

Dirk Seeber, h_da, Fb Informatik

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.)

Für eine Reihe von x gegebenen ganzzahligen Messwerten ($x \leq 200$) soll der größte Wert herausgesucht werden. Geben Sie jeweils in C/C++ die dazu notwendigen Typvereinbarungen an sowie die Anweisungsfolge, die das gesuchte Ergebnis ermittelt (ohne Ein- und Ausgabe).

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

int main()
{
    // Variablendeklaration = Typvereinbarung
    int daten[200];
    int i = 0, maximum = 0;

    // Eingabe (nicht gefordert)

    // Anfangselement setzen
    maximum = daten[0];

    // Messreihe verarbeiten
    for ( i = 1; i < 200; i++ )
    {
        if ( maximum < daten[i] )
        {
            maximum = daten[i];
        }
        else
        {
            ; // nichts zu tun
        }
    }

    // Ausgabe (nicht gefordert)

    return 0;
}
```

} **3 Punkte**

} **4 Punkte**

} **1 Punkt**

} **6 Punkte**

} **1 Punkt**

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

2. Aufgabe (/ 8 Pkt.)

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.

Editor: Erstellen/Modifizieren des Programmtextes (Header- und Quellcode-Dateien).

Beispiele: Editor in Visual C++ Express, notepad

jeweils 2 Punkte

Compiler: Programmtext (Quellcodefile) 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

3. Aufgabe (/ 6 Pkt.)

- a) Geben Sie die Darstellung des dezimalen Zahlenwertes 82 im Zahlensystem mit der Basis 6 an (inklusive Rechenweg).

$$82 : 6 = 13 \text{ R } 4$$

$$13 : 6 = 2 \text{ R } 1$$

$$2 : 6 = 0 \text{ R } 2$$

3 Punkte

Daraus folgt: $82_{10} = 214_6$

Probe (nicht gefordert): $2 * 6^2 + 1 * 6^1 + 4 * 6^0 = 72 + 6 + 4 = 82$

- b) Wie sieht eine Zeichendarstellung im character (char) - Format bei der im Praktikum verwendeten C++ - Programmierumgebung aus? Wie viele Zeichen kann man theoretisch darstellen?

Das character - Format belegt 1 Byte (= 8 bit).

3 Punkte

Damit können $2^8 - 1 = 255$ Zeichen dargestellt werden.

4. Aufgabe (/ 10 Pkt.)

- a) Geben Sie für die folgende umgangssprachlich beschriebene Anweisung die Umsetzung in einen Teil eines Ablaufdiagramms oder eines Struktogramms (Nassi-Schneiderman-Diagramm) an:

Setze n auf 100

Wiederhole die Anweisungen

Ausgabe von n ;

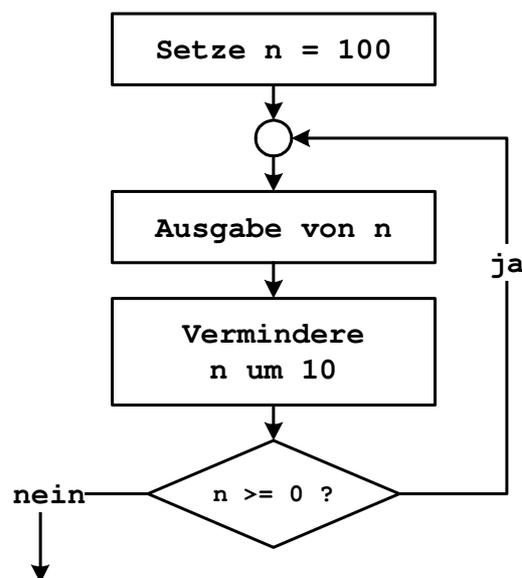
Vermindere n um 10

bis n den Wert 0 oder kleiner 0 hat (d.h. dann soll die Wiederholung abgebrochen werden)

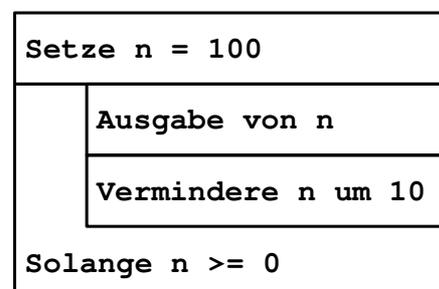
Ablaufdiagramm:

oder

Struktogramm:



5 Punkte



- b) Geben Sie für die oben umgangssprachlich beschriebene Anweisung die Umsetzung in einen Teil eines C-Programms an:

```

n = 100;

do
{
    cout << n << endl;
    n = n - 10;
}
while ( 0 <= n );
  
```

5 Punkte

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

5. Aufgabe (/ 15 Pkt.)

Was liefert das folgende Programm an Bildschirmausgaben?

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

int main()
{
    int x, y;
    for ( x = 1, y = 10; y - x > 0; x++, y-- )
    {
        cout << x << ". Produkt: " << y * x << endl;
    }
}
```

