient Mesopotamians used sixty as their base for a number system. The Babylonians and the Mayans used systems that used only two characters. The number four is represented by four similar marks arranged in a cluster in both systems. The Babylonians used little 'v' shaped characters to represent numbers by drawing that many vs'. In order for large numbers to be expressed more concisely, one big, sideways 'V represented the number ten.

Information on the history of counting systems was found in the *Encyclopedia Americana*, but the presentation of the information in a German book entitled *Zahlen, Spiralen und magishe Quadrate* (Numbers, Spirals, and Magical Squares) was more suited for children and aided in the translation of this information to German. Many of the topics researched during the PQP were researched again in Germany because it was easier than translating the material. This book also explained that the Mayan base five system was based on hands and feet because they have five appendages each. Unlike the encyclopedia, the book used many pictures to show examples of different counting systems. We used this concept in our workshop and used the blackboard to illustrate our points.

The idea of the quantity zero was discovered after these numbers systems were used. The zero is particularly important in the world of computers because computers use the binary system. The binary system is base two and it is the simplest way to communicate a number because it uses only two symbols, 0 and 1.

People have always tried to make their lives easier by being able to count and do calculations faster and with fewer errors. Children use their fingers to count. The Chinese invented the abacus, which were originally just lines drawn in the sand with small stones positioned appropriately on the lines. Eventually the abacus evolved into a frame of bars with beads that could slide rule back and forth. Surprisingly, abacuses are still commonly used in some parts of the world.

In 1614, John Napier wrote a paper on logarithms and showed how they can be used to multiply and divide numbers quickly and easily. Eventually logarithms were written in tables to save time making calculations. In 1632, William Oughtred invented the slide rule to save even more time when doing calculations.

The slide rule and abacus have been made obsolete by the invention and widespread use of the calculator and computer but the history of these devices give a student perspective on how computers came into existence.

Isaac Asimov's book, *How Did We Find Out About Computers,* explained how computers were invented. It was important to our workshop that we show the children that computers were not invented by one person, or even in one iteration. Many computing-machine inventions preceded what we consider the modern computer and they were important discoveries that influenced the modern design.

Mathematik (Mathematics) by Carol Vorderman presented similar information and was extremely helpful to us because it helped present the information in German in a format that children would appreciate. This book presented minimal information in the form of text. The book suggested activities for children through which they learned about math. One of the activities was to build a computer of your own out of a cardboard cereal box. The purpose of this computer was to store information about several people on cards and sort the cards according to this information. All of the stored information was the result of a survey of yes or no questions. The computer could only perform this single function and but it demonstrated the concept of binary logic used by a computer. Since we wanted our workshop to be as activity-oriented as possible, we thought about building our own computer.

Our own homemade computer was quite different from the one suggested in Carol Vorderman's book, because the idea evolved into a punch card reader. This was inspired by the following information found in *Inventors & Inventions: Computers* by David Wright.

In 1880, the United States census took so long to complete that by the time it was finished, it was time for the 1890 census. Herman Hollerith, who worked in the US census office, invented a machine that used punch cards to gather data for the census and add it up using a machine that read the punch cards. Hollerith was not the first to invent punch cards. He got the idea from a French weaver, who made the fabric we still know today as Jacquard, on a weaving loom controlled by punch cards. In 1896, Hollerith founded the Tabulating Machine Company that eventually became the International Business Machines Corporation, or IBM.

Mathematik showed how flowcharts could be written to represent the steps in any task. The task in the book that was diagrammed in a flowchart was much too complicated for the purpose of our workshop, but it showed us that we could teach the children in our workshop about flowcharts for simple tasks, and use a flowchart to explain to them how a computer program should be written.

Flow charts are a logical way of approaching a task step by step. Computers work logically, solving problems by breaking them down into very simple steps. A flow chart is a way of presenting these steps so that people can see the logical paths. In the flowcharts are questions. The answers to the questions determine a path that leads to an end result. A flow chart can be made for any activity such as taking a bath or writing a computer program to solve a problem.

The best way to learn about computers and how they work is by using them. Therefore most books have several activities that involve using a computer to discover how it works. One activity to stimulate the children's interest was writing a computer program. Simply reading a program will not tell an amateur computer user how a computer functions, but understanding what each command means, and then seeing what a computer does when it executes these commands in a program is a good place to begin. A BASIC language program to calculate your weight on other planets was given in the project section of *All About Computers* by Jean Atelsek. The instructions in the book suggested that the student simply type it into the computer and try it and see what happens. Then the student can make modifications to the program and see what effect the changes have. We found that writing a program to calculate the weight of a person on another planet was a simple problem for our workshop to solve and would show how a program controls a computer.

IV. On-Site Work Report

A. Site Description

1. Fulda, Germany

Fulda, Germany is a city of 62,780 inhabitants located in the German state of Hessen. It is about one hour northeast of Frankfurt. The oldest surviving church in Germany is located there in addition to a large Baroque-style cathedral. Fulda is the site of the German conference of Catholic Bishops. Fulda has a technical school much like an American technical college or college of applied sciences and engineering where students study subjects such as nutrition, computer science, hotel management, electrical engineering, mechanical engineering, nursing, and economics. Our project had little to do with the school except that we lived in their dorm apartments with their students.

In Germany museums are not usually privately owned or run. The government uses taxpayers' money to operate museums. Museums in the US often began as privately owned collections that grew into non-profit institutions, often with some government funding. KAF is unique in Germany because it was funded privately in the beginning and now covers all its costs using admission fees, donations, and some government grants for individual exhibits.

2. The Kinder-Akademie Fulda

After a trip to the Boston Children's museum, Frau Helen Bonzel was inspired to found a similar children's museum of her own in Germany. She realized her dream in 1994 when she used private funds to renovate a former factory and found the Kinder-Akademie Fulda, or KAF. The museum houses art and science exhibits specifically for children and plenty of educational toys as well. This type of museum is also rare in Germany.

The museum is purposefully arranged so that visitors may wander from exhibit to exhibit at their leisure and spend time on what interests them most. Minimal direction and explanations accompany each display in order to allow the children to experiment and understand the phenomena through their own discovery.

An example of the KAF learning philosophy is the walk-through heart where children and parents learn about the physiology of the heart. The children and parents sit on cushions that are model red-blood cells and listen to a lecture that is highly interactive, actually a series of demonstrations about the heart. The heart is a huge model, large enough that a person can climb through it during parts of the lecture. The participants listen to their own resting heartbeats with a stethoscope and discover what happens to their heartbeat after they run around for a few minutes.

The museum also has an Inventors' Club that meets after school in the winter. The young inventors experiment with materials such as pieces of wood, fabric, nails, screws, glue, paint, flashlight bulbs, wire, batteries, electric buzzers, and switches to create their own dynamic masterpieces. The program stresses learning in an entertaining and fun way rather than in a formal setting.

In the summertime, the KAF holds workshops based on a common theme for children of all ages with many different interests. In the summer of 1999, there were eleven different workshops all having something to do with numbers or mathematics. There was something for almost every possible interest, since the workshops included theatre production, gadget inventing and building, art classes, explorations of the mysteries of Pascal's triangle, geometry, and even computer programming.

B. Project Preparation and Execution

1. Preparation After PQP

Before we left for Germany, we prepared what we thought we would teach in our workshop in English and brought with us anything we thought we would not be able to find readily there, such as old computer parts and information in English. As soon as we arrived, we went on a tour of the museum and met with Frau König, assistant director of the museum, to discuss our plans. She read our proposal and list of requisite equipment. She suggested that we should have two weeks to prepare a presentation in German to the staff of the museum (about 15 people) on what our workshop would entail and during that presentation try to teach them some highlights of the syllabus. It actually took us about two weeks to simply translate our proposal. But after those two weeks, our German had improved markedly. After the presentation, we had some time to further prepare and refine our lesson plans, as well as gather any materials we needed for the workshop.

In order to teach a workshop on how computers work, we wanted to have enough computers so that each child would be able to use a computer alone or in a group of three children maximum. The museum arranged for us to use the computer lab at a local commercial business school named Handelsschule Hermann. The school donated the use of the computers and the time in the room to the workshop free of charge.

In order to see how workshops are traditionally run at the KAF, we attended three other workshops that were given before ours in the summer program. We helped out where we could and were given a few opportunities to explain mathematical concepts to the children when the instructor felt it was appropriate. These other workshops were well staffed and the workshop teachers took care of all the planning. We used this time to make observations in teaching techniques that might be useful in our own workshop. The time we spent in the other workshops was also good German conversation experience, which we found helpful in the improvement of our German skills.

2. Description of Other Workshops

The first workshop, entitled "Ein Trümlein, sechs Füsse und eine Nasenlänge" (A Trümlein, six feet and a nose length), was taught by artist Linda Doernbach. A Trümlein, foot, and nose length are all units of length derived from the body; such as a foot is twelve inches or approximately the length of an adult's foot. The workshop covered the history of different measuring systems and perceptions of the human figure. Often artists use geometrical shapes or preconceived proportions to draw a person. Certain aspects of the drawing are always the same, such as the placement of the eyes always above the nose and mouth. However, other aspects of such drawings change with the times and the artist. An example of these differences is the variety of shapes and relative sizes of human features.

The children in this workshop experimented with the concepts of the ancient Greeks as portrayed by the medieval artist Villard de Honnecourt, and Vitruvius' idea that was expressed in Leonardo DaVinci's sketch. To this end, they were told that Vitruvius said that the human body was the perfect form. He claimed it was perfect because with outstretched limbs, a person could make a square and a circle, but he did not say exactly how. The children were asked to try to find a way to draw a person in this so-called perfect form that Vitruvius wrote about. Later, they saw DaVinci's sketch and then made a mural of the "Vitrivian Kid" using themselves as the models.

An important aspect of our participation was to understand how to present the information to the children and keep it interesting and unlike school. Once the background information was explained, it was simple to motivate them to try an activity as long as they were able to experiment and try out their own ideas. The most important thing was to keep the explanations concise and make the activities the focus of the workshop. It quickly became obvious that the children must discover for themselves and come to conclusions on their own terms. Granting artistic license in a drawing class is quite simple, but we felt it was also important for the computer workshop to foster creativity.

