

Übungen zur Vorlesung Einführung in das Programmieren für TM

Serie 3

Aufgabe 3.1*. Schreiben Sie eine Funktion `bin2dec`, die eine im Binärformat gegebene natürliche Zahl $0 \leq z < 256$ in eine Dezimalzahl umrechnet. Die Binärzahl werde als Vektor der Koeffizienten $a_i \in \{0, 1\}$ für $i = 0, \dots, 7$ dargestellt. Dann lässt sich der Wert der Binärzahl mittels $z = \sum_{i=0}^7 a_i 2^i$ berechnen. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Koeffizienten a_i eingelesen und der Wert von z als Dezimalzahl ausgegeben werden. Speichern Sie den Source-Code unter `bin2dec.c` in das Verzeichnis `serie03`.

Aufgabe 3.2*. Schreiben Sie eine Funktion `maxabs`, die von einem gegebenem Vektor $x \in \mathbb{R}^n$ das erste Element x_j mit maximalem Betrag berechnet und zurückgibt, d.h. $|x_j| = \max\{|x_i| : i = 1, \dots, n\}$ und für $|x_i| = |x_j|$ gilt $i \geq j$. Schreiben Sie ferner ein aufrufendes Hauptprogramm, das den Vektor x einliest und das Ergebnis von `maxabs` ausgibt. Die Länge des Vektors soll eine Konstante im Hauptprogramm sein, die Funktion `maxabs` ist für beliebige Länge zu implementieren. Speichern Sie den Source-Code unter `maxabs.c` in das Verzeichnis `serie03`.

Aufgabe 3.3*. Welchen Output liefert das folgende Programm und warum?

```
#include <stdio.h>

void square(double* x)
{
    double* y;
    *y=(*x)*(x);
}

main(){
    double x=2.1;
    square(&x);
    printf("x^2=%f\n", x);
}
```

Verändern Sie das Programm geeignet, sodass der Output den Erwartungen entspricht.

Aufgabe 3.4*. Schreiben Sie eine Funktion, die von einem gegebenem Vektor $x \in \mathbb{R}^n$ das Minimum $\min_{j=1}^n x_j$, das Maximum $\max_{j=1}^n x_j$ und den Mittelwert $\frac{1}{n} \sum_{j=1}^n x_j$ berechnet und zurückgibt. Schreiben Sie zwei Versionen der Funktion mit folgenden Deklarationen:

```
double* minmaxmean1(double* x, int n);
void minmaxmean2(double* x, int n, double* min, double* max, double* mean)
```

Ferner schreibe man ein aufrufendes Hauptprogramm, das n und x einliest und die von `minmaxmean1` und `minmaxmean2` berechneten Kenngrößen ausgibt. Speichern Sie den Source-Code unter `minmaxmean.c` in das Verzeichnis `serie03`.

Aufgabe 3.5. Schreiben Sie eine `void`-Funktion `dec2bin`, die zu einer natürlichen Zahl $0 \leq z < 256$ die Binärdarstellung berechnet und ausgibt. Es sollen die Koeffizienten $a_i \in \{0, 1\}$ für $i = 0, \dots, 7$ ermittelt werden, sodass $z = \sum_{i=0}^7 a_i 2^i$ gilt. Anschließend soll die Binärdarstellung in einem geeignetem Format ausgegeben werden. Beispielsweise gebe die Funktion für $z = 77$ die Zeichenfolge `0 1 0 0 1 1 0 1` aus. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem z eingelesen und `dec2bin` aufgerufen werden.

Aufgabe 3.6. Gegeben sei eine endliche Zahlenfolge x (dynamisches Array vom Typ `int`) und eine Zahlenkombination y (dynamisches Array vom Typ `int`). Man schreibe eine Funktion `check`, die überprüft, ob die Zahlenkombination y in der Zahlenfolge x vorkommt (`return 1`) oder nicht (`return 0`). Zusätzlich schreibe man ein aufrufendes Hauptprogramm, in dem die Länge der Felder x und y sowie deren Werte eingelesen werden und das Resultat von `check` geeignet ausgegeben wird.

Aufgabe 3.7. Schreiben Sie eine Funktion `maxcount`, die von einem Vektor $x \in \mathbb{R}^n$ das Maximum zurückliefert und die Anzahl, wie oft dieses im Vektor vorkommt. Schreiben Sie ferner ein Hauptprogramm, das die Länge des Vektors x sowie seine Einträge von der Tastatur einliest und die von `maxcount` ermittelten Größen ausgibt.

Aufgabe 3.8. Die Fibonacci-Folge ist definiert durch $x_0 := 0$, $x_1 := 1$ und $x_{n+1} := x_n + x_{n-1}$. Schreiben Sie eine nicht-rekursive Funktion `fibonacci(k)`, die zu gegebenem Index k das Folgenglied x_k berechnet und zurückgibt. Schreiben Sie ferner ein Hauptprogramm, das k von der Tastatur einliest und x_k am Bildschirm ausgibt.

Aufgabe 3.9. Ein Tripel $(x, y, z) \in \mathbb{N}^3$ natürlicher Zahlen heißt *pythagoräisches Zahlentripel*, falls $x^2 + y^2 = z^2$ gilt. Das wohl bekannteste Beispiel ist $(3, 4, 5)$. Offensichtlich gelten $z > \max\{x, y\}$ sowie $x \neq y$ und ohne Beschränkung der Allgemeinheit ferner $x < y$. Schreiben Sie eine `void`-Funktion `pythagoras`, die zu gegebener Schranke $n \in \mathbb{N}$ alle pythagoräischen Zahlentripel mit $x < y < z \leq n$ bestimmt und ausgibt. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Schranke n eingelesen und `pythagoras` aufgerufen wird.

Aufgabe 3.10. Gegeben sei ein Polynom $p(x) = \sum_{j=0}^n a_j x^j$ in Form seines Koeffizientenvektors $a = (a_0, \dots, a_n) \in \mathbb{R}^{n+1}$. Schreiben Sie eine Funktion `evalpolynomial`, die für gegebenen Koeffizientenvektor a und Auswertungspunkt x den Funktionswert $p(x)$ berechnet. Die Funktion `pow` zur Berechnung von x^j soll *nicht* verwendet werden. Schreiben Sie eine Funktion, die möglichst nur *eine* Schleife verwendet. Der Grad $n \in \mathbb{N}$ des Polynoms soll eine Konstante im Hauptprogramm sein, die Funktion `evalpolynomial` soll aber beliebigen Grad zulassen. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem die Koeffizienten a_j sowie der Auswertungspunkt x eingelesen werden und $p(x)$ ausgegeben wird.