

Nachname: \_\_\_\_\_ Vorname: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_ Punkte: \_\_\_\_\_

Hiermit bestätige ich, dass ich die Übungsleistungen als Voraussetzung für diese Klausur in folgender Übung erfüllt habe.

**Jahr:** \_\_\_\_\_ **Übungsleiter:** \_\_\_\_\_ **Unterschrift:** \_\_\_\_\_

### 1. Aufgabe ( / 4 Pkt.)

Gegeben ist folgende Wahrheitstabelle:

a	b	f1(a,b)	f2(a,b)	f3(a,b)	f4(a,b)
0	0	1	1	0	1
0	1	0	1	1	0
1	0	0	0	1	1
1	1	0	1	0	1

Geben Sie bitte die Boole'schen Funktionen an, die die Bedingungen erfüllen, wobei nur die booleschen Operatoren "und", "oder", "nicht" erlaubt sind.

$$f1(a,b) = ((\text{nicht } a) \text{ und } (\text{nicht } b))$$

$$f2(a,b) = (\text{nicht } a) \text{ oder } (a \text{ und } b)$$

$$f3(a,b) = ((\text{nicht } a) \text{ und } b) \text{ oder } (a \text{ und } (\text{nicht } b))$$

$$f4(a,b) = (\text{nicht } a \text{ und nicht } b) \text{ oder } a$$

**jeweils 1 Punkt**

### 2. Aufgabe ( / 6 Pkt.)

a) Erklären Sie den Unterschied zwischen einer **while** und einer **do ... while** Schleife.

**while-Schleife:**

**Nach der Initialisierung wird geprüft, ob die Schleife durchlaufen werden muss (Test auf Fortsetzung der Schleife). Im Schleifenkörper wird das Inkrement der Schleife durchgeführt.**

**2 Punkte**

**do ... while-Schleife:**

**Nach der Initialisierung wird der Schleifenkörper auf jeden Fall einmal durchlaufen. Im Schleifenkörper wird das Inkrement der Schleife durchgeführt, um anschließend zu prüfen, ob die Schleife wiederholt werden muss (Test auf Fortsetzung der Schleife).**

Nachname: \_\_\_\_\_ Vorname: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_ Punkte: \_\_\_\_\_

- b) Erklären Sie den Unterschied zwischen einem **switch** und einer **if ... else** Kette.

**switch ... case: (Fallunterscheidung)**

Es wird der Ausdruck ausgewertet und zum entsprechenden "Fall" gesprungen. Falls es keine Übereinstimmung gibt, wird zum default-Teil gesprungen. Der abzutestende Ausdruck muss vom Typ char oder int sein. Es dürfen keine 2 case Konstanten denselben Wert im selben switch haben. Jeder Fall wird mit der break-Anweisung abgeschlossen.

4 Punkte

**if ... else: (Bedingte Anweisung, Verzweigung)**

Es wird geprüft, ob ein Ausdruck wahr (true) oder falsch (false) ist. Falls er wahr ist, werden die Anweisungen des if-Teils durchgeführt, andernfalls die Anweisungen des else-Teils. Der Ausdruck kann Bereiche ( z.B.  $n \geq 0$  ) abprüfen. Er kann auch aus mehreren Ausdrücken zusammengesetzt sein.

**3. Aufgabe ( / 8 Pkt.)**

- a) Geben Sie die Darstellung des dezimalen Zahlenwertes 90 im Zahlensystem mit der Basis 7 an (inklusive Rechenweg).

$$\begin{array}{r} 90 : 7 = 12 \text{ R } 6 \\ 12 : 7 = 1 \text{ R } 5 \\ 1 : 7 = 0 \text{ R } 1 \end{array}$$

3 Punkte

Daraus folgt:  $90_{10} = 156_7$

$$\text{Probe (nicht gefordert): } 1 * 7^2 + 5 * 7^1 + 6 * 7^0 = 49 + 35 + 6 = 90$$

- b) Erklären Sie den Datentyp bool, der im Praktikum verwendeten C/C++ - Programmierumgebung.