The second workshop was "Zahlenwerkstatt" or number workshop lead by Ursel Cornelius, Bärbel Buch and assisted by Beate Hirsch and Nila Almstrom. The subject of the workshop was patterns and interesting tricks one can do with numbers. We began by sitting around a large piece of plywood prepared with a number line drawn in pencil. The number 1 was written in the center of the board, and the numbers were written sequentially in a straight line up to a square number. The number line took a 90 degree turn at each square number and so formed a sort of square spiral. The children painted the even numbers white and the odd numbers different colors. They each made a flag and marked their favorite number. With square pogs they were asked to make a large square with the smaller squares and count the number it took to make a square. Thus they discovered square numbers were the product of a number and itself and that the spiral of numbers had the square numbers in the corners.

Each time a discovery of a pattern was made, the children made something to mark it on the board and wrote a little about it in a notebook that they were able to take home. The square numbers were marked with miniature streetlights made out of flashlight bulbs, wire, paperclips, batteries, and small pieces of wood.

The next day we examined the amount one has to add to a square number to get to the next square number. This led to the discovery of triangular numbers and they were encouraged to play with pennies (actually a German coin called a Pfennig) to further see how triangular numbers fit into the puzzle. One can arrange a triangular number of pennies in an equilateral triangle. When two triangular numbers are added together, the sum has to be a square number. The coins were glued onto the board to mark all triangular numbers.

Through this workshop, the children were able to experiment with numbers and mathematics, and create artwork that represented what they learned.

The workshop entitled “Ich bin doch keine Null” (“I am not a zero”) has a two-fold meaning. One is that all people are important, not zeros, and the other is that the number zero is even important. The workshop goal was to write, produce, and perform a short theatre piece for the summer festival at the KAF.

The general plot was suggested to the children at the beginning. Based on the plot, they chose characters, made costumes, and wrote the story together with some help from the staff. The theme of the play came about when toy designer Elfi Bätz, the workshop leader, was asked to hold a workshop about numbers for the number summer theme at the KAF. She explained to the children, that since she was never clever with mathematics in school, she would do a workshop about the number zero, eliminating the need for math, but still conforming to the theme. She felt intimidated by teachers in grade school because she did not understand math, but nevertheless believed that did not make her less important a person or less intelligent. She also knew that the Romans did not use the number zero, as it was a concept that had not yet been explicitly discovered. Elfi also asked Nila to explain to the children why zero is important in computer work and binary code. Finally, she suggested that maybe in the play we could have an ignorant king, who was somehow introduced to the concept of zero.

The cast of characters consisted of a teacher, two students, a spider (the spider had nothing to do with the plot at all, but the child who played the spider was not interested in playing any other role), a queen, two princesses, a computer, an inventor, a number one, and four zeros, one of which was a zero-goat with an evil decimal point, determined to ruin the play. The children made their own costumes, which all turned out to be fairly elaborate and colorful. The computer costume was two cardboard boxes painted gray, with computer parts drawn and glued on. The on and off switches on the monitor and CPU lit lights on the front indicating that the computer was powered. The computer was also equipped with an electric noisemaker that the actor could control with a switch in his hand. The costume making lasted about a day and a half.

The story was about a student who did not know what one minus one was, and the teacher said she was stupid, just a zero. During the two-hour detention she received for her wise retort, she fell asleep and dreamed that a queen sat on her throne while a computer haphazardly strolled through her palace beeping and saying binary code. She called her daughter, explained the incident with the computer and asked what the zero was that the computer talked about. An inventor was called to explain, but he said that the Romans did not use the zero, which made the lazy queen believe that it was unnecessary to learn about. Then the computer came in and said that the zero was actually very important for the computer. He called in the one and the zeros and they explained base ten place value to the queen by showing that the zero standing to the right of the one made larger numbers. Then the evil zero-goat came in with the decimal point and stood to the left of the numbers, changing the place value held by the one and making the number very small. Luckily, the other zeros and one were able to scare the zero-goat away and remain a large number. The queen then understood why the zero was so important.

The detention over, the teacher woke up the student and said he hoped she was smarter now as a result of the detention, meaning she would watch what she said in the future. She told him that she dreamed about the zero, and knew that one minus one was zero, that the zero was actually very important, so when the teacher called her a zero, it was not such an insult after all.

Whether the kids realized it or not, they probably learned about the zero and its importance in mathematics entirely through writing a play.

The workshop entitled “Coole Zahlen” (Cool Numbers) was led by Ursel Cornelius and Professor Beutelspacher and assisted by Malcolm Beaulieu. “Coole Zahlen” involved the theory of Pascal’s triangle. Pascal’s triangle is formed in the following way: the upper point of the triangle is a 1. Beneath the one are is another row of 1s, i.e.:

1
1 1

The next row is written by taking the sum (addition) of two adjacent numbers in the last row and writing the sum centered directly under these two numbers. The entire row is generated in this way.

1
1 1
1 2 1

This process is repeated again

1
1 1
1 2 1
1 3 3 1

After this process is repeated for the desired length of time, certain tricks with the numbers can be performed. For instance, a downward triangle can always be found within the triangle. What is interesting is that the tip will always be a negative number. Also, opposing triangles can be drawn around a number and the results of addition, or multiplication, will always be the same. These activities took up most of the first day.

The second and third days were primarily spent working on a display for the museum. This was a 3D model of Pascal's triangle. It was composed of blocks of wood that were painted and numbered. The children got to use power tools to drill holes in the blocks. Pegs were inserted and the blocks were arranged in rows, with a board as a separator. This enabled the blocks to spin freely and create a multi-colored Pascal's triangle. Of course, there were breaks for playing in the museum but when things were finally finished, the triangle was anchored to the wall and put on display.

The teaching technique in the workshops also emphasizes hands-on learning and self-guided discovery. Workshops utilize more activities than explanations to demonstrate the material. Most activities involve creating a product that the children may eventually take home, after it goes on display in an exhibit about the workshop in the museum. In some cases, the children make their own individual creations, other times one product represents the combined ideas and work of all the children. In order to apply this philosophy, we tried to make our workshop as hands-on as possible by initiating activities to demonstrate the wonders of computers. We quickly found that the children would eagerly participate in activities, while explanations leading up to activities were difficult.

2. Description of Our workshop

The title of our workshop was "Vom Abakus zum Mikrochip – Wie ein Computer Funktionert" ("From Abacus to Microchip, How a Computer Functions"). The topics covered on the first day of the workshop were various historical number systems, binary numbers, the computer programming concept and its history, a demonstration of the evolution of data storage and building a device that shows how a punch card reader could work. The second day included the history of computing machines, a chance to see a wafer of microchips under a microscope and a trip to the computer lab. On the third day, the children participated in activities to learn how a printer works, how to use a flowchart, and how to write a program in C++ to calculate the relative weight of something on another planet.

The children arrived to the workshop on the first day and made nametags and reluctantly played a name game. They were quite shy, but willing to participate. The thirteen children who came to our workshop were between 9 and 13 years old, the majority on the younger end of the age scale. Twelve of the children were boys and one was a girl. All of the children filled out a survey before they came to the workshop. All of them had home computers that they used mostly for games and painting, and to a lesser extent for writing. The favorite subjects were math, sports, social studies, and geography. ✓

Naturally, the children in our workshop eagerly wanted to know immediately what was planned. They hopefully asked if we would get a chance to look inside a computer, use a computer, and wondered what we were going to build with the materials they saw prepared on the supply table. These were the sorts of activities that we planned to show them how a computer functioned, or at least an introduction to whet their appetite for the topic.

We began with historical developments that led to the computer. We introduced the idea that different people throughout history have used different methods of counting. Even today we use several number systems. Computers count in the binary code, which is far different from the everyday counting system people use. We cited examples of this and drew them on the chalkboard. The ancient Mayans used a base twenty system for about 2000 years. We explained to the children that the number one was represented with one dot; two was represented as two dots. The children were asked to guess how the next few numbers would be written, and correctly guessed that the number five would be a dash mark instead of five dots. They continued to guess how the Mayans represented the numbers up to twenty. Topics in our workshop included the history of computers and various counting systems. The intent was to show that many systems of counting have been used through history and computing devices that are not currently considered computers were quite important inventions that eventually led to the development of the modern computer. A slide rule, for instance is a device that allows one to approximate products of multiplication quickly and easily.

The children were given an introduction to the binary system. After explaining that a computer uses electricity to 'think' and how electricity can only be on or off, place value was explained. In order to make binary place value a bit easier to understand, we started with decimal place value because they already understood that. After a few examples, the concept of place value in the binary system was somewhat understood. We moved on to addition of binary numbers. This was done in a similar manner, reviewing the process of addition in the decimal system and then relating that to the binary system. Several examples were done in order to familiarize them with the method. Lastly, we introduced how one can convert numbers from the base 10 system to binary.

The "8 bits in 1 byte" display was used here to help illustrate a few different things. One of the things that it illustrated was the fact that the computer can only use electricity and electricity only has two states, on and off. The display was a row of 8 bulbs that literally plugged right into the wall. Each bulb had a switch that controlled the electricity to it. The second thing that it was used to demonstrate was place value and how counting in binary was done. Each "column" holds the value of twice the one to its right, instead of 10 as in the decimal system. Lastly, the display showed storage. Even though the computer thinks in bits, capacity is measured in bytes. 1 byte is composed of 8 bits, which is the amount of bulbs used in this display.

We hoped that this discussion would allow the children to be able to visualize how the computer actually uses the numbers. But simply explaining it and hoping they would remember was not how we wanted to conduct the workshop. In order to make it more fun and less like school, we played a game so they could practice. The game is competitive but still organized enough so that children can have a chance to practice without pressure when it is not their turn.

The game was called around the world. The children sat in a circle except for one child who stood behind one of the children sitting in the circle. These two children got a question from one of the instructors and the object was to yell out the answer as fast as possible. The first to say the correct answer would move to the next child and the game would continue this way. If the child who was standing got the right answer he would simply move to the next child in the circle, if the sitting child answered first, he would stand up and move on, and the other child in that round would sit in that place. Anyone who made it all the way around the circle won the game.

