

++
Nachklausur 06/07

Aufgabe 1:

(4 Punkte)

Schreiben Sie ein vollständiges C++-Programm, das eine Temperatur von Grad Celsius in Grad Fahrenheit umrechnet. Es gilt:

$$T_{\text{Fahrenheit}} = 1.8 \cdot T_{\text{Celsius}} + 32.$$

Ihr Programm soll die Temperatur in Celsius vom Benutzer erfragen und dann den Wert in Fahrenheit ausgeben. Es sollen Eingaben und Ausgaben mit Nachkommastellen möglich sein.

Aufgabe 2:

(5 Punkte)

Schreiben Sie eine C++-Funktion `MaximumNorm`, die das betragsgrößte Element eines Feldes ermittelt und als Funktionswert diesen Betrag zurückgibt. Die Funktion soll als Argument die Länge des Feldes n und das Feld von `double`-Variablen selbst erhalten. Die gültigen Feldindizes bewegen sich zwischen 0 und $n - 1$.

Alle benötigten Variablen sollen lokal definiert oder Argumente der Funktion sein – verwenden Sie keine globalen Variablen.

Aufgabe 3:

(12 Punkte)

Gegeben sei folgendes Programm, das eine Klasse für dreidimensionale Vektoren benutzt:

```
#include <iostream>
#include <cmath>
using namespace std ;

// ... hier fehlt ein Programmteil ...

int main()
{
    vec3d v1 (1.0, 2.0, 2.0) ;
    vec3d v2 (0.0, 1.0, 0.0) ;
    vec3d v3 ;
    v3 = kp(v1,v2) ; // Kreuzprodukt
    cout << "Laenge v3 vorher: " << sqrt(v3*v3) << endl ;
    v3.norm() ; // v3 auf Laenge 1 normieren
    cout << "Laenge v3 nachher: " << sqrt(v3*v3) << endl ;
    cout << "Ist v3 senkrecht auf (v1,v2)? " << v3*v1 << " " << v3*v2 << endl ;
}
```

Hierbei bezeichnet: 'kp' Kreuzprodukt zweier Vektoren; '*' Skalarprodukt zweier Vektoren; 'norm' Normierung des Vektors selbst auf Länge 1 per Euklidnorm.

Schreiben Sie die benötigte Klasse `vec3d` und alle benötigten Konstruktoren, Mitgliedsfunktionen und freien Funktionen, die in dem gegebenen Hauptprogramm aufgerufen werden. Die Vektorkomponenten selbst sollen von außen unzugängliche `double` Variablen sein. Die Definition der Mitgliedsfunktionen soll jeweils außerhalb der eigentlichen Klassendefinition erfolgen.

Aufgabe 4:**(7 Punkte)**

Setzen Sie an der markierten Stelle in folgendem Programm die letzte Ziffer Ihrer Matrikelnummer ein. Schreiben Sie dann *vollständig* auf, was von dem Programm ausgegeben wird.

```
#include <iostream>
using namespace std ;

int f(int & z) {
    z += 1 ;
    cout << "UP1: " << z << endl ;
    return z*3 ;
}

int f(double z) {
    z += 1 ;
    cout << "UP2: " << z << endl ;
    return int(z*2) ;
}

int f(int r, int s) {
    if (s==0) return 0 ;
    else return r + f(r+1,s-1) ;
}

int main() {
    int i= 12 ; // <- setzen Sie hier bitte die letzte Ziffer
                // Ihrer Matrikelnummer ein.
    double p=i*1.1 ;

    int j=f(i) ;
    cout << "HP1: " << i << " " << j << endl ;

    j=f(p) ;
    cout << "HP2: " << p << " " << j << endl ;

    cout << "HP3: " << f(6,3) << endl ;
}
```

Aufgabe 5:**(12 Punkte)**

Eine n -stellige positive Zahl ($1 \leq n \leq 8$) heie *rckwrts-summen-schn*, wenn die Summe der Zahl und der Zahl rckwrts gelesen nur aus identische Ziffern besteht.

Bsp: 2154 ist rckwrts-summen-schn, weil: $2154 + 4512 = 6666$.

19 ist nicht rckwrts-summen-schn, weil: $19 + 91 = 110$.

222

Schreiben Sie ein vollstndiges C++-Programm, das Folgendes leistet:

- Eine long-Zahl wird von der Tastatur eingelesen. Sie brauchen nicht zu berprfen, ob die eingegebene Zahl positiv ist und ob sie weniger als neun Stellen hat.
- Ausgabe der Zahl rckwrts gelesen.
- Ausgabe, ob die Zahl *rckwrts-summen-schn* ist oder nicht.