

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 (/ 6 Pkt.)

- a) Geben Sie die binäre Darstellung des dezimalen Zahlenwertes 57 an (inklusive Rechenweg und Proberechnung). (3 Punkte)

$$\begin{aligned} 57 / 2 &= 28 \text{ R } 1 \\ 28 / 2 &= 14 \text{ R } 0 \\ 14 / 2 &= 7 \text{ R } 0 \\ 7 / 2 &= 3 \text{ R } 1 \\ 3 / 2 &= 1 \text{ R } 1 \\ 1 / 2 &= 0 \text{ R } 1 \end{aligned}$$

$$111001 = 32 + 16 + 8 + 1 = 57$$

- b) Geben Sie die Darstellung des dezimalen Zahlenwertes 67 im Zahlensystem mit der Basis 5 an (inklusive Rechenweg und Proberechnung) (3 Punkte).

$$\begin{aligned} 67 / 5 &= 13 \text{ R } 2 \\ 13 / 5 &= 2 \text{ R } 3 \\ 2 / 5 &= 0 \text{ R } 2 \end{aligned}$$

$$232 = 50 + 15 + 2 = 67$$

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

2. Aufgabe (/ 15 Pkt.)

Was liefert das folgende Programm an Bildschirmausgaben?

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

int main()
{
    const int obereGrenze = 19;
    const int startWert = 2;
    int x = 7, y = startWert, zeile = 1;
    while ( y + x < obereGrenze )
    {
        cout << zeile << ". Zeile: " << (y + x) * 2 << endl;
        x++;
        y++;
        zeile++;
    }
}
```

(5 Punkte)

```
1. Zeile: 18
2. Zeile: 22
3. Zeile: 26
4. Zeile: 30
5. Zeile: 34
```

```
zeile = 1;
for ( x = 0; x <= obereGrenze; x = x + 4 )
{
    cout << "Ausgabe " << zeile << ": ";
    for ( y = obereGrenze; y > 0; y = y - 4 )
    {
        cout << (x * 2) + y << " ";
    }
    cout << endl;
    zeile++;
}
cout << endl;
```

(10 Punkte)

```
Ausgabe 1: 19 15 11 7 3
Ausgabe 2: 27 23 19 15 11
Ausgabe 3: 35 31 27 23 19
Ausgabe 4: 43 39 35 31 27
Ausgabe 5: 51 47 43 39 35
```

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

3. Aufgabe (/ 4 Pkt.)

Gegeben ist folgende Wahrheitstabelle:

x	y	z	f1(x,y,z)	f2(x,y,z)	f3(x,y,z)	f4(x,y,z)
1	1	1	1	1	0	0
1	1	0	0	1	0	1
1	0	1	0	0	1	1
1	0	0	1	1	1	1
0	1	1	1	1	0	0
0	1	0	0	1	1	1
0	0	1	1	0	0	1
0	0	0	1	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 (x,y,z) =

$$xyz + x!y!z + !xyz + !x!yz + !x!y!z \implies yz + !y!z + !xz$$

f2 (x,y,z) =

$$xyz + xy!z + x!y!z + !xyz + !xy!z + !x!y!z \implies y + !y!z$$

f3 (x,y,z) =

$$x!yz + x!y!z + !xy!z + !x!y!z \implies x!y + !x!z$$

f4 (x,y,z) =

$$xy!z + x!yz + x!y!z + !xy!z + !x!yz \implies y!z + x!y + !yz$$

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

4. Aufgabe (/ 9 Pkt.)

- a) Skizzieren und beschreiben Sie kurz die Wirkungsweise der System-Software-Komponenten (Compiler, Debugger, Editor, Linker), die man benötigt, um von einem Quellprogramm-Entwurf zum lauffähigen Maschinenprogramm zu kommen. Nennen Sie jeweils Beispiele und beachten Sie dabei die richtige Reihenfolge. (6 Punkte)

Editor: Erstellen/Modifizieren des Programmtextes (Header- und Quellcode-Dateien).
Beispiele: Editor in Visual C++ Express, notepad

jeweils 1,5 Punkte

Compiler: Programmtext (Quellcodedatei) übersetzen, dabei Objectcode erzeugen. Nur ein vollständig fehlerfreies Programm kann in Objectcode übersetzt werden.
Beispiele: Compiler in Visual C++ Express, gcc, g++, cc

Linker: Ausführbares Programm erzeugen (aus den zuvor erzeugten Objectcode-Dateien).
Beispiele: Linker in Visual C++ Express, gcc, g++, cc

Debugger: Programm ausführen und testen. Der Debugger ist nicht nur zur Lokalisierung von Programmierfehlern, sondern auch zur Analyse eines Programms durch Nachvollzug des Programmablaufs hilfreich.
Beispiele: Debugger in Visual C++ Express, gdb, dbx

- b) Was bedeutet „rekursive Funktion“?
Nennen Sie Anwendungsbeispiele. (3 Punkte)

Eine rekursive Funktion ist eine Funktion, die sich selbst wieder aufruft.

Beispiele: Fakultätsberechnung
Fibonacci-Zahlen

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

5. Aufgabe (/ 11 Pkt.)

a) Welche Ausgaben liefert das folgende Programm an Bildschirmausgaben?

(11 Punkte)

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

int main()
{
    int info[10] = { 9, 12, 13, 15, 18, 21, 23, 24 ,27, 30 };
    int n = 10, i;
    int erg = 0;
    char c;

    //Verarbeitung + Ausgabe
    for ( i = 0; i < n; i++) {
        c = '-';
        if ( 0 == (info[i] % 3 ) ) {
            c = '*';
            erg = erg + info[i];
        }
        cout << "c = " << c << "    info = " << info[i] << endl;
    }
    cout << "Ergebnis = " << erg << endl;
    return 0;
}
```

```
c = *    info = 9
c = *    info = 12
c = -    info = 13
c = *    info = 15
c = *    info = 18
c = *    info = 21
c = -    info = 23
c = *    info = 24
c = *    info = 27
c = *    info = 30
```

```
Ergebnis = 156
```

