C2110 *UNIX and programming*

Lesson 5 / Module 2

PS / 2020 Distance form of teaching: Rev1

Petr Kulhanek

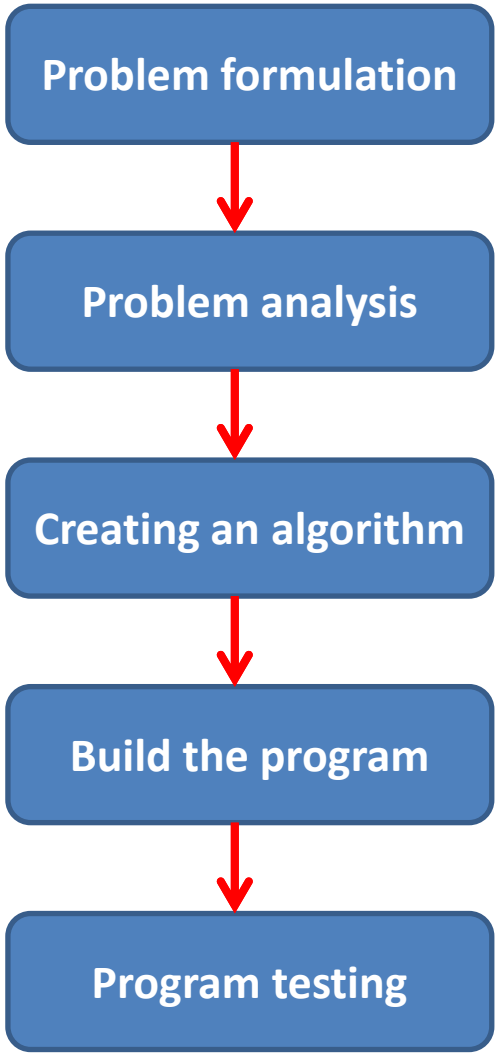
kulhanek@chemi.muni.cz

National Center for Biomolecular Research, Faculty of Science
Masaryk University, Kamenice 5, CZ-62500 Brno

Basics of Programming

https://cs.wikipedia.org/wiki/Algoritmus

Algorithmization



Problem formulation

It is necessary to precisely formulate the requirements, determine the initial data, the required results and the accuracy of the solution (for numerical problems).

Problem analysis

We will verify that the task is solvable for the expected input data and, depending on the nature of the task, we will suggest the most suitable solution.

Creating an algorithm

We will design a clear sequence of operations that need to be performed in order to correctly solve the task.

Build the program

Based on the algorithm, we will create the source code of the program in a selected programming language.

Program testing

Finding syntax errors in the source code and logical errors in our own design. Verification of the program's functionality on the entered data.

Algorithm

Algorithm is a precise guide or procedure by which a given type of task can be solved. The concept of an algorithm most often appears in programming, where it means a theoretical principle of problem solving (as opposed to the exact notation in a particular programming language). In general, however, the algorithm can occur in any other scientific field.

Required features:

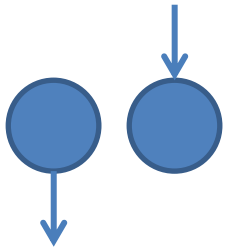
- **Determinability** - the algorithm must be accurate, comprehensible and unambiguous, i.e. the next step is clearly defined at each point and must always provide the same results for the same input data. (The operation of the algorithm must not depend on the will of the person or on the characteristics of the devices that run it).
- **Generality -** the algorithm is not used to solve only one task, but it is a solution of a whole group of tasks, which differ from each other only by input data. Input data may vary within certain limits.
- **Finality** - we must obtain the searched results after a finite number of steps, the algorithm must end after a finite number of steps.

Writing algorithms:

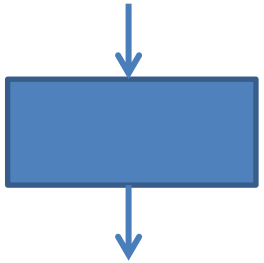
- verbally
- pseudocode
- graphically (flow chart, etc.)

