ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecturer: Prof. B. Gärtner

Informatik für Mathematiker und Physiker HS16

Exercise Sheet 6

Submission deadline: 15:15 - Tuesday 1st November, 2016

Course URL: http://lec.inf.ethz.ch/ifmp/2016/

Assignment 1 - Evaluating Functions (4 points)

[Similar to Exam Summer 2015, ex. 3]

Consider the following program:

```
#include <iostream>

void func_a (int i) {
    ++i;
}

int func_b (int a, int b) {
    return ++a + --b;
}

bool func_c (int x) {
    x = func_b(3, x);
    x = func_b(3, x);
    x = func_b(3, x);
    return x > 10;
}

int func_d (int j) {
    int tmp;
    if (j > 5)
        tmp = j - 5;
    return tmp;
}

int main () {
    // insert code from subtasks here
    return 0;
}
```

a) What does the program above output if the following piece is inserted?

```
int i = 5;
func_a(i);
std::cout << i << "\n";
```

b) What does the program above output if the following piece is inserted?

```
int x = 5;
int y = func_b(3, x++);
std::cout << "x=" << x << " y=" << y << "\n";
```

c) What does the program above output if the following piece is inserted?

```
if (func_c(0))
    std::cout << "Yes!\n";
else
    std::cout << "Nope!\n";
```

d) What does the program above output if the following piece is inserted?

```
std::cout << func_d(5) << "\n";
```

Assignment 2 - Writing Functions (3 points)

In this exercise you are given 3 mini-tasks that are intended to let you play around with functions. Please keep in mind that once again there are multiple ways to solve each subtask. **For all of your functions write suitable PRE- and POST-conditions.** The Codeboard-templates already provide `main`-functions that take care of the whole user interaction.

a) Write a function `is_odd` that takes an `int a` and returns `true` if and only if `a` is not divisible by 2, and else returns `false`.

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
Input an integer: 122
It is even
```

```
Input an integer: 123
It is odd
```

Submission: https://codeboard.ethz.ch/ifmp16E6T2a

b) Write a function `nand` that takes two `bool`s `a` and `b`, and returns `false` if and only if `a` and `b` are both `true`. This has the following truth table:

a	b	nand(a, b)
false	false	true
false	true	true
true	false	true
true	true	false

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
Input a: 1
Input b: 0
a NAND b is: 1
```

Submission: https://codeboard.ethz.ch/ifmp16E6T2b

c) Write a function `output_range` that takes two `int`s `lower` and `upper` and outputs the following into the terminal:

if `lower < upper`: `lower lower+1 ... upper-1`
if `lower == upper`: nothing

The case where `lower > upper` shall be treated as an error for this exercise. (For technical reasons the program containing your function should furthermore write a `*` before and after the range. This is already done for you in the given Codeboard link.)

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
Input lower: -3
Input upper: 4
Range: * -3 -2 -1 0 1 2 3 *
```

```
Input lower: 2
Input upper: 2
Range: * *
```

Submission: https://codeboard.ethz.ch/ifmp16E6T2c

Assignment 3 - Twin Primes (4 points)

[Skript-Aufgabe 85]

a) Write a function

```
// POST: return value is true if and only if n is prime
bool is_prime (unsigned int n);
```

b) Use your function `is_prime` in a program `twinprimes.cpp` to count the number of *twin primes* in the range $\{2, \ldots, n\}$ (two up to n) where n is a value obtained from the user. A twin prime is a pair of numbers $(i, i+2)$ both of which are prime. (Both of the primes have to be in the range $\{2, \ldots, n\}$.)

Note: Depending on your implementation, the program can be quite slow for large n. In a later lecture you will see an algorithm which is very fast.

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
1000
Number of twin primes: 35
```

Submission: https://codeboard.ethz.ch/ifmp16E6T3

Assignment 4 – Perfect Numbers (4 points)

A perfect number is an integer which is equal to the sum of all of its proper divisors (a divisor not equal to the number itself). For example

$$6 = 1 + 2 + 3$$

is a perfect number. Write a program `perfect_numbers.cpp` which reads a given integer n from the user and outputs each perfect number between 1 and n.

Write your program in a way such that it uses functions in a way that makes the program easy to read and easy to understand. Argue for every function why you chose to use it.

I/O-Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2016/codeboard.html)

```
99
The following numbers are perfect: 6 28
```

Submission: https://codeboard.ethz.ch/ifmp16E6T4

Challenge – Perpetual Calendar (8 points) [Skript-Aufgabe 88]

You may for example know that the Berlin wall came down on November 9, 1989, but what was the weekday? (It was a Thursday.) Or what is the weekday of the 1000th anniversary of the Swiss confederation, to be celebrated on August 1, 2291? (Quite adequately, it will be a Saturday.)

If you like questions like these and want to write a program that is longer than those you wrote above, take up our challenge! This week's challenge is exercise 88 from the script.

We remark that there are (even short) magic formulas for computing weekdays in a perpetual calendar. However, the goal of this assignment is that you come up with a solution yourself; after all, a careful understanding of the problem is the basis of any magic formula.