Perhaps all people have to be told at some point that computers are not intelligent in any way, they are simply machines, and they cannot think for themselves. A computer can only solve a problem and tell you the answer if it is given a set of very logical, precise instructions where each statement is very simple by itself. This was the next part of the workshop, an introduction to the behind-the-scenes world of computers. All of the students in the class had used computers for playing games, drawing, or homework, but probably never realized before that behind those programs are millions of lines of code directing the computer in every minuscule detail of operation. We explained this to the children in a short script that was read to them.

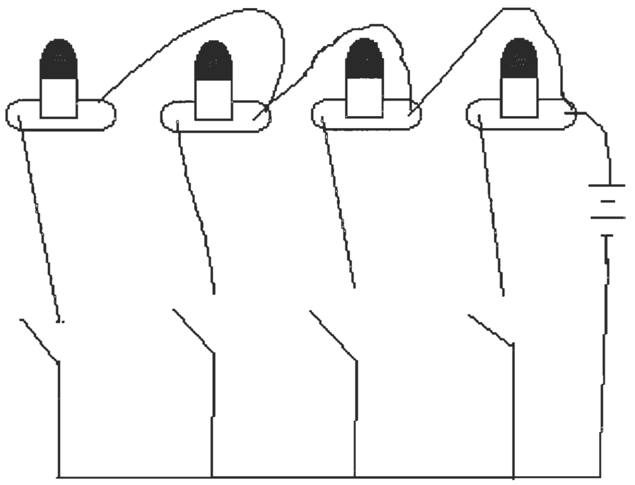
After understanding that programs were necessary to make a computer work, some history was presented to the children. We explained how programming evolved from long strings of 1's and 0's, to pseudonyms of words, and then became even more user friendly because the code mimicked the human language and was easier for programmers to understand. We also explained some of the history of data storage. Homemade punch cards were passed around for the children to look at more closely while we briefly explained how they worked. The same was done with a cassette tape and a dismantled hard-drive. A CD was also passed around but this was not really explained since a demonstration was planned for later.

Before the workshop, we had planned to have an activity where the children would build some kind of computer of their own. Of course we did not build our own

Pentium processors but a model that shows how a computer uses electricity to transmit information. Punch card readers served as our inspiration for the activity. The activity began with the story of Herman Hollerith (see literature review) who made a census machine with punch cards.

The project ended up being called the clothespin computer, because the switches on it were made out of clothespins. A 6x12x1-inch piece of wood was cut and painted for a base. Four light bulbs were connected to a battery in parallel, the connection to each light bulb having a switch made out of a clothes pin with metal contacts to selectively turn lights on and off. If a piece of paper was inserted between the metal contacts of the clothespin, or if the pin was held open, the light corresponding to that switch would go out. Thus, if a punch card, with appropriately placed holes were put between all the contact points, the pattern on the card would be transferred to the lights. (See Figure 1)

Figure 1 – Sketch of Clothespin Computer Wiring



The children saw the prototype of the clothespin computer and were challenged to show how a pattern on the punch card that went with it could be transferred to the same pattern on the lights. The children worked together and found how to put the card to make it work after several tries. The children were given enough materials to build their own clothespin computers and the tools to do so. The materials were pieces of wire with pre-fabricated connecting hardware attached, a small board, light bulbs and sockets, batteries, small screws and nails, clothespins, and thin pieces of sheet metal. After a small demonstration on the concept of electricity needing a closed circuit on which to travel, and a schematic drawing of the wiring put on the chalk board, the children set to work making their clothespin computers. It took about two hours total to finish this project and make a punch card custom fit to each machine. In the end, all of the models worked.

Since people are always trying to make their lives easier, they are always inventing new ways to have machines do work for them. For example, a farmer uses a plow to harrow the soil for his crops, but then finds it easier to have an animal pull the plow. Assembly lines make fabrication easier, and robots and computers on assembly lines mean less hard work for people. Mathematical calculations are sometimes difficult and incorrect answers have led to disasters in the past, so inventors have made various calculating machines to make math easier. We asked the children to think of as many examples of machines or techniques people use that make less work. As they shouted out the answers, Malcolm furiously scribbled them on the chalkboard, so they could see how quickly we could fill the board with examples.

Then we showed them some old calculating inventions and how they worked. We explained that the abacus was a frame of wires holding beads that could represent numbers. Each row of beads represented one place in the base ten system, where the number of beads on the right side was the digit that filled that place.

The slide rule is based on logarithms. To multiply two numbers, on can add the logarithm of the two numbers and take the anti-log of the sum. This is the product. The slide rule eliminates the need for tables, because the distance on the slide rule between the numbers is proportional to the log. We did not explain any more to the children about the slide rule except to demonstrate that it worked for the basic multiplication tables that they already knew.

Then we had a race to show the kids that the slide rule really worked. One child came to the chalkboard and wrote a multiplication problem with two four-digit numbers. The children calculated the product with pencil and paper, while Malcolm multiplied on the chalkboard, and Nila used the slide rule to find the product. One child knew some trick for multiplying extremely quickly and finished before Nila even picked up the slide rule. Nila found the approximate product next and the rest of the children finished calculating after.

The race showed the children that the product could be found relatively quickly and accurately with the slide rule, and the results of multiplying with paper and pencil varied greatly. The children complained that the answer that the slide rule produced was not exact, and so we had to admit that it was only a tool for approximating, but usually one could still be safe with this approximation as long as one used it as such and were careful. The next race was between a child with a calculator, Malcolm with the chalkboard, and Nila with the slide rule. Another long multiplication problem was given and the calculator of course produced the exact answer in seconds.

That brought us to the invention of microchips and how they have made computers what they are today. The children used microscopes to see what microchips look like under the microscope at 2X and 10X magnification. The activity was interactive because the children were able to change the magnification on the microscope and adjust the focus. They were amazed how small each one was, how compact, and how complicated.

We had introduced many concepts on how computers function, and it was finally time to use a computer in the workshop. We took the children to a computer lab in order for them to gain some practical programming experience. The lab was equipped with 30 PC's running Windows 95 that were networked to a linux server. Since there was not any sort of development application installed on the Windows Computers, the students telneted into the linux server and performed all of their programming and compiling there.

The first thing that the children did was to run a program that was already made. This program simply asked for some biographical information and then repeated it back to them. It was more to illustrate the point of how computers are not smart; they just do what they are told really quickly.

Since computer programming is a difficult field that is extremely conceptual, and for most hard to grasp, very little 'real' programming was actually done. There were only three basic concepts that we desired to get across to the children. These were input/output (I/O), variables, and the 'if' statement.

The third day involved a little bit of insight into programming and also some explanations of how printers functioned. In order to explain how a printer functions, its description was taken down to the very basic level. It was explained to the children that the computer sends the printer a list of commands and the printer does whatever the commands tell it to. The computer sees a piece of paper not as a plain piece of paper but as a sheet of paper with a coordinate system. The different commands that are sent to the printer tell the printer where to place a mark and if possible, what color the mark should be. To better explain this to the children, they were given a sheet of graph paper with the rows labeled with letters and the columns labeled with numbers. This can be seen in figure 2 below.

Figure 2 – Printer Co-ordinate System

	01	02	03	04	. . . .
A					
B					
C					
. . .					
Y					
Z					

They were given a piece of paper with the coordinates of the squares to fill in. In order to make this somewhat interesting, colors were also taken into consideration. Special characters designated these. For instance, the code '!A12' would mean fill in the square located at row A, column 12 with the color black. When the children were finished filling in the squares, they would have a resulting picture of a boat, house, dog, or other items. The interesting thing about this activity was that it was intended for the children to start at one side of the page and work to the other. In reality, what happened is that some of the children would do all of one color first, or a certain sections of the picture. Some of the pictures did not really come out at all but it was all for fun. It was interesting to note that the four children who took the time to work from one end to the other competed their pictures quicker and more accurately than the others. These children were also the ones that appeared to be the most interested in what we were teaching and enjoyed programming the most. Since the children were having fun this activity ended up taking a bit more time than anticipated. Once they started to become restless we moved onto the next item, which was flowcharts.

When it came time to explain flowcharts, only two basic symbols were used. These were the rectangle and the rhombus. The rectangle represented a command in the flowchart while the rhombus represented a choice. One of two different paths could be followed depending on the outcome of that choice. The flowchart we used involved having an individual move from one chair to another. In order to keep their attention we had one of the children be a robot and the others tell the robot what to do based on the options available in the flowchart. By doing this, we hoped to show that a program only did what it was told. Therefore, when one was programming, every situation had to be accounted for and each step explained in detail. This was also shown in scenarios where a seat was available but the flowchart did not provide directions to that particular area. After the children became familiar with these concepts we decided to let them try and create their own flowcharts. After providing them with some ideas they went to work, discovering how many steps were actually needed when performing simple tasks.

Unfortunately, these activities took up more time than anticipated. As a result we did not get a chance to explain the different parts of the computer or even describe the CD-ROM demonstration and how a CD-ROM functions.

The afternoon of the final day of the workshop was spent at the computer lab. The problem at hand was explained to the students and how they were supposed to solve this problem on their own; we were not going to write the program for them. In reality, this is not what happened. The students were given a brief overview of the needed commands to open, save, or compile their files. They were given two sheets of paper.

The first paper had all the commands learned from the previous day and examples of how to use them. The second sheet of paper was a description of steps that needed to be taken in order to successfully complete the program. We spent the afternoon wandering around the lab, fixing typographical errors and giving guidance to those students that were stuck and unsure of how to proceed. A few of the students finished sooner than the others and they went on to expand their program from what was necessary to complete it. The students got very excited when the programs worked or when fictitious inputs (e.g. "I weigh 400 kilograms on earth") yielded outrageous weights on other planets.

When the students were finished with their programs, we wrapped things up and went back to the KAF where we had them fill out a simple questionnaire. After this was completed, pictures were taken, good-byes were said and they were allowed to play in the KAF for the remainder of the time that they were there.

4. Why and How We Used C++ Programming:

The theory behind having the children write a program of their own was to demonstrate how precise everything had to be but at the same time, that they could do it. The idea that we wanted to convey to them was that the computer had to be told everything to do. Although this may have been frustrating at times (when fixing syntax errors for instance), with a little effort this could be done enjoyably and when they were finished, they would have a simple program about which they could say, "I made this."