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

6. Aufgabe (/ 15 Pkt.)

Geben Sie für die folgende umgangssprachlich beschriebene Anweisung die Umsetzung in ein Teil eines Ablaufdiagramms (Flußdiagramms) oder einen Strukturblock (Struktogramm) (6 Punkte)

sowie in einen Teil eines C/C++-Programms an: (6 Punkte)

Was wird ausgegeben, wenn der Programmausschnitt so ausgeführt wird? (3 Punkte)

Setze a auf 11

Setze b auf 35

Solange (a den Wert kleiner gleich 40) und (b den Wert kleiner gleich 50) hat, wiederhole die folgenden Anweisungen

Falls a gleich b ist

Setze b auf 60

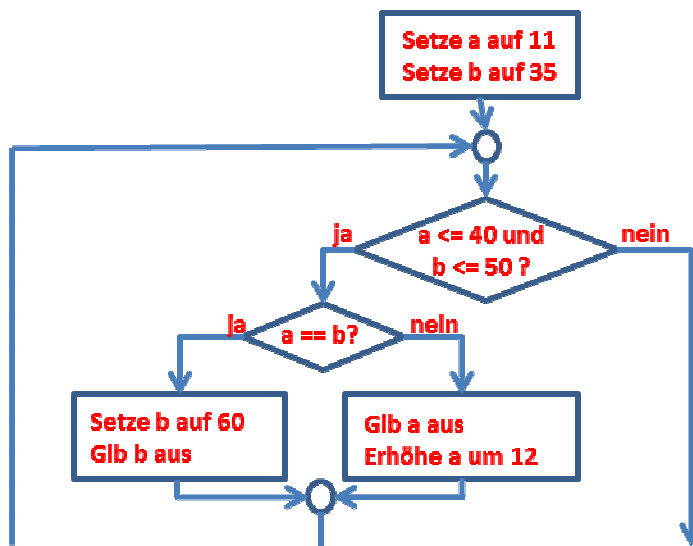
Ausgabe von b

sonst

Ausgabe von a

Erhöhe a um 12

a)



6 Punkte

b)

```

a = 11;
b = 35;
while ( ( a <= 40 ) && ( b <= 50 ) ) {
    if ( a == b ) {
        b = 60;
        cout << "b = " << b << endl;
    }
    else {
        cout << "a = " << a << endl;
        a = a + 12;
    }
}
  
```

6 Punkte

c)

```

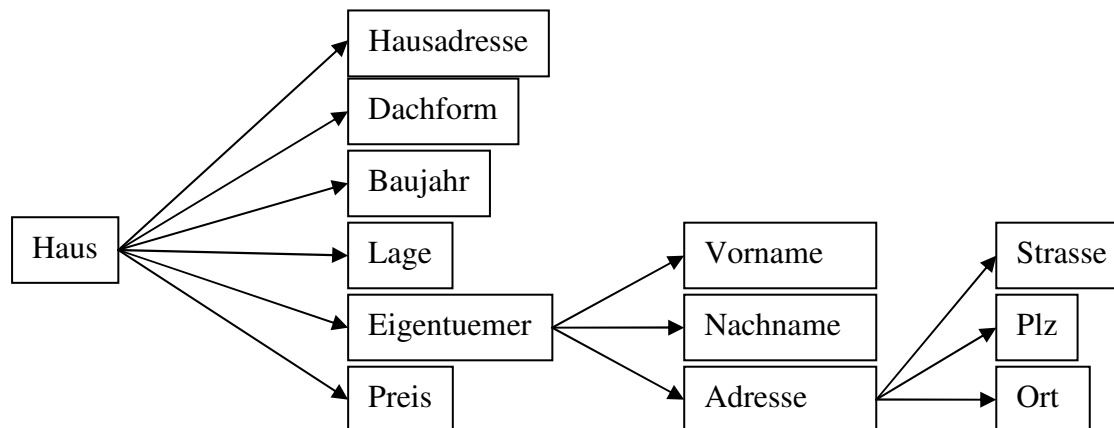
a = 11
a = 23
b = 60
  
```

3 Punkte

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

7. Aufgabe (/ 20 Pkt.)

Für ein Straßenkataster sollen für maximal 1234 Häuser Informationen gespeichert werden, die alle nach der gleichen Weise wie folgt strukturiert sind:



