

**Informatik für Mathematiker und Physiker****Serie 9****HS 10**URL: [http://www.ti.inf.ethz.ch/ew/courses/Info1\\_10/](http://www.ti.inf.ethz.ch/ew/courses/Info1_10/)**Skript-Aufgabe 92 (4 Punkte)**

Find pre- and postconditions for the following functions.

```
a) double f (const double i, const double j, const double k)
{
    if (i > j)
        if (i > k)
            return i;
        else
            return k;
    else
        if (j > k)
            return j;
        else
            return k;
}
```

```
b) double g (const int i, const int j)
{
    double r = 0.0;
    for (int k = i; k <= j; ++k)
        r += 1.0 / k;
    return r;
}
```

**Skript-Aufgabe 94 (4 Punkte)**What is the output of the following program, depending on the input number  $i$ ? Describe the output in mathematical terms, ignoring possible over- and underflows.

```
#include<iostream>

int f (const int i)
{
    return i * i;
}

int g (const int i)
{
    return i * f(i) * f(f(i));
}

void h (const int i)
{
    std::cout << g(i) << "\n";
}
```

```

int main()
{
    int i;
    std::cin >> i;
    h(i);

    return 0;
}

```

### Skript-Aufgabe 101 (4 Punkte)

Write a program `unique.cpp` that implements and tests the following function.

```

// PRE: [first, last) is a valid range and describes a sequence
//       of elements that are sorted in nondecreasing order
// POST: the return value is true if and only if no element
//       occurs twice in the sequence
bool unique (const int* first, const int* last);

```

### Skript-Aufgabe 105 (4 Punkte)

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. An example run of the program might look like this.

```

    day =? 13
    month =? 11
    year =? 2007
    Tuesday

```

To structure your program, implement the subtasks from a) as functions, and put the program together from these functions.

Die **Aufgaben 110** und **111** aus den Vorlesungsunterlagen sind die **Challenge Aufgaben** und geben jeweils 8 Punkte, wenn sie vollständig gelöst werden.

**Abgabe:** Bis 30. November 2010, 15.15 Uhr.