Up until a couple weeks before the workshop, we had no definite program for the children to write. We wanted to be sure to match the program's difficulty level to the ability of the children. The only way to do this was to observe the children in the other workshops and apply what we discovered to what we wanted to do. After arranging time in the computer laboratory and observing the children's abilities in other workshops, we decided the children would write a program to calculate weights on other planets.

We wrote a short introduction program that the children would run when they first arrived at the lab. The program asked them for some brief biographical information and then printed what was inputted. It asked the children if they thought the computer was smart and would print out something different depending on if the answer was a yes or a no. Since this was a simple program, it was done quickly.

In order to decide if having them write their own program was actually achievable, someone with no programming experience was first asked to write the intended program. Although there was some slight syntactical confusion along the way, they were able to successfully write the program. Not only did this reveal to us that someone with no programming experience could write the intended program but also where the children may have trouble along the way in theirs. At this time, the exact layout of the program, and how much programming we wanted to teach the children, was decided. We realized that the program really did need to be kept simple and to just doing simple I/O (inputting and outputting text), and "if" statements (if a condition is true, execute a certain section of code). This was done not only because anything more would have been too complicated, but we also only had two 2.5 hour segments to do this in. We also decided to use C++ rather than C. The main reason for this is I/O, which is much easier in C.

In order to more effectively learn the material, the students were taught these few programming techniques in a building block approach, that is, new material required use of the previous material. This was done not only to reinforce the previous ideas but also to make sure they had a good understanding of them since we were limited by time from going back and re-teaching them. Our two days were used for two separate functions. The first day involved an introduction to the needed commands and making sure they had an understanding of them. The second day was more of a "let the children go at it on their own" day where they would be given a goal and have to figure out how to get there.

The first day was a little difficult as it involved actually getting everyone together and going to the lab. The children needed to be familiarized with the PC and how things worked. Luckily they all had home computers so things were not as difficult as they could have been. After everyone was situated and files were opened, the children wrote the program that every programmer first writes, "hello world." The program is pretty self-explanatory and when run, just prints the line "hello world." The next thing that was explained was variables and how they are just names that can store information. Since output is essentially the opposite of input, explaining that was relatively easy. After a quick example, most of the students were having no problems with this. The program that they ended up writing asked how they were, took their response and outputted "You said . . . ." The final thing which they needed to learn was the "if" statement. This is where the younger students had trouble. Not only because they were younger than the age group that we requested, but also since it was a nice day out and they were getting distracted being in a new place in the middle of the city.

The "if" statement took the longest amount of time to get across since it not only requires a bit of abstract thinking but also because of the language barrier that was present. Eventually though, almost all of the children did understand what we wanted to explain. A few of them did very well and picked it up extremely quickly.

As a result of the younger children's difficulty, it was uncertain of how well giving them a flowchart would work, or even if they would be able to handle their own program. It was decided to give them a sheet with all the learned commands on it and instead of a flowchart, they received a detailed outline of what needed to be included in their program. The program would simply ask for a weight (in kilograms), print out a list of different planets, request a planet and print out the new weight value on that planet. This would keep things simple and at the same time, go along with the "Number" theme that the KAF was promoting.

When the students returned to the lab the next day, they were given a brief review and most of them got right into the required programming. It was explained that everything they needed was on the two pieces of paper and the afternoon was spent walking around correcting syntax errors and answering questions. In an attempt to make things easier, the students were divided up into teams of three. By doing this, we had hoped that they would use each other as resources. Surprisingly, the day went better than expected as some of the children even finished their program ½ hour – 45 minutes before we had to leave. The ones that finished early went on to improve their programs. They either added more planets or made things more pleasing to the eye by changing the way in which things were printed out. Eventually, everyone finished their program and as it neared time to leave, the children found out about paint and a couple of edutainment games that were installed on the computers.

C. Results

In order to evaluate the workshop, the children completed a post-workshop survey that we wrote. The questions on the survey were:

- Did you have fun in the workshop?
- On a scale of 1 to 10, how much do you feel you learned?
- What did you find most fun?
- What did we do that you liked best?
- What would you have liked to learn that we did not cover?
- Would you like to learn more about computers?
- Was there anything about the workshop you did not like? What?
- Would you recommend this workshop to a friend?

The pre-survey to find out ahead of time what the children's specific interests were and the evaluation survey are new procedures for the KAF. Most of the questions were answered in a few words or a sentence, and the information from the children was not written in as much detail as perhaps and adult would write. The information was enough to indicate what the favorite activities and topics were and allow us to reflect on the workshop's successes and failures.

The survey showed that the reactions of the children to our workshop were in general positive and enthusiastic. The children signed up for this workshop voluntarily and their parents had to pay for it, therefore we expected that they would attend because they were genuinely interested in computers and we planned as many activities as possible. Some of the children indicated in the survey that the workshop was very intensive. They listened thoughtfully to most of our instruction, however we found that long, detailed explanations were too tedious for them and some of our activities were more self-explanatory than we thought. According to the post-workshop survey, the children especially liked the hands-on activities, especially the clothespin computer and the programming activity in the computer laboratory.

The printer activity taught the children how a printer receives information from the computer and subsequently transforms it into print. No explanation of how a printer physically works was necessary for the activity. We were surprised to see that the children simply went to work coloring blocks according to the code they were given and understood how the printer worked simply by doing the activity. The explanation that we prepared for this activity was less necessary than expected. This is an example of how activities can replace explanations.

The explanations of how a slide rule worked, the history of computers, and binary place value were obviously tedious to the children. They appreciated these lessons to some degree and we know that they understood because they were able to use what was taught to play a game. Perhaps these lessons could have been turned into activities instead of instructions like the printer activity.

At least one child in the class already knew something about most of what we covered. We found that the children who already knew something were likely to explain it to the rest of the class whether they had the floor or not. It was not until late in the workshop that we used this to our advantage instead of fighting it but once we did found it much more effective and efficient at getting the material covered.

V. Conclusions and Future Recommendations:

If another workshop like this one were attempted, we would advise future project teams to make the workshop as activity oriented as possible. Any explanation will take longer than expected and the children are really much happier doing activities. It does not matter if at the end of the workshop the children are experts or not, but that they learned something about the topic and had fun doing so. Any explanation or activity should be very well prepared in advance, so that everything runs smoothly. One technique that worked well was to write down what was going to be said in German and have it critiqued by a native German speaker. That way, the explanation could be read quickly and the activities clarify everything else.

Explanations alone are far too monotonous for this age group and they quickly become restless and stop paying attention. We found that when we were telling the class about history or some other lengthy explanation, having props for show and tell held their interest longer. Passing the abacus and slide rule around and having microchips to look at under the microscope as well as the race activity proved a good balance between activity, show and tell, and lecture style teaching. The explanation of binary code was successful in large part due to the use of the chalkboard to illustrate the details, while perhaps it could have been better if we had used some activities or games in between each step of the lesson.

Recognizing when a student already knows something and could explain it to the class is also helpful. Allowing the children to explain to the rest of the class makes the child who is explaining feel proud, and the others are likely to understand their peers better than us.

Some of the activities planned will take longer or shorter than expected and it is important that the syllabus be flexible to accommodate the children's interests. If the children are not enjoying one activity, it is important to move onto something else, or modify the activity to suit them by making it easier, more challenging, or allowing them to work in teams for example.

In any project that takes place abroad, familiarity with the language where one is studying is a must. Without this familiarity, it is very difficult to get around in daily life. If other IQP's were completed at this site, we would recommend the participants have a strong background in German, such as completing a German Sufficiency. Language skills were especially important in the workshops because the children did not speak English. However, most adults speak and understand some English, so interaction with adults is somewhat easier.

We recommend that someone be available for the workshop to translate when necessary. This person should be competent in both languages and reasonably familiar with the vocabulary for subject of the workshop. The translator serves to translate the children's questions or comments in the event that the instructors does not understand after a reasonable attempt. The translator should understand that his or her job is not to take over teaching the class, just to translate occasionally. We would not recommend that the entire class be taught through a translator, and the instructor should do whatever is necessary to teach the class in the native language.

The project students should try to get involved with the activities at the museum and get to know the people around them. Getting to know them will not only make your interaction more pleasant, it will allow a more in depth experience of the local culture. Daily conversation is the easiest and most fun way to learn the language. Making grammatical mistakes in conversation is probably inevitable, but we recommend that students undertaking this project strengthen German skills by trying to speak German as much as possible.

Bibliography

Asimov, Isaac. *How Did We Find Out About Computers?* New York: Walker Publishing Co., 1984.

Atelsek, Jean. *All About Computers.* Emeryville, California: Ziff-Davis Press, 1993.

Berger, Melvin. *Computers: A Question and Answer Book.* New York: Thomas Y. Crowell, 1985.

Dahl, Kristin. *Zahlen, Spiralen, und Magische Quadrate.* Hamburg: Verlag Friedrich Oetinger, 1996.

The Encyclopedia Americana International Edition Danbury, Connecticut. Grolier Incorporated.

Gourlay, Carol. *Visual Science, Computers and Mathematics.* Morristown, New Jersey: Macdonald & Co., 1982.

Graham, Ian. *The Inside Story, Computers.* New York: Gloucester Press, 1983.

Simon, Seymour. *Computer Sense, Computer Nonsense.* New York: J.B. Lippincott, 1984.

Vordermann, Carol. *Mathematik.* Singapore: Tien Wah Press, 1997.

Wright, David. *Inventors & Inventions: Computers.* Tarrytown, NY: Marshall Cavendish Corporation, 1996.

Web Resources:

http://www.fulda.de/inhaltsseiten/region_ueberblick/index.htm

Appendix 1 - Initial Survey Data:

The following data was gathered from the children attending this workshop via a questionnaire that was mailed to them prior to the workshop.