Flowchart

A flowchart is a graphical representation of an algorithm. The diagram consists of marks (blocks) that are executed in order from the beginning to the end of the diagram in the direction of the arrows.

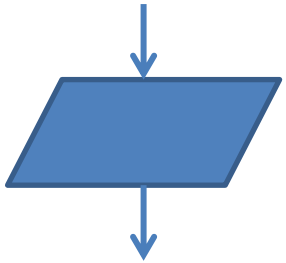


junction (start, end)

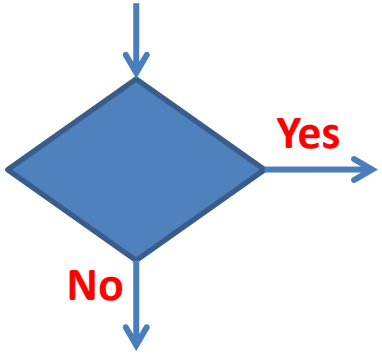


processing block

The marks are combined so that the diagram describes a clear sequence of operations that must be performed in order for the task to be solved correctly. The operations or groups of operations to be performed are entered into the individual marks.



input/output block

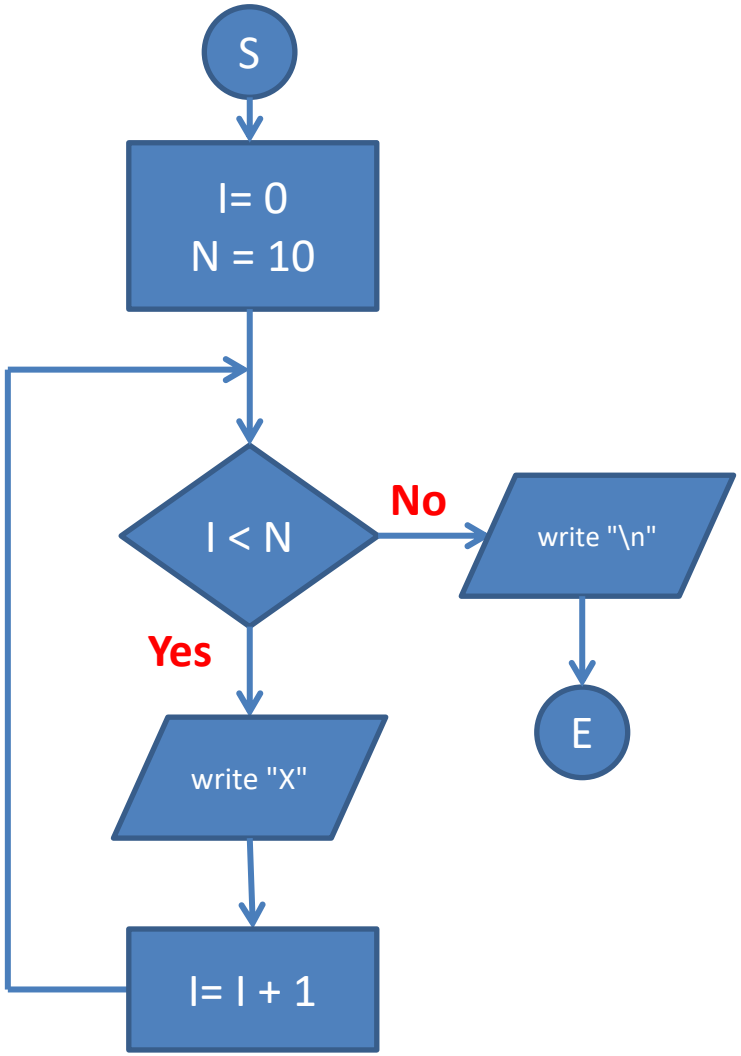


decision block

There are also other marks that we will not use yet.