**Der Datentyp bool belegt 1 Byte (= 8 bit).**

2 Punkte

**Er kann die Werte wahr (true, != 0) oder falsch (false, == 0) annehmen**

- c) Erklären Sie den Datentyp int, der im Praktikum verwendeten C/C++ - Programmierumgebung.

**Der Datentyp int belegt 4 Byte (= 32 bit).**

3 Punkte

**In ihm werden ganzzahlige Zahlenwerte gespeichert.**

**Als Zahl mit Vorzeichen (signed) kann der Wertebereich von  $-2^{31}$  (ca. -2 Milliarden ( $-2 \cdot 10^9$ )) bis  $2^{31}-1$  (ca. 2 Milliarden ( $10^9$ )) dargestellt werden. Als vorzeichenlose (unsigned) Zahl kann der Wertebereich von 0 bis  $2^{32}-1$  (ca. 4 Milliarden ( $4 \cdot 10^9$ )) dargestellt werden.**

Nachname: \_\_\_\_\_ Vorname: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_ Punkte: \_\_\_\_\_

#### 4. Aufgabe ( / 25 Pkt.)

Für eine Reihe von  $n$  gegebenen ganzzahligen Messwerten ( $n \leq 255$ ) soll der Mittelwert in einer Funktion *mittelwert* berechnet werden.

- a) Formulieren Sie den Funktionskopf (d.h. die erste Zeile der Definition) dieser Funktion in der im Praktikum verwendeten C/C++ - Programmiersprache.

```
double mittelwert( int daten[], int anz );
```

2 Punkte

- b) Erstellen / programmieren Sie die Funktion in der im Praktikum verwendeten C/C++ - Programmiersprache.

```
double mittelwert( int daten[], int anz )
{
    double rueckgabeWert = 0.0, summe = 0.0;
    int i;

    if ( 0 != anz )
    {
        for ( i = 0; i < anz; i++ )
        {
            summe = summe + daten[i];
        }
        rueckgabeWert = summe / anz;
    }
    else
    {
        rueckgabeWert = 0.0;
    }
    return rueckgabeWert;
}
```

10 Punkte

- c) Erstellen / programmieren Sie in der im Praktikum verwendeten C/C++ - Programmiersprache das zugehörige vollständige Hauptprogramm main(), welches die obige Funktion aufruft.

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

int main()
{
    int i, anz, daten[255];
    double berechneterMittelwert = 0.0;

    // Eingabe
    cout << "Wieviele Messwerte sollen eingegeben werden? ";
    cin >> anz;
```

13 Punkte

Nachname: \_\_\_\_\_ Vorname: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_ Punkte: \_\_\_\_\_

```

for ( i = 0; i < anz; i++ )
{
    cout << "Bitte " << i+1 << ". Messwert eingeben: ";
    cin  >> daten[i];
}

// Berechnung
berechneterMittelwert = mittelwert( daten, anz );

// Ausgabe
cout << "Der Mittelwert der Messreihe ist: ";
cout << berechneterMittelwert << endl;

return 0;
}

```

### 5. Aufgabe ( / 20 Pkt.)

Welche Ausgaben liefert das folgende Programm an Bildschirmausgaben, wenn folgende Werte nacheinander eingegeben werden?

- a) 1. Eingabegruppe: 6 33 27 31 25 30 24 28  
b) 2. Eingabegruppe: 6 27 33 25 31 24 30 28

```

#include <iostream>
using namespace std;

int main()
{
    int vec[10], grenze, n, i;
    int summe = 0;
    char c;

    //Eingabe
    cout << " Anzahl + Elemente + Grenze: " << endl;
    cin >> n;
    for ( i = 0; i < n; i++) {
        cin >> vec[i];
    }
    cin >> grenze;
    cout << " i vec Grenze" << endl;