Chart 1 – Age Distribution of Students

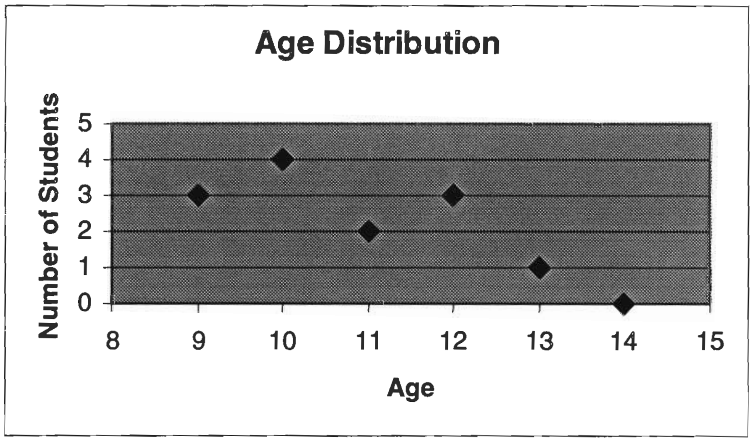


Chart 2 details the actual questions and responses from the students.

Chart 2 – Questionnaire Questions and Student Responses

Name	Age	Family Comput er	Used For	Know how it functions	Used the Internet	Likes Spaghetti	Favorite Subject	Why they are in this workshop*	Can Speak English
Augustinski, Silke	11	Yes	Painting, inventory	No	No	Yes	Geography	1	A Little
Hütter, Jan-Christian	9	Yes	Games	Yes	Yes	Yes	Math	2	Poorly
Hartmann, Simon	13	Yes	Games, Work	Somewhat	Yes	Yes	Social Studies	2	Yes
Lindner, Dominik	10	Yes	Games	No	Yes	Yes	Practical Learning	3	No
Lorsbach, Christian	11	Yes	Games, Edutainment	Yes	No	Yes, but w/o meat	Math, Sports	2	A Little
Niebling, Max	12	Yes	Games	No	No	Yes	Social Studies	3	Yes
Rousch, Marius	10	Yes	Games, writing, & learning	Somewhat	No	Yes	Sports	1	A Little
Rousch, Tobias	10	Yes	Math, games	Yes	No	Yes	Sports, Art	2	Somewhat
Rubay, Patrick	12	Yes	Games,	Yes	No	Yes	Geography	1	A little
Sauer, Alexander	9	Yes	Games	No	No	Yes	Math	4	No
Sauer, Daniel	12	Yes	Games	No	No	Yes	Math	3	Somewhat
Schäub, Michael	10	Yes	Games, Work	Yes	No	Yes	Math, Sports	2	A Little
Wieczorek, David	9	Yes	Games	Yes	No	Yes	Sports	1	A Little

*

1 – Would like to learn something about/are interested in, the computer.
2 – To learn MORE about how the computer functions.
3 – Would like to learn how it functions.
4 – Thinks it would be fun

Appendix 2 - Questionnaire Data:

The following data was taken from questionnaires administered after the completion of the workshop. As one can see, the results are very positive and the hands-on activities were most liked.

Chart 3 – Post Workshop Questionnaire Questions and Responses

Name	Had Fun	Amount Learned *	Favorite thing learned	Favorite activity
Augustinski, Silke	Yes	10	Typing with the Computer	Getting to know you game; Going to the Park
Hütter, Jan-Christian	Yes	9	Programming	Explaining things
Hartmann, Simon	Yes	8	Programming	That everything was well prepared
Lindner, Dominik	Yes	10	Programming	<blank>
Lorsbach, Christian	Yes	10	Typing on the Computer	Getting to know you game
Niebling, Max	Yes	7	Building the Punch Card Computer	The Computer
Rousch, Marius	Yes, very	8.5	Everything	That everything was well prepared
Rousch, Tobias	Very much	8	Working with the Computer	The Computer
Rubay, Patrick	Everything	10	The Computer (Programming)	Everything
Sauer, Alexander	Yes	7	Looking at the Microchips	The stuff we did with the computer
Sauer, Daniel	Yes	8	Everything	Writing Programs
Schaüb, Michael	Yes, very	10	Programming	Programming and Building the Clothespin Computer
Wieczorek, David	Yes	7	Building the Punch Card Computer	Going to the park; Lunch

* On a scale of 1 to 10.

Name	Want to learn but didn't	Learn more about Computers	Didn't Like	Recommend to a friend
Augustinski, Silke	Nothing	Yes	<blank>	Yes
Hütter, Jan-Christian	Nothing	Yes	That we so intensively learned at times	Yes
Hartmann, Simon	How one would create graphical programs	Yes	Tuesday's lunch	Yes
Lindner, Dominik	<blank>	Yes	<blank>	Yes
Lorsbach, Christian	Nothing	Yes	<blank>	Yes
Niebling, Max	Seeing the inside of the computer	Yes	Lunch	Yes
Rousch, Marius	Nothing	Yes, but they have already learned so much	Nothing	Yes
Rousch, Tobias	<blank>	Something more	<blank>	Yes
Rubay, Patrick	<blank>	Yes	Nothing	Yes
Sauer, Alexander	Nothing	Yes	<blank>	Yes
Sauer, Daniel	How the Platters of the Hard Disk actually work	Something more	<blank>	Yes
Schaüb, Michael	Nothing	Yes, much more	Nothing	Yes
Wieczorek, David	<blank>	Yes	<blank>	Yes

Appendix 3 - Tips for Students Who Go to Fulda

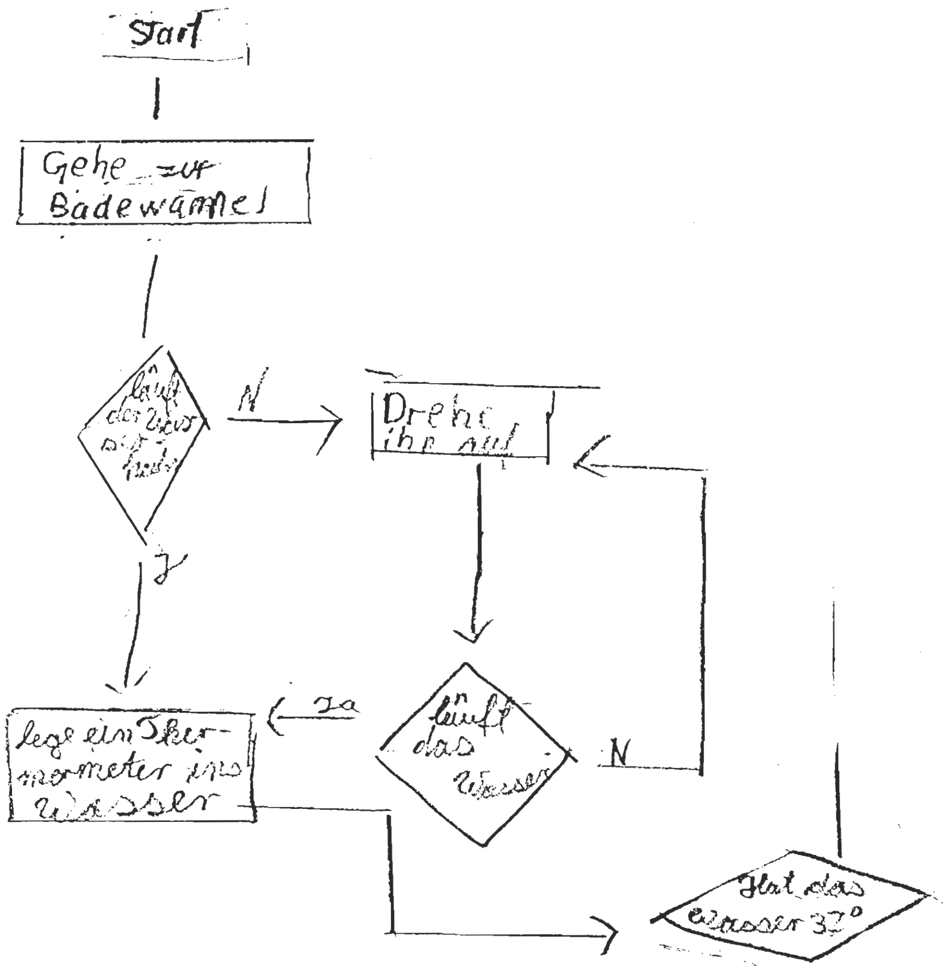
In the FHS Studentenwohnheim (Student Dormitory) the apartments have six single-bedrooms, a kitchen, two bathrooms, dining area, and balcony. All furniture was provided, and we found all the appliances we needed there for cooking. A laundry room and bicycle storage room were located in the basement of our dorm.

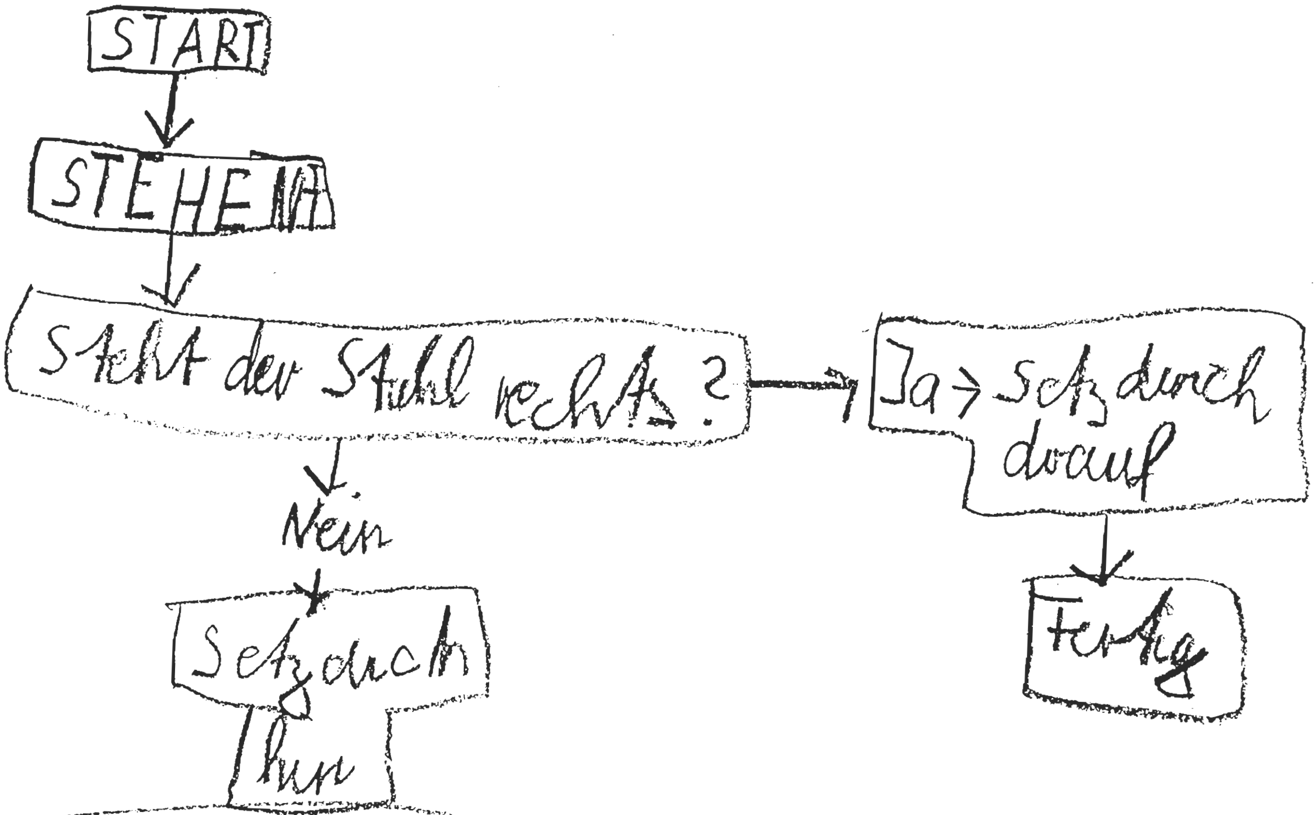
The dorm was located within walking distance of the KAF, old city, grocery stores, shopping centers, bus stop, train station, and restaurants.

