Dirk Praetorius,
Thomas Führer

Sommersemester 2014
24.03.2014

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 4

Aufgabe 4.1. Write a *non-recursive* function `binomial` which computes the binomial coefficient $\binom{n}{k}$. Use an appropriate loop and the identity $\binom{n}{k} = \frac{n\cdot(n-1)\cdots(n-k+1)}{1\cdot 2\cdots k} = \frac{n}{1}\cdot\frac{n-1}{2}\cdots\frac{n-k+1}{k}$. Additionally, write a main program that reads in the values $k, n \in \mathbb{N}_0$ with $k \le n$ and prints out $\binom{n}{k}$.

Aufgabe 4.2. Use a recursive function for this exercise! For $x > 0$, the recursively defined sequence

$$x_1 := \frac{1}{2}(1+x), \quad x_{n+1} := \frac{1}{2}\Big(x_n + \frac{x}{x_n}\Big) \quad \text{for } n \ge 1$$

converges towards $\sqrt{x}$. Write a function `sqrt_` which computes for given $x > 0$ and $\tau > 0$ the *first* element x_n of the sequence that satisfies

$$\frac{|x_n - x_{n+1}|}{|x_n|} \le \tau \quad \text{or} \quad |x_n| \le \tau.$$

Moreover, write a main program which reads in x and τ, computes the approximation x_n of $\sqrt{x}$ and compares it to the exact value, i.e. prints out the absolute error $|x_n - \sqrt{x}|$.
Hint: You can use the function `sqrt` from the math library to compute the exact value $\sqrt{x}$. For the computation of the absolute value $|x|$ of a real number x, you can use the function `fabs` from the math library.

Aufgabe 4.3. You place your money with your trusted bank for a fixed annual percentage rate. Write a function `capital` which computes your capital after $n \in \mathbb{N}$ years for a fixed annual percentage p (in percent %), and your starting capital $x \in \mathbb{R}_{\ge 0}$. The function should print out your money as follows

```
Year    Capital
====    =======
0      1000.00
1      1010.00
2      1020.10
3      1030.30
..      .......
10      1104.62
```

For this example holds $p = 1$, $n = 10$, and $x = 1000.00$. Furthermore, write a function `runtime` which computes how long (at least) you have to wait to increase your starting capital x to $x_{\max}$ for a fixed percentage p. The function reads in x, p, and $x_{\max}$. Additionally, write a main program that tests both functions. How long does it take to be a millionaire, if you invest $x = 1000$ with a fixed percentage $p = 4$?

Aufgabe 4.4. Write a function `double powN(double x, int n)` which computes x^n for all exponents $n \in \mathbb{Z}$ and $x \in \mathbb{R}$. It holds $x^0 = 1$ for all $x \in \mathbb{R}$. For $n < 0$ use $x^n = (1/x)^{-n}$. Moreover, $0^n = 0$ for $n > 0$. The term 0^n for $n < 0$ is not defined. In that case, the function should return the value `0.0/0.0`. You must not use the function `pow` from the math library.

Aufgabe 4.5. Compute the sum $\sum_{j=1}^{n}(-1)^j/j$. From a numerical point of view, it is attractive to first sum the negative and positive contributions separately and then return the sum of these partial sums. Why could this be advantageous? Write a function `sum` which realizes this. Furthermore, write a main programm which reads in $n \in \mathbb{N}$ and prints out $\sum_{j=1}^{n}(-1)^j/j$.

Aufgabe 4.6. Write a function `maxabs` which returns the very first entry x_j with largest absolute value of a given vector $x \in \mathbb{R}^n$, i.e. x_j satisfies $|x_j| = \max\{|x_i| : i = 1, \ldots, n\}$ and if $|x_i| = |x_j|$, then it holds $i \geq j$. Additionally, write a main program that reads in the vector x and calls the function `maxabs`. The vector x should be realized as a static array, where the length is given by a constant in the main program, but the function `maxabs` should be implemented for arrays with arbitrary lengths!

Aufgabe 4.7. Write a function `geometricMean` that computes and returns the geometric mean value

$$\overline{x}_{\text{geom}} = \sqrt[n]{\prod_{j=1}^{n} x_j}$$

of a given vector $x \in \mathbb{R}^n_{\geq 0}$ Furthermore, write a main program that reads in $x \in \mathbb{R}^n$ and computes the geometric mean value thereof. The length $n \in \mathbb{N}$ of the vector should be a constant in the main program, but the function `geometricMean` should be implemented for arbitrary lengths.

Aufgabe 4.8. The following code should compute the maximum entry of the given matrix, but the output is `5.0000`. Where is the error?

```
#include <stdio.h>

main() {
  double A[2][3] = { {1,2,3},{6,-4,5} };
  double max = A[0][0];

  int j=0, k=0;

  for(j=0; j<2; j=j+1) {
    for(k=1; k<3; k=k+1) {
      if(A[j][k] > max) {
        max = A[j][k];
      }
    }
  }
  printf("Maximum = %f\n",max);
}
```

Correct the program and extend it, such that also the minimum of all matrix entries is computed.