    //Verarbeitung+Ausgabe
    for ( i = 0; i < n; i++) {
        c = '-';
        if ( vec[i] >= grenze ) {
            c = '*';
            summe++;
        }
        cout << " " << i << " " << vec[i] << " " << c << endl;
    }
    cout << "Ueberschreitungen der Grenze " << grenze;
    cout << " : " << 100 * summe / n << '%' << endl;
}

```

Nachname: \_\_\_\_\_ Vorname: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_ Punkte: \_\_\_\_\_

```
    return 0;  
}
```

a) 1. Eingabegruppe: 6 33 27 31 25 30 24 28

Anzahl + Element + Grenze

i vec Grenze

0 33 \*

1 27 -

2 31 \*

3 25 -

4 30 \*

5 24 -

Ueberschreitubg der Grenze 28 : 50 %

10 Punkte

b) 2. Eingabegruppe: 6 27 33 25 31 24 30 28

Anzahl + Element + Grenze

i vec Grenze

0 27 -

1 33 \*

2 25 -

3 31 \*

4 24 -

5 30 \*

Ueberschreitubg der Grenze 28 : 50 %

10 Punkte

**6. Aufgabe ( / 18 Pkt.)**

Implementieren / programmieren Sie eine rekursive Algorithmus für die Berechnung des Binomialkoeffizienten „n über m“ ( $n, m \in \mathbb{N}_0$ ).

Gefordert sind sowohl das Hauptprogramm, als auch die rekursive Funktion.

Der Binomialkoeffizient ist definiert durch:

$$\binom{n}{m} = \begin{cases} 0 & \text{falls } n < m \\ 1 & \text{falls } n = m \text{ oder } m = 0 \\ \binom{n-1}{m} + \binom{n-1}{m-1} & \text{sonst} \end{cases}$$

**Rekursive Funktion:**

```
int binomial_rekursiv( int n, int m )
{
    int binom;

    if ( n < m )
    {
        binom = 0;
    }
    else
    {
        if ( ( n == m ) || ( 0 == m ) )
        {
            binom = 1;
        }
        else
        {
            binom = binomial_rekursiv( n - 1, m ) +
                    binomial_rekursiv( n - 1, m - 1 );
        }
    }

    return binom;
}
```

9 Punkte

**Hauptprogramm:**

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

int main()
{
    int n, m, binom;

    // Eingabe
    cout << "Fuer welches n ueber m soll der Binomialkoeffizient ";
    cout << "berechnet werden?" << endl;
```

9 Punkte

```

cout << " Bitte Wert fuer n eingeben: ";
cin >> n;
cout << " Bitte Wert fuer m eingeben: ";
cin >> m;

// Berechnung
binom = binomial_rekursiv( n, m );

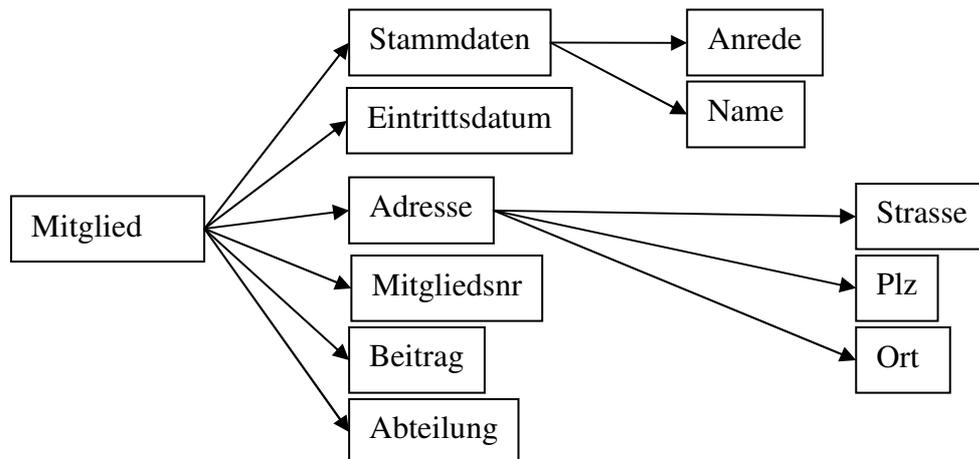
// Ausgabe
cout << "Der Binomialkoeffizient ist " << binom << endl;

return 0;
}