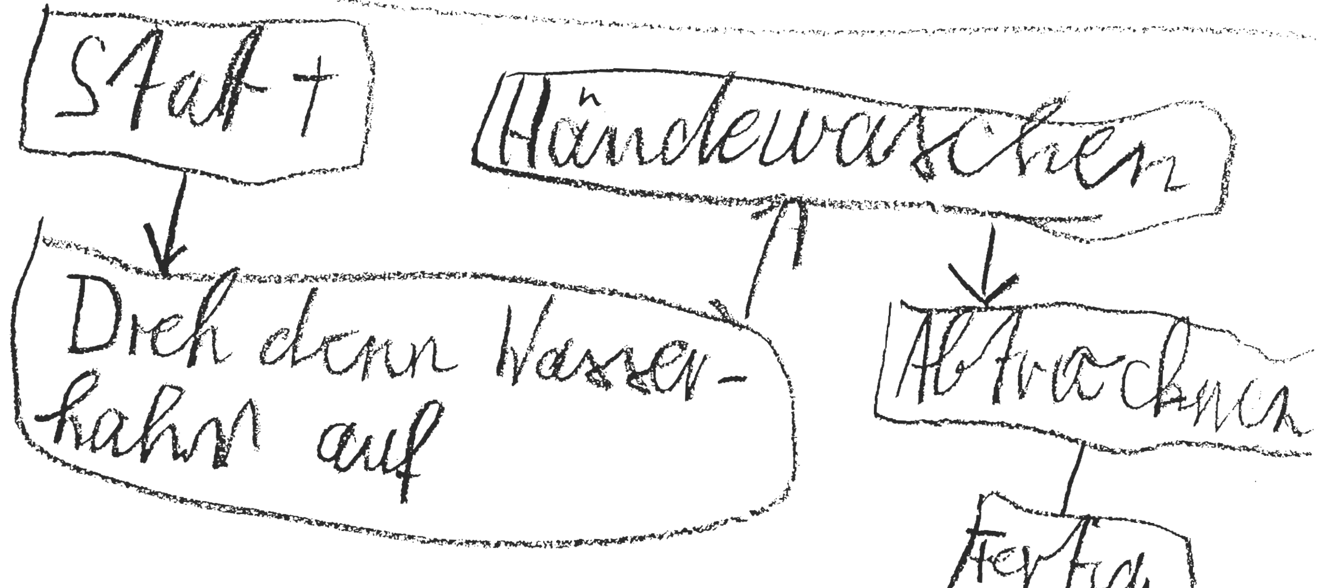
Appendix 4 - Sample of children's work from activities

Flowcharts created by workshop children

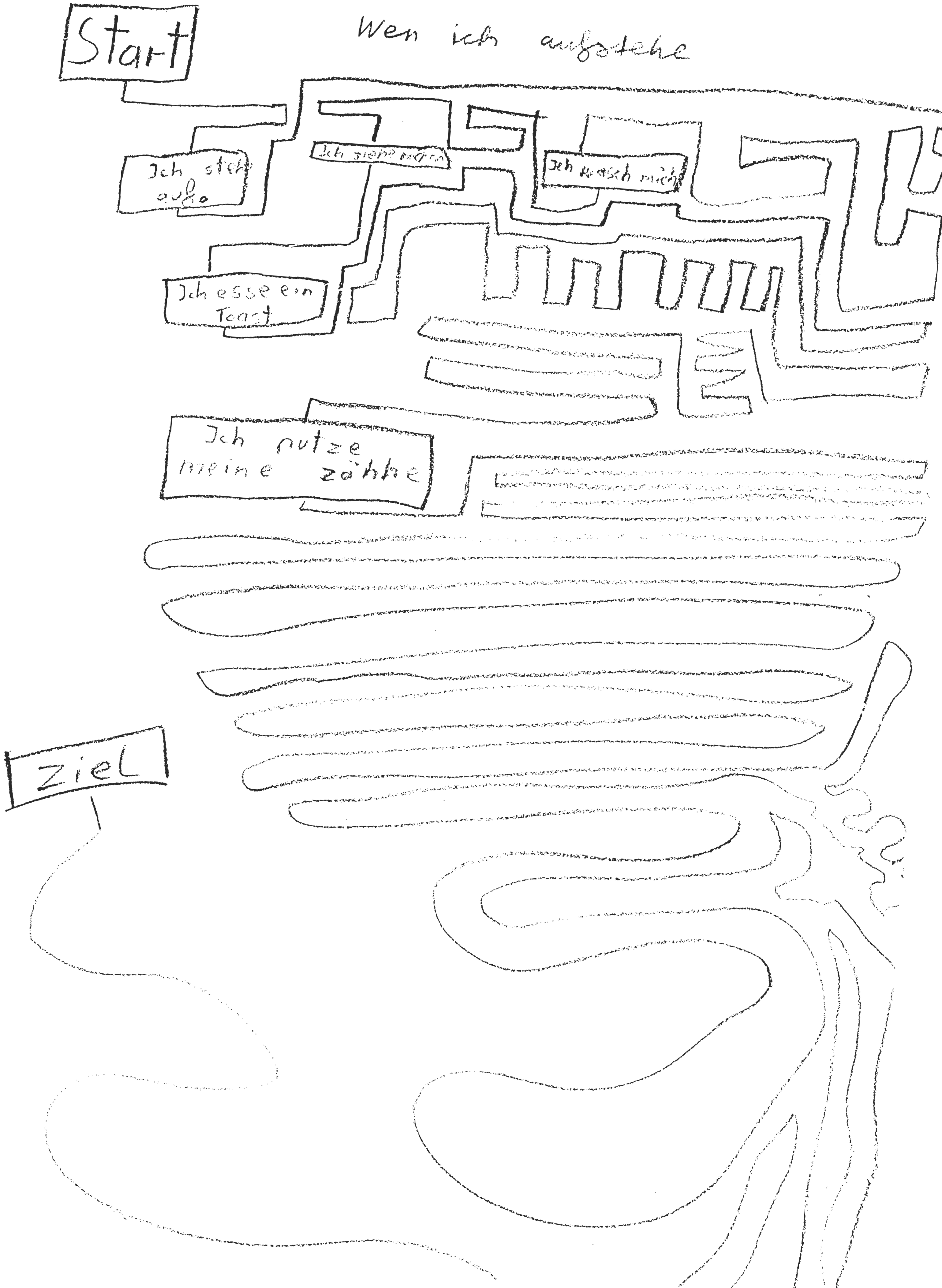
Bad nehmen


Start
Gehe zur Badewanne
läuft das Wasser nicht
N
J
Drehe ihn auf
läuft das Wasser
ja
N
lege ein Thermometer ins Wasser
Hat das Wasser 37°

