Programming I (PROG1 119/2) notes

Bachelor in Computer Science and Engineering

First year - First semester

Contents

1 Console Input and Output

1.1 A C program structure

Let's review the structure and characteristic of simple program:

Listing 1: *Simple Hello World program in C.*

```
#include<stdio.h>
int main()
{
    printf("Hello\n");
    printf("World\n");
    return 0;
}
```

The *#include* directive instruct the compiler to read the header file *stdio.h* enclosed by angle brackets.

> **Definition 1.1.** Pre-processor directives are lines in the program that begin with the has key, #. They are executed before the compilation starts.

Header file *stdio.h* contains input and output function prototypes. A h-file corresponds to a header file and contains definitions of new variable types and function prototypes.

All C programs must have the *main()* function. This function has one or several statements enclosed by curly brackets. The program starts its execution in the first statement of this function. The tpe of the main function is an integer, *int*.

> **Definition 1.2.** A statement is a command written in the program that instructs it to take a specific action. In the previous example, the *printf* statement displays a word on the screen. A C program is made up of a series of statements each of them ends with a semicolon (;).

A program has a sequential execution order starting from the first statement.

In this case, the program has three statements. The first two statements display two words in different lines on the screen. The statement *printf* prints the string enclosed by double quotes on the screen.

The last statement in the main function returns the 0 value. 0 in the *return* statement represents the returned value of the *main()* function. 0 is used as a convention to mean that the function ends without errors. A value different from 0 informs the type of error generated during the execution of the function.

C is a case sensitive language. That is, *Printf* is not correct because it is not the same as *printf*.

1.2 Information output with printf

The function *printf* performs formatted ouputs. It can write data on the screen in various formats under your control. It can operate with any of the built-in data types that will be discussed in the next weeks like *strings*, *characters* and any type of *numbers*. Its prototype is in the header file *stdio.h* which has to be included in the program using the *#include* preprocessor directive.

Listing 2: *Another simple Hello World program.*

```
#include<stdio.h>
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

In this simple program, the *printf* function displays on the screen the sentence enclosed by double quotes. The *\n* character at the end of the sentence is the line break character. That is, any further *printf* will display in the next line of the screen.

The *fprintf* function is also used to display information. The equivalent function would be

Listing 3: *The fprintf function.*

```
fprintf(stdout, "Hello world\n");
```

Here, *stdout* represents the standart output device, the screen. It can be modified to specify a new output device like a file.

1.2.1 Special characters

The previous program displays the sentence *Hello world* without the double quotes. The double quote is a special character that can be displayed using the escape sequence \¨. The following table shows the most frequent special characters and their corresponding escape sequence.

Special character	**Escape sequence**
alert (beep)	\a
backslash (\)	\\
backspace	\b
double quote	\(dq)
percentage (%)	%%
horizontal tab	\t
vertical tab	\v
new line	\n
single quote	\(sq)
question mark	\?

Table 1: *Most frequent special characters and their corresponding escape sequence.*

The following statement

```
printf("My favorite character is %%\a\nbut I also like \"\\\"\n");
```

displays:

My favorite character is %
but I also like "\"

1.2.2 Colors and styles

If we want the text to be displayed in a color different from black the following codes have to be added right before the text.

Color	**Code**
Red	\033[31m
Green	\033[32m
Yellow	\033[33m
Blue	\033[34m
White	\033[37m
Reset	\033[0m

Table 2: *Color codes.*

If we want the text to be displayed using a different style, the following codes have to be placed right before the text.

Style	**Code**
Boldface	\033[1m
Italic	\033[3m
Underline	\033[4m
Strikethrough	\033[9m
Reset	\033[0m

Table 3: *Codes for different text styles.*