```

### 7. Aufgabe ( / 19 Pkt.)

Für eine Mitgliederverwaltung eines Sportvereins sollen für maximal 500 Mitglieder Informationen gespeichert werden, die alle nach der gleichen Weise wie folgt strukturiert sind:



Dabei gelten folgende Beschreibungen:

Eintrittsdatum ist max. 10 Zeichen lang (hat die Form TT.MM.JJJJ)

Mitgliedsnr ist ein ganze Zahl

Beitrag ist eine reelle Zahl (in Euro)

Stammdaten enthält die zwei Elemente:

Anrede besitzt den Wertevorrat: Frau, Herr, Familie, Eheleute

Name ist max. 35 Zeichen lang

Adresse enthält die folgenden Elemente:

Strasse ist max. 22 Zeichen lang

Plz ist eine 5-stellige Zahl

Ort ist max. 22 Zeichen lang

Abteilung besitzt den Wertevorrat: Fussball, Handball, Rugby, Schwimmen, Turnen

Nachname: \_\_\_\_\_ Vorname: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_ Punkte: \_\_\_\_\_

a) Beschreiben Sie in C/C++ diesen Datentyp vollständig (alle notwendigen Angaben).

```
enum myAnrede { Frau, Herr, Familie, Eheleute };
enum myAbteilung { Fussball, Handball, Rugby,
                  Schwimmen, Turnen };
```

2 Punkte

```
struct myStammdaten
{
    myAnrede  anrede;
    char      name[35]; // oder string name;
};
```

3 Punkte

```
struct myAdresse
{
    char  strasse[22]; // oder string strasse;
    long  plz;
    char  ort[22];    // oder string ort;
};
```

3 Punkte

```
struct mitglied
{
    char      eintrittsDatum[10]; // oder als string
    int       mitgliedsNr;
    double    beitrag;
    myStammdaten stammdaten;
    myAdresse  adresse;
    myAbteilung abteilung;
};
```

6 Punkte

Nachname: \_\_\_\_\_ Vorname: \_\_\_\_\_ Matr.-Nr.: \_\_\_\_\_ Punkte: \_\_\_\_\_

- b) Zeigen Sie in einem Hauptprogramm, wie Ihr Datentyp instantiiert wird und zeigen Sie an untenstehendem Beispiel, wie das neue Mitglied an die 117. Stelle der Liste in die Variablen eingetragen wird:

Mitgliedsnr: 31  
Anrede: Frau  
Name: Lehmann  
Eintrittsdatum: 15.05.1974  
Abteilung: Schwimmen  
Strasse: Rheinstrasse 25  
Plz: 64367  
Ort: Muehltal  
Beitrag: 25,50

<b>5 Punkte</b>
-----------------

```
mitglied verein[500];
```

```
verein[116].mitgliedsNr = 31;
verein[116].stammdaten.anrede = Frau;
strcpy( verein[116]. stammdaten.name, "Lehmann" );
// oder falls string: verein[0].stammdaten.name = "Lehmann";
verein[116].eintrittsDatum = "15.05.1974";
// oder falls string: verein[0].eintrittsDatum = "15.05.1974";
verein[116].abteilung = Schwimmen;
strcpy( verein[116].adresse.strasse, "Rheinstrasse 25" );
// oder falls string: verein[0].adresse.strasse = "Rheinstrasse 25";
verein[116].adresse.plz = 64367;
strcpy( verein[116].adresse.ort, "Muehltal" );
// oder falls string: verein[0].adresse.ort = "Muehltal";
verein[116].beitrag = 25.50;
```