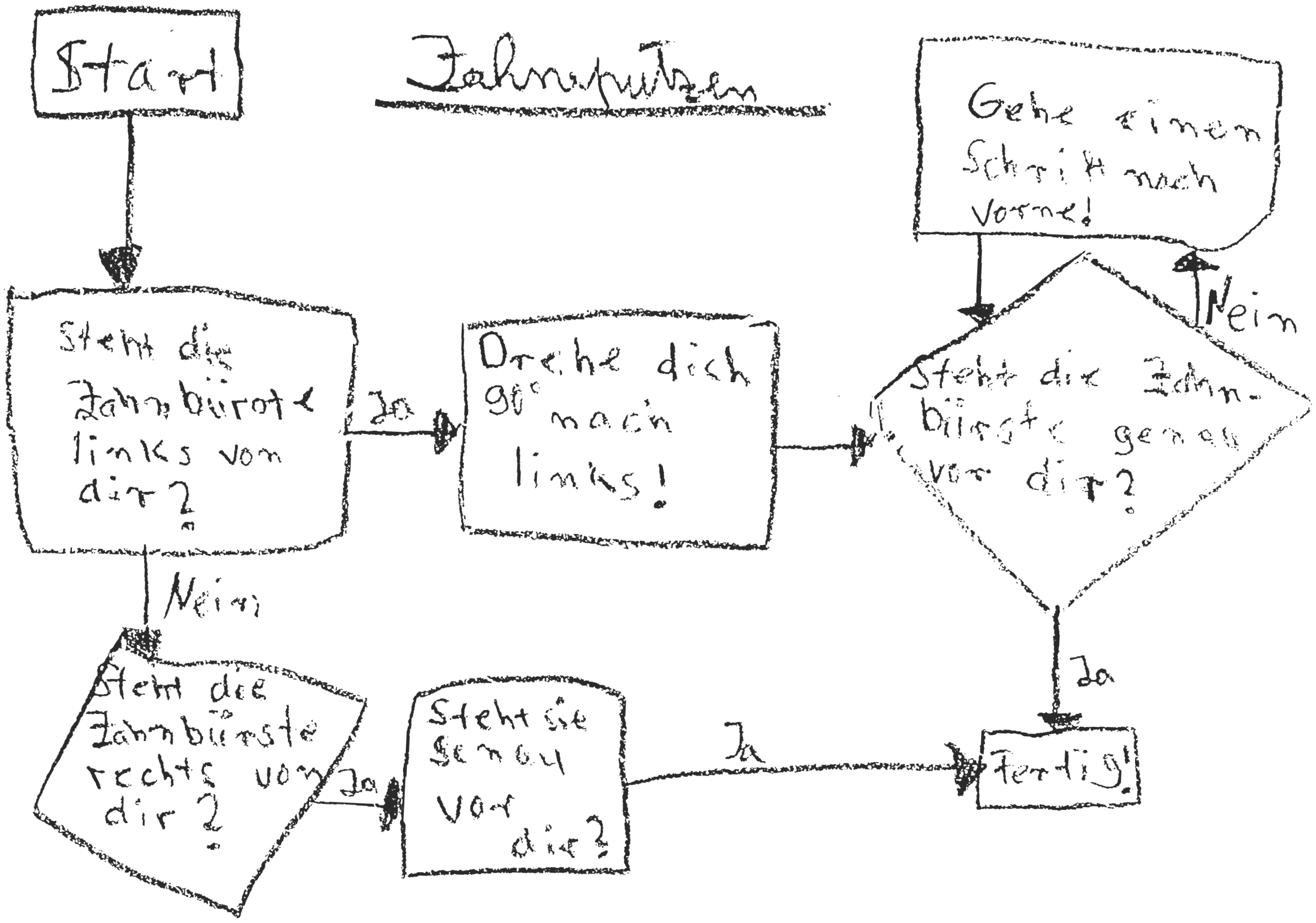

START
STEHE [illegible]
Steht der Stuhl rechts?
Ja → Setz dich darauf
Fertig
Nein
Setz dich hin


Start
Händewaschen
Dreh den Wasser-hahn auf
Abtrocknen
Fertig

Wen ich aufstehe



Zahnputzen


Start
Steht die Zahnbürote links von dir?
Ja
Drehe dich 90° nach links!
Steht die Zahnbürste genau vor dir?
Nein
Gehe einen Schritt nach vorne!
Ja
Fertig!
Nein
Steht die Zahnbürste rechts von dir?
Ja
Steht sie genau vor dir?
Ja

Handouts

Handouts

WIE FUNKTIONERT EIN DRUCKER

Der Code:

!	schwarz
@	braun
#	lila
$	rot
*	grün
%	orange
&	gelb
(	blau

Der Buchstabe stellt die Zeile dar.

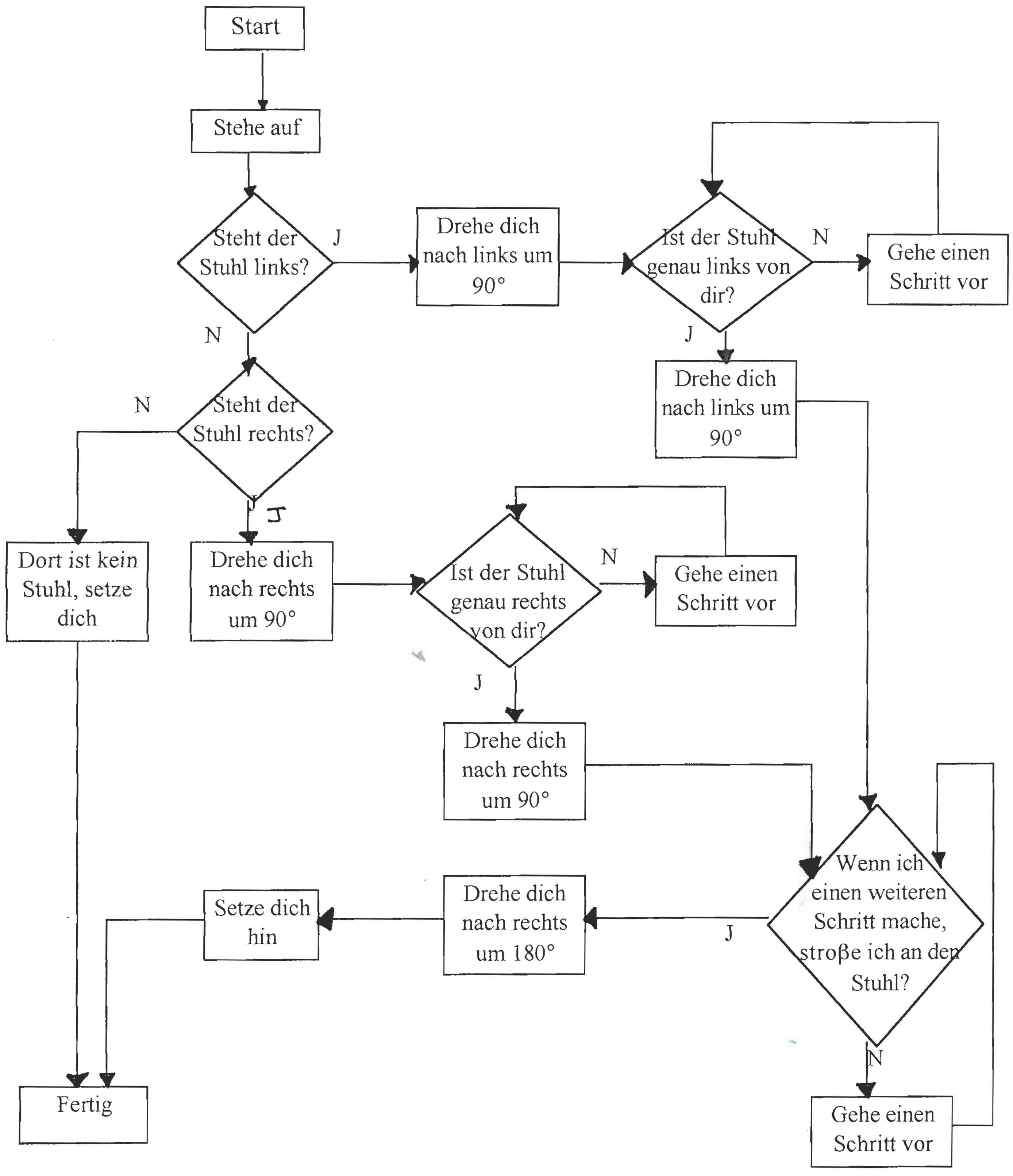
Die Zahl stellt für die Spalte dar.

Der Segelboot Code

@B32	@P32	@Q43	!U42	@X37	(Y26
!B33	@Q15	@Q44	@U48	@X38	(Y27
@C32	@Q16	@Q45	!U51	@X39	(Y28
!C34	@Q17	@Q46	@V21	@X40	(Y29
!C35	@Q18	@Q47	@V22	(X41	(Y30
@D32	@Q19	@Q48	@V46	@X42	(Y31
!D36	@Q20	@Q49	@V47	@X43	(Y32
!D37	@Q21	@Q50	!V50	@X44	(Y33
@E32	@Q22	@Q51	!V51	(X45	(Y34
!E38	@Q23	@R16	!V52	(X48	(Y35
!E39	@Q24	@R50	@W23	(X53	(Y36
@F32	@Q25	@S17	@W45	(Y08	(Y37
!F40	@Q26	!S26	!W51	(Y09	(Y38
@G32	@Q27	!S34	(X10	(Y10	(Y39
!G39	@Q28	!S42	(X14	(Y11	(Y40
@H32	@Q29	@S50	(X18	(Y12	(Y41
!H38	@Q30	@T18	(X24	(Y13	(Y42
@I32	@Q31	@T19	@X25	(Y14	(Y43
!I37	@Q32	1T25	@X26	(Y15	(Y44
@J32	@Q33	!T27	@X27	(Y16	(Y45
!J36	@Q34	!T33	@X28	(Y17	(Y46
@K32	@Q35	!T35	(X29	(Y18	(Y47
!K35	@Q36	!T41	@X30	(Y19	(Y48
@L32	@Q37	!T43	@X31	(Y20	(Y49
!L34	@Q38	@T49	@X32	(Y21	(Y50
@M32	@Q39	!T51	@X33	(Y22	(Y51
!M33	@Q40	@U20	@X34	(Y23	(Y52
@N32	@Q41	!U26	(X35	(Y24	(Y53
@O32	@Q42	!U34	@X36	(Y25	(Y54

01 02 04 06 08 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54

Gehe zu einem anderen Stuhl


Start
Stehe auf
Steht der Stuhl links?
J
N
Drehe dich nach links um 90°
Ist der Stuhl genau links von dir?
N
J
Gehe einen Schritt vor
Drehe dich nach links um 90°
Steht der Stuhl rechts?
N
J
Dort ist kein Stuhl, setze dich
Drehe dich nach rechts um 90°
Ist der Stuhl genau rechts von dir?
N
J
Gehe einen Schritt vor
Drehe dich nach rechts um 90°
Wenn ich einen weiteren Schritt mache, stroβe ich an den Stuhl?
J
N
Drehe dich nach rechts um 180°
Setze dich hin
Gehe einen Schritt vor
Fertig

Programm Flowchart

1 – Ihr müsst 2 Variables haben:
```
double Gewicht;
double Planet;
```

2 – Der Computer muβ nach eurem Gewicht fragen
Ihr könnt den “cout” Befehl benutzen.
Z.B. `cout << “Was wiegst du? “;`

3 – Der Computer muβ muß euer gewicht lesen.
Ihr könnt den “cin” Befehl benutzen
z.B.: `cin >> Gewicht;`

4 – Der Computer muβ sagen, welche Planeten er weiß. (3 Planeten)
Ihr könnt den “cout” Befehl benutzen.
Benutzt ein Nummer für jeden Planeten z.B.: 1 Erde
2 Mars

5 - Der Computer sollt die Nummer lesen.
Mit dem “cin” Befehl

6 – Der Computer soll die Nummer prüfen.
Ihr könnt den “if” Befehl benutzen z.B.
```
if(Planet == 1)
{ . . . }
else if(planet == 2)
{ . . . }
```

7 – Wenn der Computer hat den richtigen Planeten gefunden, er soll das neue gewicht finden. Er soll auch das Gewicht sagen. z.B.:

```
if(Planet == 1)
{
Gewicht = Gewicht * 1.23;
cout << “\n\nAuf Pluto, du wiegst “ << Gewicht << “ kilos!\n\n\n”;
}
else if(Planet == 2)
{ . . . .
```

8 – Das ist alles, du bist fertig!