Examples

Flowchart:

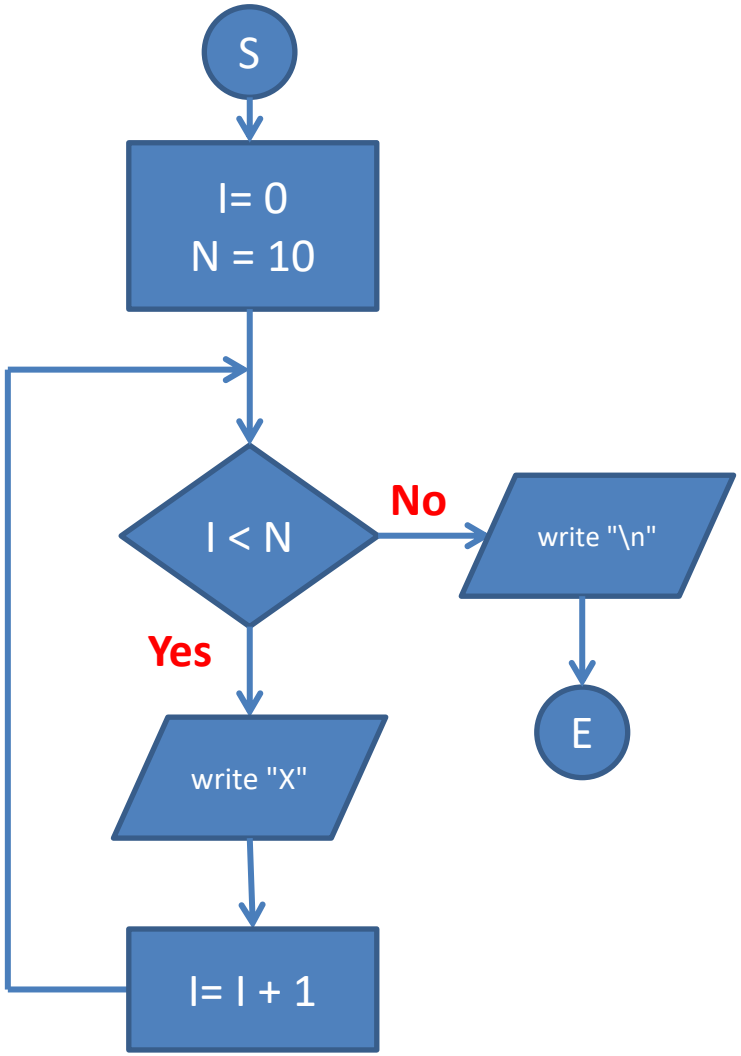


Word description:

1. Insert value 0 into variable I.
2. Insert value 10 into variable N.
3. Is the value of variable I less than N?
   YES - continue with point 4
   NO - continue with point 7
4. Print the X character.
5. Increase the value of variable I by one.
6. Continue with point 3.
7. Print the end of the line.
8. End

Examples

Algorithm



Bash Script

```
#!/bin/bash

I=0
N=10

while [ $I -lt $N ]; do
    echo -n "X"
    ((I=I+1))
done
echo ""
```

Result:

```
$ ./my_script
XXXXXXXXXX
$
```

Data Structures

Data structures are used to store data. The basic data structures include:

a) **variable**
b) field
c) record
d) object

Variable is a **named location** in memory that **contains a value**. Each variable is of defined **type** which restricts possible operations with the variable. The type of a variable can be specified when creating the variable (explicit type) or it can be specified only when using the variable (implicit type). Variable type affects how data is stored in computer memory.

