Dirk Praetorius
Michele Ruggeri

Wintersemester 2014/15
09.12.2014

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 11

Aufgabe 11.1. Write a program that reads in a word (string) und checks if this word is a *palindrome*. A palindrome is a word which reads the same backward or forward, e.g., radar, level, madam. Save your source code as `palindrome.c` into the directory `serie11`.

Aufgabe 11.2. Write a structure data-type `cdouble`, which stores the real part a and the imaginary part b of a complex number $a + bi \in \mathbb{C}$ as `double` variables. Implement the functions `cdouble* newCDouble(double a, double b)`, `cdouble* delCDouble(cdouble* c)` as well as `setCDoubleReal`, `getCDoubleReal`, `setCDoubleImag` sowie `getCDoubleImag`. Divide your source code into the header file `cdouble.h` and the file `cdouble.c`, and save it into the directory `serie11`.

Aufgabe 11.3. Write functions `cadd, csub, cmul, cdiv`, which realize addition, subtraction, multiplication, and division of complex numbers. Furthermore, implement the function `double cnorm(cdouble* c)`, which computes the squared norm $|a + ib|^2 := a^2 + b^2$. Use the structure from Exercise 11.2 to store complex numbers and use the corresponding `get` and `set` functions. Additionally, write a main program that reads two complex numbers $w, z \in \mathbb{C}$ from the keyboard and displays $|w|^2$, $|z|^2$, $w + z$, $w - z$, $w \cdot z$ as well as w/z (if $z \neq 0$). Save your source code as `carithmetik.c` into the directory `serie11`.

Aufgabe 11.4. Write a structure `CMatrix` for the storage of $m \times n$-matrices $A \in \mathbb{C}^{m\times n}$ with complex entries. Use the structure `cdouble` from Exercise 11.2. Furthermore, write the functions `newCMatrix`, `delCMatrix`, `getCMatrixM`, `getCMatrixN`, `getCMatrixCoeff`, `setCMatrixCoeff`. Save your source code as `CMatrix.c` into the directory `serie11`.

Aufgabe 11.5. Write a structure `CPoly` for the storage of polynomials, where the coefficients are complex numbers, i.e., $p(x) = \sum_{j=0}^{n} a_j x^j$ with $a_j \in \mathbb{C}$. The structure should contain the degree $n \in \mathbb{N}$ and the coefficients $(a_0, \dots, a_n) \in \mathbb{C}^{n+1}$. Use the structure from Exercise 11.2. Moreover, implement the functions `newCPoly`, `delCPoly`, `getCPolyDegree`, `getCPolyCoefficient`, and `setCPolyCoefficient`. Save your source code as `cpoly.c` into the directory `serie11`.

Aufgabe 11.6. Write a function `addCpolynomials` that computes the sum $r = p + q$ of two complex polynomials p, q and returns r. Use the structure from Exercise 11.5. Moreover, write a main program that reads in two polynomials p, q and prints out the sum $r = p+q$. Save your source code as `addcpoly.c` into the directory `serie11`.

Aufgabe 11.7. Write a structure `CVector` for the storage of vector with complex coefficients. Use the structure `cdouble` from Exercise 11.2. Moreover, implement the functions `newCVector`, `delCVector`, `getCVectorLength`, `getCVectorEntry`, `setCVectorEntry`. Save your source code as `cvector.c` into the directory `serie11`.

Aufgabe 11.8. Write a function `CVectorVector`, which, given two complex vectors $x, y \in \mathbb{C}^n$, computes the scalar product $x \cdot y := \sum_{j=1}^{n} x_j \overline{y_j}$. Use the structure `CVector` from Exercise 11.2. Then, write a main program, which reads two complex vectors $x, y \in \mathbb{C}^n$ from the keyboard and displays the value of the scalar product $x \cdot y \in \mathbb{C}$. Save your source code as `CVectorVector.c` into the directory `serie11`.