

(4 Punkte)

### Aufgabe 1: Tribonacci-Zahlen

(a) Die Folge  $(a_n)_{n \in \mathbb{N}_0}$  der sogenannten Tribonacci-Zahlen ist definiert durch

$$a_0 = a_1 = a_2 = 1 \quad \text{und} \quad a_{n+3} = a_{n+2} + a_{n+1} + a_n \quad \text{für } n \geq 0.$$

Schreiben Sie eine rekursive Funktion `tribo`, die die  $n$ .te Tribonacci-Zahl berechnet. Die Funktion soll  $n$  als Argument erhalten und die Tribonacci-Zahl als Rückgabewert.

(b) Warum wird in diesem Falle die einfache rekursive Lösung langsamer sein als eine geeignet programmierte iterative Lösung? Geben Sie bitte eine kurze Begründung in Worten (ein oder zwei Sätze genügen).

(4 Punkte)

### Aufgabe 2: Numerisches Zeichenfeld?

Eine C++-Funktion `is_numeric` soll feststellen, ob ein Feld von Zeichen nur Zahlzeichen, also '0'...'9', enthält. Die Funktion soll als Argument das Feld und die Länge des Feldes  $n$  erhalten. Gültige Feldindizes sind im Bereich  $0, \dots, n-1$ . Der Rückgabewert soll ein Logikwert sein. Hinweis: Die Zeichen '0'...'9' sind hintereinander im Standardzeichensatz angeordnet. Falls Sie dies benötigen: '0' hat Zeichennummer 48, '9' hat Zeichennummer 57.

### Aufgabe 3: Was wird ausgegeben?

(5 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* a) {
    *a += 2;
    cout << "Lima: " << *a << endl;
    return *a+3;
}

int f(int& a) {
    a -= 1;
    cout << "Paris: " << a << endl;
    if (a>=0) return f(&a);
    else return 23;
}

int main() {
    int m7 = 5; // <- setzen Sie hier bitte die letzte
               // Ziffer Ihrer Matrikelnummer ein.
    int m8 = 11;

    int q1 = f(&m7);
    cout << "Rom: " << m7 << endl;
    q1 = f(m8);
    cout << "Tokio: " << q1 << endl;
}
```

---

**Aufgabe 4: Klasse für Polarkoordinaten****(8 Punkte)**

Schreiben Sie eine Klasse `polar2d`, die zur Beschreibung von Punkten in zwei Dimensionen dient. Ein Punkt soll dabei durch Polarkoordinaten  $(r, \alpha)$  dargestellt werden. Der Winkel  $\alpha$  wird dabei in üblicher Weise im Bogenmaß gegen den Uhrzeigersinn ab der positiven  $x$ -Achse gemessen. Die Klasse soll in folgendem Programmteil zum Einsatz kommen:

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

// Klasse polar2d zu ergaenzen

int main()
{
    const double pi=3.141592654 ;

    polar2d p1 (1.0, pi/2) ; // vereinbare einen Punkt durch Radius und Winkel
    p1 *= 2.0 ;             // aendere Radius von p1 um angegebenen Faktor ;
    p1.rotate(-pi/6) ;     // rotiere p1 um das Winkelargument ;
    cout << "Kartesische X-Koordinate: " << get_x (p1) << endl ;
}
```

Schreiben Sie die benötigte Klasse `polar2d` und alle benötigten Konstruktoren, Mitgliedsfunktionen und freien Funktionen, die in dem gegebenen Hauptprogramm aufgerufen werden. Der Radius und die Winkelvariable sollen dabei von außen unzugängliche `double`-Variablen sein. Sie brauchen bei beiden Variablen der Einfachheit halber keine Bereichsüberprüfungen vorzunehmen. Die beide Methoden `*` und `rotate` sollen jeweils den Punkt selbst ändern, beide Methoden haben jeweils ein `double` Argument. Die freie Funktion `get_x` soll die  $x$ -Koordinate des Punktes ermitteln.

---

**Aufgabe 5: Sieb des Eratosthenes****(9 Punkte)**

Es soll ein Programm geschrieben werden, das alle Primzahlen bis zu einer einzugebenden Grenze  $n$  nach folgender Methode (Sieb des Eratosthenes) berechnet:

1. Das Sieb  $S$  soll alle Zahlen  $2, \dots, n$  enthalten.
2. Setze  $p = 2$ .
3. Streiche alle nichttrivialen Vielfachen von  $p$  (d.h.  $2p, 3p, \dots$ ) aus der Menge  $S$ .
4. Wähle als nächstes  $p$  die kleinste Zahl, die größer als  $p$  ist und Element von  $S$ .
5. Wiederhole 3. und 4. bis  $p^2 > n$ .
6. Geben Sie die Zahlen, die im Sieb  $S$  geblieben sind, auf den Bildschirm aus.

Fragen Sie zunächst die obere Grenze für die größte zu berechnende Primzahl,  $n$ , ab. Die Menge  $S$  ist als dynamisch allokiertes Feld von Logikwerten zu realisieren und am Ende des Programms ist der Speicher wieder freizugeben.

Kommentieren Sie die einzelnen Abschnitte in Ihrem Programm, so dass sofort verständlich ist, was gemacht wird. Sie brauchen nicht zu überprüfen, ob die eingegebene Zahl positiv ist.

---