Examples:

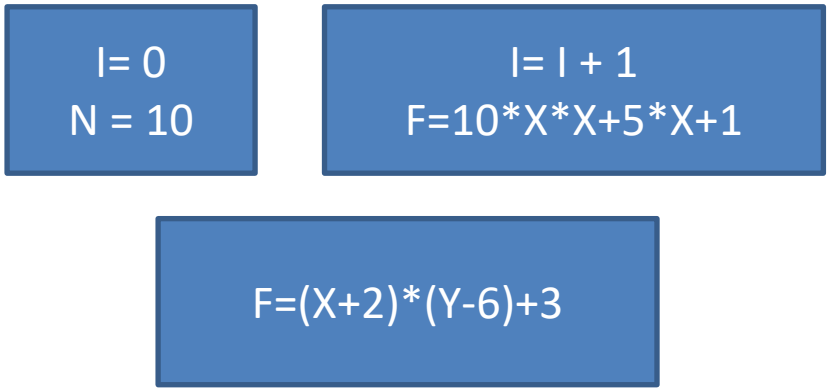
`A="test value"`	text (string)
`B=5`	integer
`C=10.458`	real number (floating point number)
`D=B+C`	
~~`E=A+B`~~	

Operations

Basic operations:

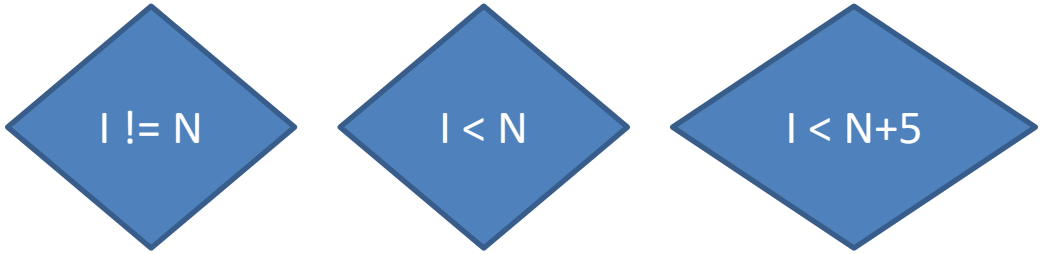
=	assignment
+	addition
-	subtraction
*	multiplication
/	division

processing block



Logical operations:

==	equal
!=	not equal
<	smaller than
<=	less than or equal
>	larger than
>=	greater than or equal



decision block

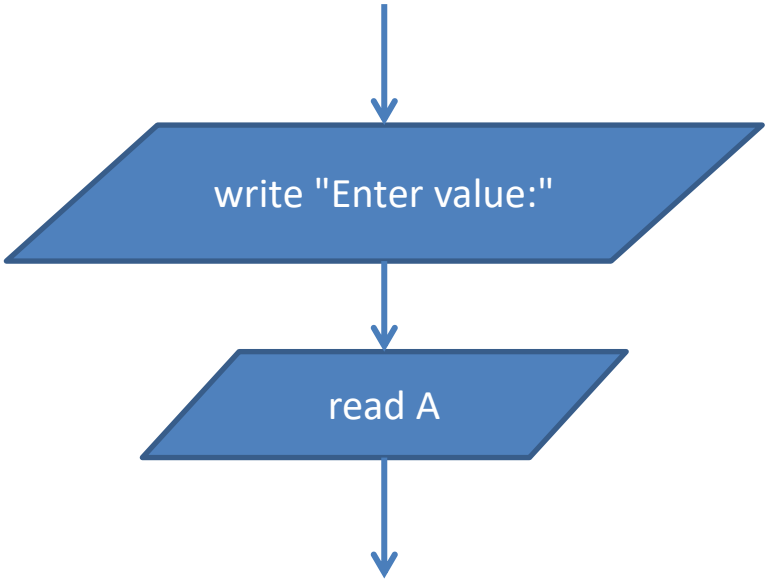
Input

Program **input** can be information entered by user from the terminal or can be redirected from a file, or from another program using a pipe.

Basic operations (pseudocode):

`read var` load a value into a variable *var*

Example:



Program writes a query to the user and expects input, which is loaded into the variable A after entering it by the user.

