

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 ( / 15 Pkt.)

Was liefert das folgende Programm an Bildschirmausgaben?

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

int main()
{
    const int obereGrenze = 24;
    const int startWert = 2;
    int x = 12, y = startWert, zeile = 1;
    while ( y + x < obereGrenze )
    {
        cout << zeile << ". Zeile: " << (y + x) * 2 << endl;
        x++;
        y++;
        zeile++;
    }
}
```

(5 Punkte)

```
1. Zeile: 28
2. Zeile: 32
3. Zeile: 36
4. Zeile: 40
5. Zeile: 44
```

```
zeile = 1;
for ( x = 4; x <= obereGrenze; x = x + 5 )
{
    cout << "Ausgabe " << zeile << ": ";
    for ( y = obereGrenze; y > 0; y = y - 5 )
    {
        cout << (x + y) << " ";
    }
    cout << endl;
    zeile++;
}
cout << endl;
```

(10 Punkte)

```
Ausgabe 1: 28 23 18 13 8
Ausgabe 2: 33 28 23 18 13
Ausgabe 3: 38 33 28 23 18
Ausgabe 4: 43 38 33 28 23
Ausgabe 5: 48 43 38 33 28
```

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

## 2. 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 23

Setze b auf 35

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

Falls a gleich b ist

Setze b auf 60

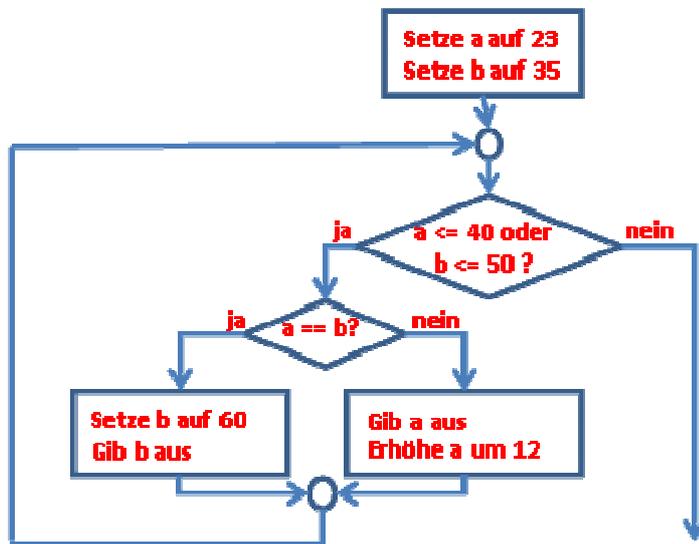
dann Ausgabe von b

sonst

Ausgabe von a

Erhöhe a um 12

a)



6 Punkte

b)

