Computer science 2

Summer semester 2021

Prof. Dr. T. Grust, B. Dietrich, C. Duta, D. Hirn, T. Fischer

Exercise sheet 1 (April 28, 2021)

Submission by: Wednesday, May 5th, 2021, 2 p.m.

Relevant videos up to and including:
Computer Science 2 - Chapter 2 - Video # 12
https://tinyurl.com/Informatik2-SS2021

Exercise 1: [5 points] (Submission: arg-eval-order.c0)

Write a C0 program whose output in the terminal (see the functions in the conio library) you can clearly see that the arguments e1, e2 and e3 in a function call f (e1, e2, e3) in the Order e1, e2, e3 are evaluated. The function f is supposed to be any function with three parameters be.

Exercise 2: [15 points] (Submission: printbitsfit.c0)

Write a function **int** printbitsfit (**int** x) which, similar to the function **void** printbits (**int** x) from the preceding reading, which outputs bits of x in the terminal using printchar (...) . The function should output all Suppress preceding 0 bits. The return value of the function specifies the number of characters that the Representation of x in the terminal. The function printbitsfit (...) has a side effect and a return value.

Build in at least three function calls to printbitsfit (...) in your main () function , which the functionality make the function clear.

Note: Uses the function **bool** is_bit_set (**int** x, **int** n) known from the lecture .

Examples: Here are some example calls for printbitsfit (...) :

```
-> printbitsfit (0); printchar ('\n');
1 (int)
0
(void)
-> printbitsfit (42); printchar ('\n');
6 (int)
101010
(void)
-> printbitsfit (5249); printchar ('\n');
13 (int)
1010010000001
(void)
```

Exercise 3: [7 points] (Submission: bitwidth.c0)

Write a function **int** bitwidth () . This function determines how many bits the language C0 uses internally Representation of values of the type **int is** used (currently the function should therefore return the result 32).

Writes the function assuming that the number of bits for the type **int will** change in the future could.

Build your main () function so that when you run the compiled program, the return value is output by bitwidth () on the terminal.

Note: Use the bit operators that we got to know in the lecture. One implementation of the form **return** 32; is worth exactly 0 points.

Exercise 4: [7 points] (Submission: table.c0)

Write a C0 program that outputs a table on the terminal for each combination of values
a ∈ {−10, −9, −8, ..., 0, ..., 8, 9, 10} and b ∈ {−5, −3, −1, 1, 3, 5} the following five Entries contains:

a b from from (a / b) * b + (a% b)

Here you can see an excerpt from the table that the program should output:

a	b	from	from	(a / b) * b + (a% b)
-10	-5	2	-1	-10
-10	-3	3rd	-1	-10
-10	-1	10	0	-10
-10	1	-10	0	-10
-10	3rd	-3	-1	-10
-10	5	-2	0	-10
-9	-5	1	-4	-9
...				
10	5	2	0	10

Note: The output of a tabulator using printchar('\t') between the entries of a line helps with the formatting of the table.

Exercise 5: [6 points] (Submission: huh.txt)

When searching in old folders on your computer you will find a C0 program, the following uncomfortable-defined function:

```
1   int huh ( int x) {
2       int c = 0;
3rd
4th     while (x! = 0) {
5           c = c + 1;
6th         x = x & x - 1;
7th     }
8th
9       return c;
10  }
```

1. [3 points] Applies huh to a series of arguments to build a guess what the Function. Briefly explain your assumption in your own words. (max. 280 characters) and saves this in the plain text file huh.txt .
2. [3 points] Warning: tricky part of the task! Explains how and why in detail and as precisely as possible huh works at all. Add this explanation to the file huh.txt .

   Note: In your explanation, pay particular attention to line 6 in the code.