Dabei gelten folgende Beschreibungen:

Dachform besitzt den Wertevorrat: flach, giebel, pult, walm

Baujahr ist eine ganze Zahl

Preis ist eine reelle Zahl

Eigentuerer enthält die Elemente:

Vorname ist max. 25 Zeichen lang

Nachname ist max. 25 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

Hausadresse ist vom gleichen Aufbau wie die Eigentuerer-Adresse

Lage besitzt den Wertevorrat: hang, zentrum, randlage, flussnah

a) Beschreiben Sie in C/C++ diesen Datentyp vollständig (alle notwendigen Angaben) (11 Punkte)

```
enum myDachform { flach, giebel, pult, walm };
enum myLage { hang, zentrum, randlage, flussnah };
```

1 Punkt

```
struct myAdresse
{
    string strasse;
    long plz;
    string ort;
};
```

2 Punkte

```
struct myEigentuerer
{
    string vorname;
    string nachname;
    myAdresse adresse
};
```

2 Punkte

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

```

struct haus
{
    myDachform    dachform;
    int           baujahr;
    double        preis;
    myEigentuemer eigentuemer;
    myAdresse     hausadresse;
    myLage        lage;
};

```

6 Punkte

- b) Zeigen Sie in einem Hauptprogramm, wie Ihr Datentyp instantiiert wird und zeigen Sie an untenstehendem Beispiel, wie ein neues Haus an der **12** Stelle der Liste in die Variablen eingetragen wird: (9 Punkte)

Straße (Hausadresse):	Weserstr. 23
PLZ (Hausadresse):	64331
Ort (Hausadresse):	Weiterstadt
Name (Eigentuemer):	Rot
Vorname (Eigentuemer):	Peter
Straße (Eigentuemer):	Elbestr. 47
PLZ (Eigentuemer):	64367
Ort (Eigentuemer):	Muehltal
Lage:	zentrum
Preis:	21545 EUR
Baujahr:	2002
Dachform:	pult

9 Punkte

```
haus liste[1234];
```

```

liste[11].hausadresse.strasse = "Weserstr. 23";
liste[11].hausadresse.plz = 64331;
liste[11].hausadresse.ort = "Weiterstadt";
liste[11].eigentuemer.nachname = "Rot";
liste[11].eigentuemer.vorname = "Peter";
liste[11].eigentuemer.adresse.strasse = "Elbestr. 47";
liste[11].eigentuemer.adresse.plz = 64367;
liste[11].eigentuemer.adresse.ort = "Muehltal";
liste[11].lage = zentrum;
liste[11].preis = 21545.0;
liste[11].baujahr = 2002;
liste[11].dachform = pult;

```


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

8. Aufgabe (/ 20 Pkt.)

- a) Formulieren Sie eine Funktionsdeklaration **mittel**, die für ein gegebenes Array **V** mit **n** reellen Komponenten **V_i** den Mittelwert **mw**, größten Wert **max** sowie den kleinsten Wert **min** ermittelt. Ein- und Ausgabegrößen sollen als Parameter (nicht als Return-Wert) übergeben werden. (3 Punkte)

```
void mittel( double [], int, double &, double &, double & );

void mittel( double V[], int anz, double &mw, double &min, double
&max );
```

- b) Man gebe die Implementierung dieser Funktion an. Zur Berechnung des Mittelwertes verwende man die bekannte Formel: (11 Punkte)

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

```
void mittel( double V[], int anz, double &mw, double &min, double
&max )
{
    int i;
    double summe = 0;

    for ( i = 0; i < anz; i++ )
    {
        summe = summe + V[i];

        // Minimum und Maximum bestimmen
        if ( min > V[i] )
        {
            min = V[i];
        }
        else
        {
            if ( max < V[i] )
            {
                max = V[i];
            }
            else
            {
                ; // nichts zu tun
            }
        }
    }
    // ende for Schleife
    // Mittelwert berechnen
    mw = summe / anz;
}
```

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**, **max** und **min** bestimmt werden. Die Messreihe sei in einem Datenarray **v [100]** gespeichert. Die Eingabe der Messwerte ist nicht gefordert! (6 Punkte)

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

int main()
{
    const int maxAnz = 100;
    int anzahl = 50;
    double V[maxAnz];
    double mw, min, max;

    // Eingabe der Werte - nicht gefordert

    // Aufruf der Funktion mittel
    mittel ( V, anzahl, mw, min, max );

    // Ausgabe
    cout << "Der Mittelwert ist : " << mw << endl;
    cout << "Der Maximalwert ist: " << max << endl;
    cout << "Der Minimalwert ist: " << min << endl;

    return 0;
}
```