```

a = 23;
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 = 23
b = 60
a = 35
  
```

3 Punkte

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

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

- a) Geben Sie die binäre Darstellung des dezimalen Zahlenwertes 53 an (inklusive Rechenweg und Proberechnung). (3 Punkte)

$$\begin{aligned} 53 / 2 &= 26 \text{ R } 1 \\ 26 / 2 &= 13 \text{ R } 0 \\ 13 / 2 &= 6 \text{ R } 1 \\ 6 / 2 &= 3 \text{ R } 0 \\ 3 / 2 &= 1 \text{ R } 1 \\ 1 / 2 &= 0 \text{ R } 1 \end{aligned}$$

$$110101 = 32 + 16 + 4 + 1 = 53$$

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

$$\begin{aligned} 82 / 6 &= 13 \text{ R } 4 \\ 13 / 6 &= 2 \text{ R } 1 \\ 2 / 6 &= 0 \text{ R } 2 \end{aligned}$$

$$214 = 72 + 6 + 4 = 82$$

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

**4. 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)
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	1	0	0	0	1	1
0	1	1	1	1	1	1
1	0	0	1	1	0	0
1	0	1	0	1	1	1
1	1	0	1	0	0	1
1	1	1	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) =

$$\mathbf{!x!y!z + !xyz + x!y!z + xy!z + xyz \implies yz + x!z + !x!y!z}$$

f2 (x,y,z) =

$$\mathbf{!x!y!z + !x!yz + !xyz + x!y!z + x!yz + xyz \implies !y + yz}$$

f3 (x,y,z) =

$$\mathbf{!xy!z + !xyz + x!yz + xyz \implies !xy + xz}$$

f4 (x,y,z) =

$$\mathbf{!x!yz + !xy!z + !xyz + x!yz + xy!z \implies !xz + y!z + x!yz}$$

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

**5. 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

**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) Wofür wird der „modulo“-Operator benötigt?  
Nennen Sie Anwendungsbeispiele. ( 3 Punkte )

Der module Operator (%) berechnet den Rest einer Ganzzahl-Division.

Beispiele: Zahlenumwandlung  
Schaltjahrberechnung

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

**6. Aufgabe ( / 20 Pkt.)**

- a) Formulieren Sie eine Funktionsdeklaration **funct**, die für ein gegebenes Array **V** mit **n** reellen Komponenten **V<sub>i</sub>** 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 funct( double [], int, double &, double &, double & );

void funct( 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 funct( 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 **44** gegebenen Werte einer aktuellen Messreihe die Größen **mw**, **max** und **min** bestimmt werden. Die Messreihe sei in einem Datenarray **v [123]** gespeichert. Die Eingabe der Messwerte ist nicht gefordert! (6 Punkte)

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

int main()
{
    const int maxAnz = 123;
    int anzahl = 44;
    double V[maxAnz];
    double mw, min, max;

    // Eingabe der Werte - nicht gefordert

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

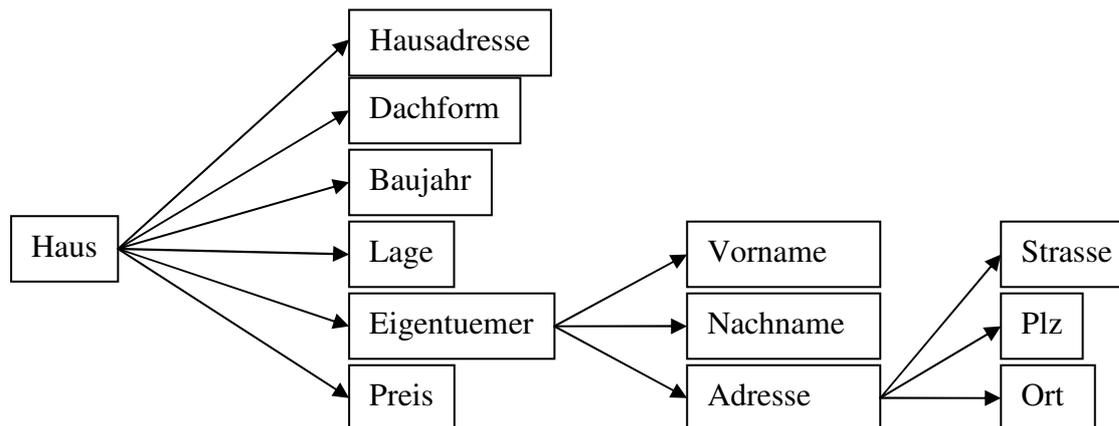
    // Ausgabe
    cout << "Der Mittelwert ist : " << mw << endl;
    cout << "Der Maximalwert ist: " << max << endl;
    cout << "Der Minimalwert ist: " << min << endl;

    return 0;
}
```

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

Eigentuemer 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 Eigentuemer-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 myEigentuemer
{
    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 **21** Stelle der Liste in die Variablen eingetragen wird: (9 Punkte)

Straße (Hausadresse):	Rheinstr. 23
PLZ (Hausadresse):	64283
Ort (Hausadresse):	Darmstadt
Name (Eigentuemer):	Schwarz
Vorname (Eigentuemer):	Nicole
Straße (Eigentuemer):	Mainstr. 47
PLZ (Eigentuemer):	64347
Ort (Eigentuemer):	Griesheim
Lage:	hang
Preis:	12545 EUR
Baujahr:	2005
Dachform:	giebel

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

9 Punkte

```

liste[20].hausadresse.strasse = "Rheinstr. 23";
liste[20].hausadresse.plz = 64283;
liste[20].hausadresse.ort = "Darmstadt";
liste[20].eigentuemer.nachname = "Schwarz";
liste[20].eigentuemer.vorname = "Nicole";
liste[20].eigentuemer.adresse.strasse = "Mainstr. 47";
liste[20].eigentuemer.adresse.plz = 64347;
liste[20].eigentuemer.adresse.ort = "Griesheim";
liste[20].lage = hang;
liste[20].preis = 12545.0;
liste[20].baujahr = 2005;
liste[20].dachform = giebel;

```

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

**8. Aufgabe ( / 11 Pkt.)**

a) Welche Ausgaben liefert das folgende Programm an Bildschirmausgaben?

(11 Punkte)

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

int main()
{
    int info[10] = { 14, 17, 21, 27, 28, 35, 37, 42, 47, 49 };
    int n = 8, i;
    int erg = 0;
    char c;

    //Verarbeitung + Ausgabe
    for ( i = 0; i < n; i++) {
        c = '-';
        if ( 0 == (info[i] % 4 ) ) {
            c = '*';
            erg = erg + info[i];
        }
        cout << "c = " << c << "    info = " << info[i] << endl;
    }
    cout << "Ergebnis = " << erg << endl;
    return 0;
}
```

```
c = -    info = 14
c = -    info = 17
c = -    info = 21
c = -    info = 27
c = *    info = 28
c = -    info = 35
c = -    info = 37
c = -    info = 42
```

```
Ergebnis = 28
```