

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 (/ 10 Pkt.)

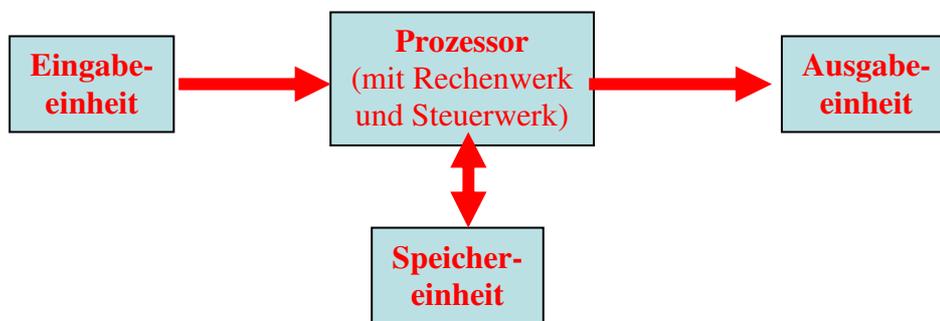
a) Wer oder was sind ALU und FPU? Wofür werden diese benötigt? (3 Punkte)

ALU:
Arithmetic Logic Unit Arithmetisch logische Einheit führt arithmetische und logische Rechenoperation durch

FPU:
Floating Point Unit Fließkommarechner; Co-Prozessor führt Berechnungen mit Gleitkommazahlen durch

Beides sind Bausteine/Bauteile in der CPU (Central Processing Unit) = Prozessors

b) Skizzieren und beschreiben Sie den Aufbau des Basisrechners nach John v. Neumann. Geben Sie Beispiele für die einzelnen Komponenten. (7 Punkte)



Daten werden über die Eingabeeinheit (z.B. Tastatur, Maus) in den Rechner eingegeben.

Diese Daten werden im Prozessor (CPU = Central Processing Unit) verarbeitet (z.B. mathematische Berechnungen, logische Vergleiche). Die Verarbeitung besteht aus Anweisungen, die nacheinander ausgeführt werden, dem Programm.

Die Speichereinheit (Arbeitsspeicher) wird benötigt, damit das Programm die Daten während dieser Verarbeitung ablegen/festhalten kann.

Das Ergebnis der Datenverarbeitung wird über die Ausgabeeinheit (z.B. Monitor, Drucker, Datei) ausgegeben.

Nachname: _____ Vorname: _____ Matr.-Nr.: _____ Punkte: _____

2. Aufgabe (/ 15 Pkt.)

Was liefert das folgende Programm an Bildschirmausgaben?

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

int main()
{
    int a = 1, b = 10, z = 1;
    while ( a + b < 25 )
    {
        cout << z << ". Zeile: " << a + (b * 2) << endl;
        a = a + 2;
        b++;
    }
}
```

(4 Punkte)

1. Zeile: 21
 1. Zeile: 25
 1. Zeile: 29
 1. Zeile: 33
 1. Zeile: 37

```
for ( a = 3; a < 8; a++ )
{
    cout << "Ausgabe " << z << ": ";
    for ( b = 20; b > 0; b = b - 4 )
    {
        cout << a * b << " ";
    }
    cout << endl;
}
cout << endl;
```

(8 Punkte)

Ausgabe 1: 60 48 36 24 12
 Ausgabe 1: 80 64 48 32 16
 Ausgabe 1: 100 80 60 40 20
 Ausgabe 1: 120 96 72 48 24
 Ausgabe 1: 140 112 84 56 28

Nachname: _____ Vorname: _____ Matr.-Nr.: _____ Punkte: _____

3. Aufgabe (/ 20 Pkt.)

- a) Formulieren Sie eine Funktionsdeklaration, die für ein gegebenes Array **V** mit **n** reellen Komponenten **V_i** den Mittelwert **mw** berechnet und zugleich das Minimum **min** ermittelt. Ein- und Ausgabegrößen sollen als Parameter (nicht als Return-Wert) übergeben werden. (4 Punkte)