Computerbefehle

Man muß Computerbefehle genau so wie hier tippen. Benutzt immer kleine Buchstaben mit Ausnahme von etwas zwichen "". Alle Befehle enden mit dem Zeichnen Semicolon (;)

Befehl	Bedeutung
cout << " Text ";	Der Computer soll zeigen, was zwichen ""ist.
cout << variable;	Der Computer soll zeigen, was unter Variable gespeichert ist.
cout << variable << " Text";	Beides
cin >> varible;	Der Computer wartet auf Information, die als"varible" gespeichert ist. Varible kann irgend ein Wort sein.
\n	neue Zeile
char variable [25] ="";	Der Computer schafft einen Platz namens "varible" für irgend eine Information. Die Zahl zwichen [ ] ist die Zahl Zeichen, die das "Varible" haben darf. Dieses Varible muß ein Wort sein.
double name;	Ähnlich wie char varible, aber für Zahlen.
if (varible= = condition) {Befehle;}	Der Computer macht was zwichen {}ist, aber nur wenn das was zwischen ( ) gilt. Wenn das was zwischen ( ) nicht gilt, macht der Computer nichts und geht zum nächsten Befehl. if (hunger= =ja) {ich esse etwas;}
else if (varible == condition) {Befehle;}	Der Computer macht was zwichen {}ist, wenn der „if Satzt" falsch war. Else if (hunger = = nein) { ich spiele; }
else {Befehle;}	Der Computer macht was zwischen {} ist, wenn der „if Satzt" und der „if else Satzt" beide falsch waren.

Kinder-Akademie Fulda

Fragebogen

Nach dem Workshop – “Von Abakus zum Mikrochip”

1. Wie heiβt du?

   ____

2. Hat der Workshop dir Spaβ gemacht?

   ____

3. Wieviel hast gelernt? (kreise eine Zahl ein)

0	1	2	3	4	5	6	7	8	9	10
Nichts					Etwas					Viel

4. Was hat **dir** am meisten Spaβ gemacht zu lernen?

   ____

5. Was hat dir von dem, was **wir** gemacht haben, am meisten gefallen?

   ____

6. Was hast du nicht gelernt, was du gern gelernt hättest?

   ____

7. Möchterst du gern gern mehr über Computer lernen?

   ____

8. Hat dir irgendetwas an dem Workshop nicht gefallen und wenn, was …?

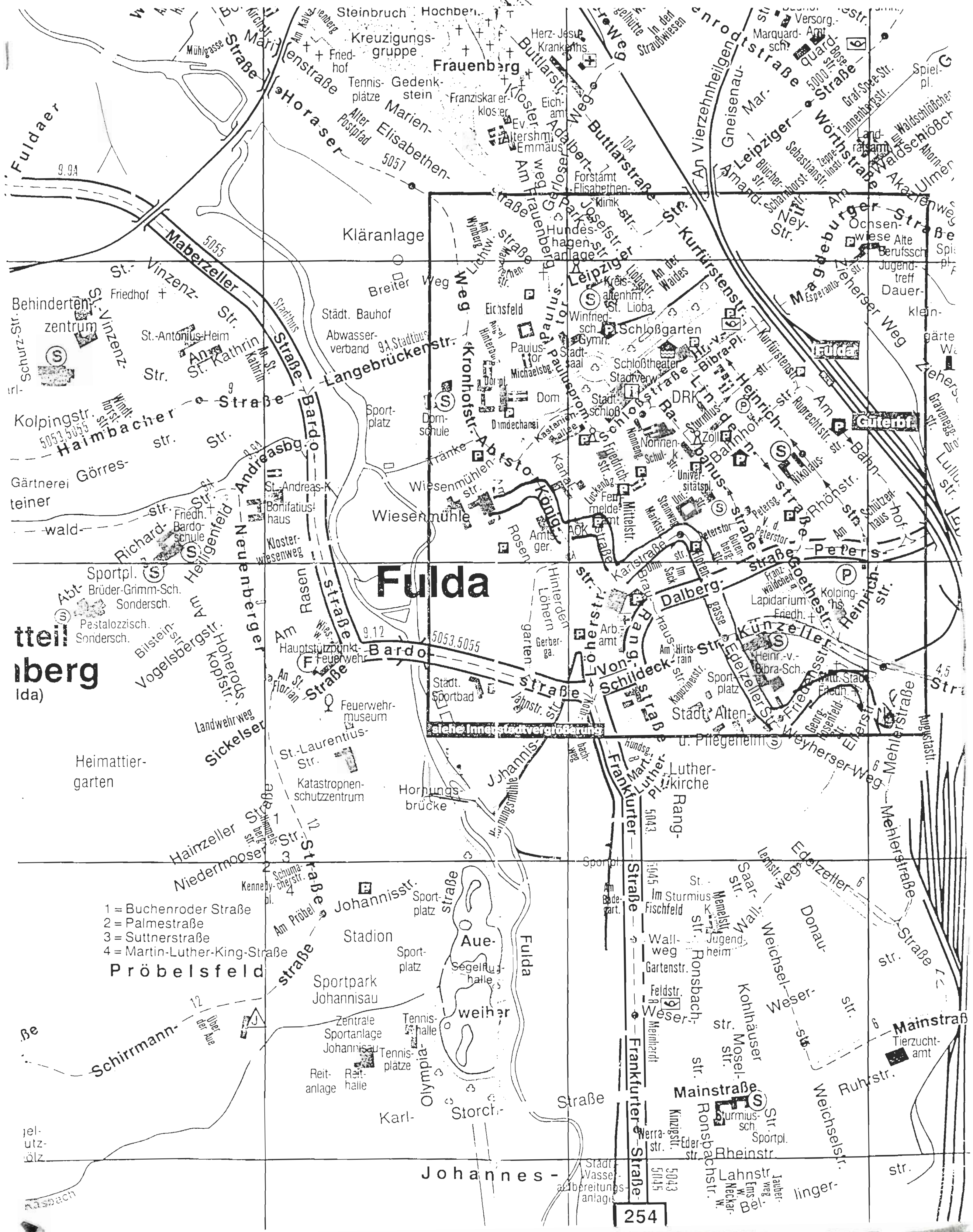
   ____

9. Würdest du diesen Workshop einem freund emfpehlen?

   ____

Map of Fulda, Germany

Old city and vicinity


Fulda
Frauenberg
Kreuzigungs-gruppe
Steinbruch
Fried-hof
Tennis-plätze
Gedenk-stein
Franziskaner-kloster
Ev. Altershm. Emmaus
Herz-Jesu-Krankenhs.
Eich-amt
Forstamt
Elisabethen-Klinik
Hundes-hagen-anlage
Kläranlage
Breiter Weg
Städt. Bauhof
Abwasser-verband
Eichsfeld
Paulus-tor
Michaelsbg.
Dom
Domdechanei
Dom-schule
Stadt-saal
Schloßtheater
Stadtverw.
Stadt-schloß
Schloßgarten
Winfried-sch.
Gymn.
Kreis-altenhm. St. Lioba
DRK
Zoll
Nonnen-K.
Univer-sitätspl.
Fern-melde-amt
AOK
Amts-ger.
Wiesenmühle
Sport-platz
St.-Andreas-K.
Bonifatius-haus
Friedh.
Bardo-schule
Sportpl.
Brüder-Grimm-Sch.
Sondersch.
Pestalozzisch.
Sondersch.
Behinderten-zentrum
St.-Antonius-Heim
Friedhof
Gärtnerei
Hauptstützpunkt-Feuerwehr
Städt. Sportbad
Feuerwehr-museum
St.-Laurentius-Str.
Katastrophen-schutzzentrum
Heimattier-garten
Hornungs-brücke
Lutherkirche
Arb.-amt
Lapidarium
Friedh.
Kolping-hs.
Heinr.-v.-Bibra-Sch.
Sport-platz
Städt. Alten- u. Pflegeheim
Güterbf.
Nikolaus-
Schützen-haus
Ochsen-wiese
Alte Berufssch.
Jugend-treff
Land-ratsamt
Marquard-sch.
Versorg.-Amt
Dauer-klein-gärten
Stadion
Sportpark Johannisau
Aue-weiher
Segelflug-halle
Zentrale Sportanlage Johannisau
Tennis-halle
Tennis-plätze
Reit-anlage
Reit-halle
Sport-platz
Jugend-heim
Sturmius-sch.
Sportpl.
Tierzucht-amt
Städt. Wasser-aufbereitungs-anlage
siehe Innenstadtvergrößerung
Maberzeller Straße
Langebrückenstr.
Haimbacher Straße
Bardostraße
Neuenberger Straße
Horaser Weg
Marienstraße
Elisabethenstraße
Buttlarstraße
Leipziger Str.
Kurfürstenstr.
Magdeburger Straße
Ziehers Weg
Heinrichstraße
Petersstr.
Dalbergstr.
Künzeller Str.
Von-Schildeck-Str.
Rangstraße
Frankfurter Straße
Johannisstr.
Johannes-Str.
Karl-Storch-Straße
Schirrmann-Str.
Sickelser Str.
Hainzeller Straße
Niedermooser Str.
Kronhofstr.
Abtstor
Königstr.
Löherstr.
Lindenstr.
Rabanusstr.
Bahnhofstr.
Rhönstr.
Edelzeller Str.
Weyherser Weg
Mehlerstraße
Mainstraße
Ronsbachstr.
Weserstr.
Weichselstr.
Wörthstraße
Gneisenaustr.
An Vierzehnheiligen
Vogelsbergstr.
Hoherodskopfstr.
St.-Vinzenz-Str.
Kolpingstr.
Görresstr.
Andreasbg.
Heiligenfeld
Pröbelsfeld
1 = Buchenroder Straße
2 = Palmestraße
3 = Suttnerstraße
4 = Martin-Luther-King-Straße