The input defined in this way reflects the basic possibilities of the bash language. I do not recommend using other input options at this stage.

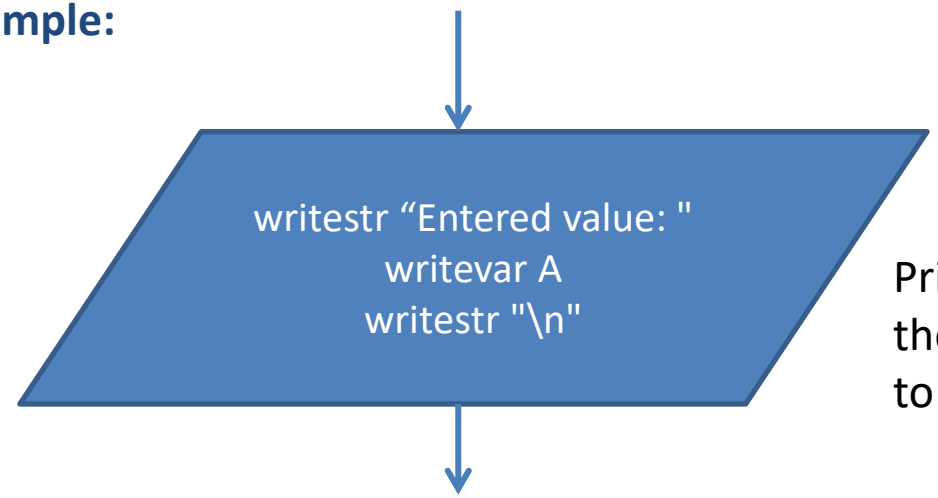
Output

Output is the terminal. The terminal behaves like a printer in which a character that has already been printed cannot be taken back. In addition, the printer can go to the beginning of a new line by typing a \n character into the terminal.

Basic operations (pseudocode):

`writestr "string”`	prints characters (string) stated in quotation marks
`writevar var`	prints the value of the variable var

Example:



Prints the text "Entered value:" followed by the value of variable A. The cursor will move to a new line.

The output defined in this way reflects the basic possibilities of the bash language. I do not recommend using other output options at this stage.

Homework

- **Algorithmization**

Homework - Instructions

1. **Create a flowchart for one from the following assignments**. In the flowcharts, use only the marks and operations, including I/O, that are listed in this presentation.
2. Draw the flowcharts in an appropriate software:
   - e.g., program **dia** which is available on the WOLF cluster (or you can install it on a VM with Ubuntu: $ sudo apt-get install dia)
3. Insert the resulting diagrams into the Homework vault **Algorithm** in pdf format. The file name will be in the following format:

   SurnameL05X.pdf

   where X is the chosen task. Files that are named differently will not be checked (automatic name changes made by IS are allowed).
4. Deadline for submission is **January 24, 2021 11:59 PM**.

Recommendation: For cycles, place the decision block before the processing block.

Task A

In the form of a flowchart, describe an algorithm that writes a square of **X** characters to a device of terminal type (printer). Side length of the square is entered by the user.

```
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
x x x x x x x x x x
```

Task B

In the form of a flowchart, describe an algorithm that writes a right triangle with characters to the output device of the terminal type (printer) **X**, so that one perpendicular is located at the top and the other on the left. The length of the perpendicular is entered by the user.

```
X X X X X X X X X X
X X X X X X X X X
X X X X X X X X
X X X X X X X
X X X X X X
X X X X X
X X X X
X X X
X X
X
```

Task C

In the form of a flowchart, describe the algorithm that writes a right triangle with characters to the output device of the terminal type (printer) **X**, so that one perpendicular is located at the bottom and the other on the left. The length of the perpendicular is entered by the user.

```
X
X X
X X X
X X X X
X X X X X
X X X X X X
X X X X X X X
X X X X X X X X
X X X X X X X X X
X X X X X X X X X X
```