```
void funktion( double V[], int n, double *mw, double *min );
```

- b) Man gebe die Implementierung dieser Funktion an. Zur Berechnung des Mittelwertes verwende man die bekannte Formel: (8 Punkte)

$$mw = \left(\sum_{i=1}^n v_i \right) / n$$

```
void funktion( double V[], int n, double *mw, double *min )
{
    int i;
    double summe = v[0];

    *min = V[0];
    for (i = 1; i < n; i++ )
    {
        summe = summe + V[i];
        if ( V[i] < *min )
        {
            *min = V[i];
        }
        else
        {
            ; // nichts zu tun
        }
    }
    *mw = summe / n;
}
```

} 3 Punkte

} 4 Punkte

} 1 Punkt

Nachname: _____ Vorname: _____ Matr.-Nr.: _____ Punkte: _____

- c) Implementieren Sie das main-Programm mit einem Funktionsaufruf Ihrer Funktion, so dass mit deren Hilfe für die **50** gegebenen Werte einer aktuellen Messreihe die Größen **mw** und **min** bestimmt werden. Die Messreihe sei in einem Datenarray **v [100]** gespeichert. Die Eingabe der Messwerte ist nicht gefordert!. (8 Punkte)

```
#include "Funktionen.h"
#include <iostream>
using namespace std;

int main()
{
    const int maxAnzahl = 100
    int anzahl = 50;
    double v[maxAnzahl] = { /* 50 Messwerte */ };
    double mw, min;

    // Aufruf der Funktion
    func( v, anzahl, &mw, &min );

    cout << endl;
    cout << "Der Mittelwert = " << mw << endl;
    cout << "Das Minimum    = " << min << endl;

    return 0;
}
```

2 Punkte

2 Punkte

2 Punkte

2 Punkte

4. Aufgabe (/ 14 Pkt.)

- a) Geben Sie die binäre Darstellung des dezimalen Zahlenwertes 53 an (inklusive Rechenweg). Wie wird diese Zahl bei einem Integer-Format mit 2 Byte im Rechner gespeichert? (3 Punkte)

```
53 / 2 = 26 R 1
26 / 2 = 13 R 0
13 / 2 = 6 R 1
6 / 2 = 3 R 0
3 / 2 = 1 R 1
1 / 2 = 0 R 1
```

```
==> 110101          ==> 00000000 00110101
```

Nachname: _____ Vorname: _____ Matr.-Nr.: _____ Punkte: _____

- b) Berechnen Sie binär (inklusive Rechenweg) und schreiben Sie das Ergebnis als Binär- und als Dezimalzahl. Führen Sie anschließend eine Proberechnung durch, um das Ergebnis zu bestätigen.

101011 * 111 (4 Punkte)

```

101011 * 111
-----
 101011
 101011
 101011
-----
100101101

```

Probe: $1 + 4 + 8 + 32 + 256 = 301$

**$101011_2 = 43_{10}$ und $111_2 = 7_{10}$
 $43 * 7 = 301$**

- c) Geben Sie die Darstellung des dezimalen Zahlenwertes 82 im Zahlensystem mit der Basis 6 an (inklusive Rechenweg und Proberechnung) (3 Punkte).

