

# ”Grundlagen der Informatik - I - Musterklausur”

Bachelorstudiengänge B\_EP, B\_FM, B\_Ma, B\_SK, B\_ET, B\_IK, B\_MB, M\_IG, M\_IW

30.1.2017

## 1. Matrixoperationen (12 Punkte)

Für eine Rechteckmatrix  $rm_{7,15}$  (7 \* 15 ganzzahlige Matricelemente) soll ein C++-Programm mit folgendem Ablauf geschrieben werden:

- Einlesen der Elemente der 1. Spalte der Matrix (die am weitesten links befindliche Spalte) aus einer Datei, deren Namen vom Nutzer einzugeben ist.
- Bilden der restlichen Elemente nach folgendem Muster:

$$rm_{ij} = \begin{cases} rm_{i,j-1} & \text{für } i = 1 \\ rm_{i,j-1} - rm_{i-1,j} & \text{sonst} \end{cases}$$

- Bestimmen und Ausgabe der Matrixzeile mit der größten Anzahl von negativen Werten (wenn mehrere Zeilen mit gleicher Anzahl ermittelt werden, so soll die Zeile ausgegeben werden, bei der die Maximalzahl zuletzt auftritt).

*Hinweis:* Die hier angegebenen Indizes beziehen sich auf die in der Mathematik üblichen Werte, nicht auf die in C++ gebräuchlichen.

## 2. Rekursive Funktionen (8 Punkte)

*Binomialkoeffizienten* sind mathematische Größen, die in der Kombinatorik eine wichtige Rolle spielen. Sie sind zwar als Produkte (bzw. als Quotienten von Produkten) definiert, lassen sich aber effektiver als rekursive Funktion berechnen. Dabei gilt die folgende Vorschrift:

$$\binom{n}{k} = \begin{cases} \binom{n-1}{k} + \binom{n-1}{k-1} & \text{falls } n > k > 0 \\ 1 & \text{falls } n = k \text{ oder } k = 0 \end{cases}$$

Schreiben Sie eine C++-Funktion zur Berechnung des *Binomialkoeffizienten* und demonstrieren Sie deren Verwendung in einer kleinen Funktion **main()**. Dabei sollen die Werte der Parameter n und k von der Kommandozeile eingelesen werden. Die eingelesenen Werte sollen auf Sinnfälligkeit hin getestet werden.

### 3. Strukturen, Funktionen (12 Punkte)

Gegeben sei eine Struktur **student**

```
struct student {
    char Name[40];
    char Vorname[20];
    int Matrikelnummer;
    int Studiengang;
};
```

Aus dieser Struktur wird ein Feld **studenten** aufgebaut:

```
const int STUDENTENZAHL=10000;
student studenten[STUDENTENZAHL];
```

welches lokal in der Funktion **main** verfügbar ist. Das Feld sei bereits mit Werten belegt. Wenn weniger als 10000 Studenten immatrikuliert sind, so ist das Strukturelement **Matrikelnummer** des ersten freien Feldelementes mit dem Wert **0** belegt.

Entwerfen und implementieren Sie eine C++-Funktion **get\_stg**:

```
int get_stg(student stud[], int len, int stg);
```

in der das Feld **studenten** sequentiell durchmustert wird. Die Zahl der Studenten, die den als Parameter **stg** spezifizierten Studiengang (hier mit einer ganzen Zahl beschrieben) spezifizierten Studiengang studieren, soll von der Funktion zurückgegeben werden.

Weiter soll eine C++-Funktion **print** entworfen und implementiert werden:

```
void print(student stud[], int len, int stg);
```

in der das Feld **studenten** sequentiell durchmustert wird. Dabei sollen alle Studenten (Name, Vorname, Matrikelnummer) auf dem Bildschirm ausgegeben werden, die den als Parameter **stg** spezifizierten Studiengang (hier mit einer ganzen Zahl beschrieben) studieren.

Desweiteren soll eine C++-Funktion **get\_anzahl** entworfen und implementiert werden:

```
int get_anzahl(student stud[], int len);
```

die die Gesamtzahl aller immatrikulierten Studenten zurück gibt.

Demonstrieren Sie die Verwendung der Funktionen **get\_anzahl** und **get\_stg** zur Ermittlung des prozentualen Anteils der Studentenzahl eines vom Nutzer einzugebenden Studienganges an der Gesamtzahl aller Studenten. Das Feld **studenten** und dessen Länge soll jeweils über die Parameter **stud** und **len** den Funktionen übergeben werden.

#### 4. Funktionen (12 Punkte)

Entwickeln Sie eine C++ - Funktion `mult` zur Multiplikation zweier vorzeichenloser ganzer Zahlen unter **ausschließlicher** Verwendung der Operationen Addition, Linksverschiebung, Rechtsverschiebung sowie bitweises UND.

Beispiel:

Operand a = 5 -> Binaerdarstellung (mit 6 Bit) 000101

Operand b = 9 -> Binaerdarstellung (mit 6 Bit) 001001

Multiplikation

```
000101 * 001001
```

```
-----
```

```
          000101
+         0000000 (Addition von 0)
+         00000000 (Addition von 0)
+         000101000
-----
```

```
000101101 -> Dezimaldarstellung 45
```

- Geben Sie den Entwurf der Funktion `mult` mit Hilfe eines Struktogramms an.
- Implementieren Sie die Funktion `mult` in der Programmiersprache C++
- Demonstrieren Sie durch ein kurzes, aber dennoch aussagekräftiges Programmfragment die Verwendungsweise der in b. implementierten Funktion.