ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecturer: Prof. B. Gärtner

Informatik für Mathematiker und Physiker HS15

Exercise Sheet 6

Submission deadline: 15:15 - Tuesday 27th October, 2015

`Course URL`: http://lec.inf.ethz.ch/ifmp/2015/

Assignment 1 (3 points)

a) Parenthesize the following expressions according to the rules of associativity and precedence.

(i) `5 / 4 / 2.0 / 2 < 0.3125`
(ii) `f < 3.5f || 5 / 4 / f == 1.25f && ++f < 1`
(iii) `3 / 2 > 1.0 && d != 3 / 4`

b) Evaluate the expressions from part a) step by step. You may assume that the variable `f` is of type `float` and has initial value `0.0f`. And `d` is of type `double` and has initial value `0.75`.

Assignment 2 - Skript-Aufgabe 85 (4 points)

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.

Judge Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2015/judge_boxes.html)

```
Input n =? 2
Number of twin primes: 0
```

```
Input n =? 5
Number of twin primes: 1
```

```
Input n =? 1000
Number of twin primes: 35
```

Submission: https://challenge.inf.ethz.ch/team/websubmit.php?cid=5&problem=MP15062

Assignment 3 (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.

Judge Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2015/judge_boxes.html)

```
Find perfect numbers up to n =? 99
The following numbers are perfect.
6 28
```

```
Find perfect numbers up to n =? 1
The following numbers are perfect.
```

Submission: https://challenge.inf.ethz.ch/team/websubmit.php?cid=5&problem=MP15063

Assignment 4 - Skript-Aufgabe 88 (6 points)

A *perpetual calendar* can be used to determine the weekday (Monday, . . . , Sunday) of any given date. 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.)

a) The task is to write a program that outputs the weekday (Monday, . . . , Sunday) of a given input date.

   Identify a set of subtasks to which you can reduce this task. Such a set is not unique, of course, but all individual subtasks should be small (so small that they could be realized with few lines of code). It is of course possible for a subtask in your set to reduce to other subtasks. (Without giving away anything, one subtask that you certainly need is to determine whether a given year is a leap year).

b) Write a program `perpetual_calendar.cpp` that reads a date from the input and outputs the corresponding weekday. The range of dates that the program can process should start no later than January 1, 1900 (Monday). The program should check whether the input is a legal date, and if not, reject it.

   To structure your program, implement the subtasks from a) as functions, and put the program together from these functions. To implement this program you can for example apply the Stepwise Refinement principle.

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.

Judge Examples (Explanation: http://lec.inf.ethz.ch/ifmp/2015/judge_boxes.html)

```
Compute weekday of date (day/month/year) for
day =? 1
month =? 8
year =? 2291
Saturday
```

```
Compute weekday of date (day/month/year) for
day =? 1
month =? 1
year =? 1900
Monday
```

```
Compute weekday of date (day/month/year) for
day =? 32
month =? 12
year =? 2000
Illegal date
```

Submission: https://challenge.inf.ethz.ch/team/websubmit.php?cid=5&problem=MP15064

Challenge – Skript-Aufgabe 93 (8 points)

Note: To reduce the large extra-workload some teaching assistants had with the corrections of the Challenge exercises, we allowed them to send you the sample solution to the Challenges instead of a fully-fledged individual correction.