Dirk Praetorius,
Michele Ruggeri Wintersemester 2015/16
11.-13.11.2015

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 5

Aufgabe 5.1. For $p \in [1, \infty)$, the ℓ_p-norm on $\mathbb{R}^n$ is defined by

$$\|x\|_p := \Big(\sum_{j=1}^n |x_j|^p \Big)^{1/p}.$$

Write a function `pnorm`, which, given a vector $x \in \mathbb{R}^n$, its length n and $p \in [1, \infty)$, returns $\|x\|_p$. Write a main program which reads in $x \in \mathbb{R}^n$ and $p \in [1, \infty)$, and prints $\|x\|_p$. The dimension $n \in \mathbb{N}$ should be a constant in the main program, but the function `pnorm` should support arbitrary lengths n. Test your implementation for different values of p and fixed x. Which behavior do you observe for $p \to \infty$? Save your source code as `pnorm.c` into the directory `serie05`.

Aufgabe 5.2. 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 $|x_j| = \max\{|x_i| \,:\, i = 1, \dots, n\}$ 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. Save your source code as `maxabs.c` into the directory `serie05`.

Aufgabe 5.3. Write the function

```
nk = binomial(n,k,type)
```

to compute the binomial coefficient $\binom{n}{k}$ using three different approaches:

- applying the formula $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, which exploits a function for computing the factorial (`type=1`),
- using the expression $\binom{n}{k} = \frac{n\cdot(n-1)\cdots(n-k+1)}{k\cdot(k-1)\cdots 1}$ driven by a suitable loop (`type=2`),
- in recursive form, exploiting the formula $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ (`type=3`).

The function should support all three options. Write a main program, which reads n and k from the keyboard and returns the binomial coefficient. Speichern Sie den Source-Code unter `binomial.c` in das Verzeichnis `serie05`.

Aufgabe 5.4. Write a *nonrecursive*function `power`, which, given two real numbers $x > 1$ and $C > 0$, determines the smallest integer $n \in \mathbb{N}$ such that $x^n > C$. In your implementation, you are not allowed to use the function `log`. Write a main program, which reads in x and C and prints out the corresponding n. Save your source code as `power` into the directory `serie05`.

Aufgabe 5.5. The quotient sequence $(a_{n+1}/a_n)_{n\in\mathbb{N}}$ corresponding to the Fibonacci sequence $(a_n)_{n\in\mathbb{N}}$,

$$a_0 := 1, \quad a_1 := 1, \quad a_n := a_{n-1} + a_{n-2} \quad \text{für } n \geq 2,$$

converges towards the *golden ratio* $(1 + \sqrt{5})/2$. In particular, the difference sequence

$$b_n := \frac{a_{n+1}}{a_n} - \frac{a_n}{a_{n-1}}$$

converges towards 0. Write a *non-recursive* function `cauchy` that returns, for given $k \in \mathbb{N}$, the smallest $n \in \mathbb{N}$ such that $|b_n| \leq 1/k$. Moreover, write a main program that reads in $k \in \mathbb{N}$ and prints out the index $n \in \mathbb{N}$. Save your source code as `goldenRatio.c` into the directory `serie05`.

Aufgabe 5.6. The sine function can be represented as a series via

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

The corresponding n-th partial sum is given by

$$S_n(x) = \sum_{k=0}^{n} (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

Write a *nonrecursive* function `sin_`, which, given $x \in \mathbb{R}$ and $\varepsilon > 0$, returns the first value of $S_n(x)$ such that

$$|S_n(x) - S_{n-1}(x)|/|S_n(x)| \le \varepsilon \quad \text{or} \quad |S_n(x)| \le \varepsilon.$$

Then, write a main program, which reads $x \in \mathbb{R}$ and $\varepsilon > 0$ from the keyboard, calls the function and displays the computed value $S_n(x)$, as well as the value $\sin(x)$, the absolute error $|S_n(x) - \sin(x)|$ and the relative error $|S_n(x) - \sin(x)|/|\sin(x)|$ (provided $\sin(x) \neq 0$). Save your source code as `sin.c` into the directory `serie05`.

Aufgabe 5.7. 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 *nonrecursive* function `sqrt_new` 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 $|x_n - \sqrt{x}|$. Save your source code as `sqrt_new.c` into the directory `serie05`. Compare your implementation to your code from Exercise 3.7. Discuss the advantages and disadvantages of both implementations!

Aufgabe 5.8. 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.