```

82 / 6 = 13 R 4
13 / 6 = 2 R 1
2 / 6 = 0 R 2

```

==> 214_6

==> Probe: $4 * 6^0 + 1 * 6^1 + 2 * 6^2 = 4 + 6 + 72 = 82$

- d) Warum können Buchstaben und sogar Zeichenketten in der Programmiersprache C/C++ auf- bzw. absteigend sortiert werden? (4 Punkte)

Für jedes Zeichen (Buchstaben) gibt es genau einen entsprechenden Zahlenwert.

Diese Zuordnung ist in der so genannten ASCII-Tabelle aufsteigend von a bis z und A bis Z abgelegt.

Eigentlich werden also die entsprechenden Zahlenwerte miteinander verglichen.

Zeichenketten werden zum Vergleich in die einzelnen Zeichen "zerlegt".

5. Aufgabe (/ 4 Pkt.)

Gegeben ist folgende Wahrheitstabelle:

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

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

$$f1(a,b,c) = (!a \text{ und } !b \text{ und } !c) \text{ oder } (!a \text{ und } b \text{ und } c) \text{ oder } (a \text{ und } b \text{ und } !c)$$

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

$$f3(a,b,c) = c$$

je 1 Punkt

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

6. Aufgabe (/ 6 Pkt.)

Beschreiben Sie, wie Variablen deklariert werden müssen, damit sie

a) auf dem Heap, (2 Punkt)

**als dynamische (Zeiger-) Variable;
in C/C++: z.B.: `int *ptr = new int,`**

b) auf dem Stack, (2 Punkte)

**innerhalb {}, also lokal
oder als Übergabeparameter**

c) im Datensegment angelegt werden. (2 Punkte)

global oder static

Nachname: _____ Vorname: _____ Matr.-Nr.: _____ Punkte: _____

7. Aufgabe (/ 11 Pkt.)

Gegeben sind das folgende Haupt- und Unterprogramm:

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

void funktion( int & zahl )
{
    int temp = 2;
    bool ausgabe = false;

    while ( ( temp <= zahl ) && ( false == ausgabe ) )
    {
        if ( 0 == ( zahl % temp ) )
        {
            zahl = zahl / temp;
            funktion( zahl );
            cout << temp << " * ";
            ausgabe = true;
        }
        else
        {
            temp++;
        }
    }
}

int main ()
{
    int zahl = 30;

    cout << "Funktionsaufruf mit " << zahl << " = ";
    funktion( zahl );
    cout << "1" << endl;

    return 0;
}
```

- a) Was wird ausgegeben, wenn das Programm mit einen Wert 30 für die Variable zahl ausgeführt wird? (8 Punkte)

Funktionsaufruf mit 30 = 5 * 3 * 2 * 1

- b) Wie nennt man das Ergebnis? Welche mathematische Funktion ist hier programmiert? (2 Punkt)

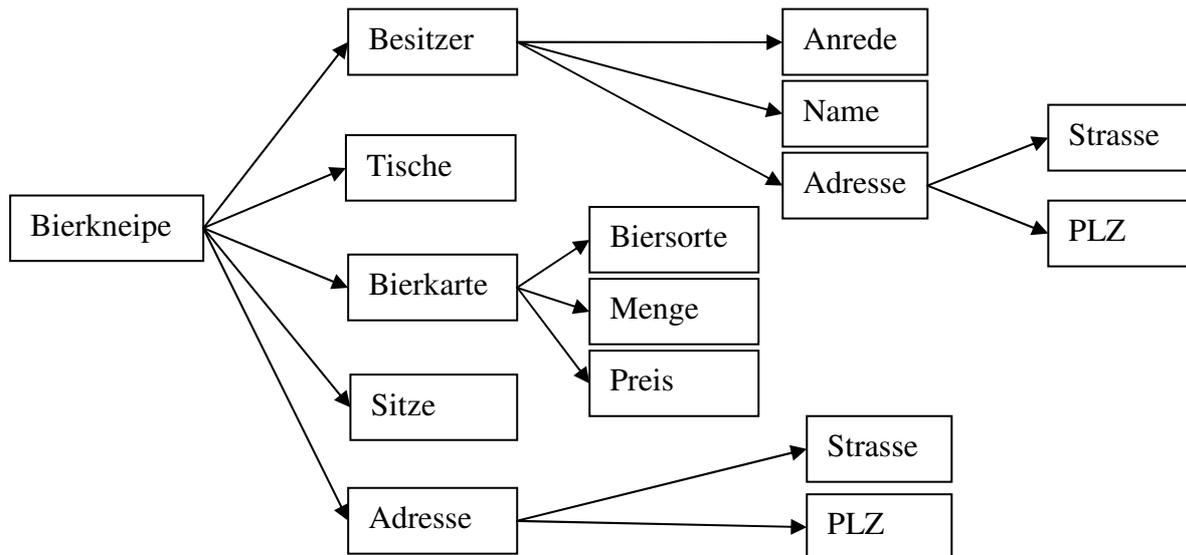
absteigende Primfaktorenzerlegung

- c) Welche besondere Programmieretechnik wurde im Unterprogramm funktion verwendet? (1 Punkt)

Rekursion

8. Aufgabe (/ 22 Pkt.)

Für einen Kneipenführer sollen für maximal 100 Bierkneipen Informationen gespeichert werden, die alle nach der gleichen Weise wie folgt strukturiert sind:



Dabei gelten folgende Beschreibungen:

Tische ist eine ganze Zahl

Sitze ist eine ganze Zahl

Besitzer enthält die folgenden Elemente:

Anrede besitzt den Wertevorrat: Frau, Herr, Eheleute

Name ist max. 35 Zeichen lang

Adresse siehe unter Adresse!

Adresse enthält die folgenden Elemente:

Strasse ist max. 22 Zeichen lang

Plz ist eine 5-stellige Zahl

Bierkarte enthält die folgenden Elemente:

Biersorte besitzt den Wertevorrat: Pils, Export, Koelsch, Weizen, Bock

Menge ist eine reelle Zahl (z.B. 0.3, 0.5, ...)

Preis ist eine reelle Zahl

Nachname: _____ Vorname: _____ Matr.-Nr.: _____ Punkte: _____

a) Beschreiben Sie in C/C++ diesen Datentyp vollständig (alle notwendigen Angaben) (12 Punkte).

```
enum myAnrede    { Frau, Herr, Eheleute };  
enum myBiersorte { Pils, Export, Koelsch, Weizen, Bock};
```

2 Punkte

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

1 Punkt

```
struct myBesitzer {  
    myAnrede  anrede;  
    string    name;           // oder char name[35];  
    myAdresse adresse;  
};
```

3 Punkte

```
struct myBierkarte {  
    myBiersorte biersorte;  
    float       menge;  
    float       preis;  
};
```

3 Punkte

```
struct bierkneipe {  
    int         tische;  
    int         sitze;  
    myBierkarte bierkarte;  
    myAdresse  adresse;  
    myBesitzer besitzer;  
};
```

3 Punkte

Nachname: _____ Vorname: _____ Matr.-Nr.: _____ Punkte: _____

- b) Zeigen Sie in einem Hauptprogramm, wie Ihr Datentyp instantiiert wird und zeigen Sie an untenstehendem Beispiel, wie ein neue Kneipe in die Variablen an die 15. Stelle der Liste eingetragen wird: (11 Punkte)

Anrede: Herr
Name: Meier
Strasse (Besitzer): Rheinstrasse 5
Plz (Besitzer): 64283
Tische: 12
Sitze: 60
Biersorte: Pils
Menge: 0,3
Preis: 1,80
Strasse (Bierkneipe): Kasinostr 23
Plz (Bierkneipe): 64293

```
bierkneipe info[100];  
  
info[14].besitzer.anrede = Herr;  
info[14].besitzer.name = "Meier";  
// oder falls char name[35];  
// strcpy( info[14].besitzer.name, "Meier" );  
strcpy( info[14].besitzer.adresse.strasse, "Rheinstr. 5" );  
// oder falls string strasse ;  
// info[14].besitzer.adresse.strasse = "Rheinstr. 5";  
info[14].besitzer.adresse.plz = 64283;  
info[14].tische = 12;  
info[14].sitze = 60;  
info[14].bierkarte.biersorte = Pils;  
info[14].bierkarte.menge = 0.3;  
info[14].bierkarte.preis = 1.8;  
strcpy( info[14].adresse.strasse, "Kasinostr. 23" );  
// oder falls string strasse ;  
// info[14].adresse.strasse = "Kasinostr. 23";  
info[14].adresse.plz = 64293;
```