1. Produkt: 10
2. Produkt: 18
3. Produkt: 24
4. Produkt: 28
5. Produkt: 30

5 Punkte

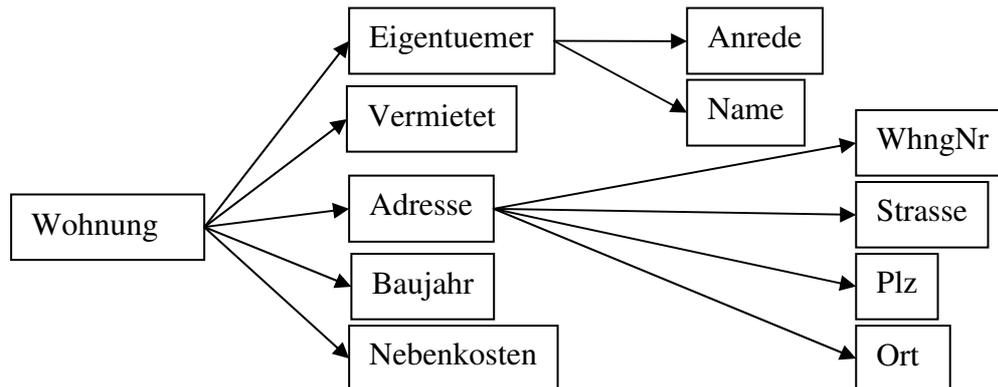
```
for ( y = 1; y <= 5; y++ )
{
    cout << "Zeile " << y << " : ";
    for ( x = 20; x > 0; x = x - 4 )
    {
        cout << x + y << " ";
    }
    cout << endl;
}
cout << endl;
```

```
Zeile 1 : 21 17 13 9 5
Zeile 2 : 22 18 14 10 6
Zeile 3 : 23 19 15 11 7
Zeile 4 : 24 20 16 12 8
Zeile 5 : 25 21 17 13 9
```

10 Punkte

6. Aufgabe (/ 24 Pkt.)

Für eine Hausverwaltung sollen für maximal 100 Wohnungen Informationen gespeichert werden, die alle nach der gleichen Weise wie folgt strukturiert sind:



Dabei gelten folgende Beschreibungen:

Vermietet ist vom booleschen Datentyp (true oder false)

Baujahr ist eine ganze Zahl

Nebenkosten ist eine reelle Zahl

Eigentuemmer 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

WhngNr ist eine ganze Zahl

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

```
enum myAnrede { Frau, Herr, Familie, Eheleute };
```

1 Punkte

```
struct myEigentuemmer
{
    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;
    int whngNr;
};
```

4 Punkte

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

```

struct wohnung
{
    bool        vermietet;
    int         baujahr;
    double      nebenkosten;
    myEigentuemer eigentuemer;
    myAdresse   adresse;
};

```

6 Punkte

- b) Zeigen Sie in einem Hauptprogramm, wie Ihr Datentyp instantiiert wird und zeigen Sie an untenstehendem Beispiel, wie ein neue Wohnung in die Variablen eingetragen wird:

| | |
|--------------|-----------------|
| Anrede: | Herr |
| Name: | Lehmann |
| Vermietet: | Nein |
| WhngNr | 56 |
| Strasse: | Neckarstrasse 5 |
| Plz: | 64293 |
| Ort: | Darmstadt |
| Baujahr: | 1932 |
| Nebenkosten: | 678,12 |

10 Punkte

```

wohnung liste[100];

liste[0].eigentuemer.anrede = Herr;
strcpy( liste[0].eigentuemer.name, "Lehmann" );
// oder falls string: liste[0].eigentuemer.name = "Lehmann";
liste[0].vermietet = false;
liste[0].adresse.whngNr = 56;
strcpy( liste[0].adresse.strasse, "Neckarstrasse 5" );
// oder falls string: liste[0].adresse.strasse = "Neckarstrasse 5";
liste[0].adresse.plz = 64293;
strcpy( liste[0].adresse.ort, "Darmstadt" );
// oder falls string: liste[0].adresse.ort = "Darmstadt";
liste[0].baujahr = 1932;
liste[0].nebenkosten = 678.12;

```

7. 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 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

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

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

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

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

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

jeweils 1 Punkt

8. Aufgabe (/ 18 Pkt.)Gegeben ist das folgende Programm, das die Berechnung von $n!$ ($n \in \mathbb{N}$) liefern soll.

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

int main ()
{
    int n, k, i;

    k = 1;
    for ( i = 2 ; i <= n ; i++)
    {
        k = k * i ;
    }
    return 0;
}
```

a) Welche der grundlegenden Eigenschaften eines Algorithmus ist hier nicht erfüllt und warum?

**Das Verhalten des Algorithmus ist nicht vorhersehbar.
Es gibt kein eindeutiges Ende der for-Schleife, da n nicht initialisiert ist.**

2 Punkte

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

b) Korrigieren Sie den Algorithmus entsprechend.

Mehrere Lösungen möglich:

1. Initialisieren der Variablen: `int n = 0, k = 0, i = 0;`
2. Eingabe von n: `cin >> n;`

2 Punkte

c) Implementieren Sie eine rekursive Alternative für die Berechnung von n! an (es gilt: n ist Element der natürlichen Zahlen: $n \in \mathbb{N}$). Gefordert sind sowohl das Hauptprogramm, als auch die rekursive Funktion.

$$n! = \begin{cases} 1 & \text{falls } n = 1 \\ n * (n - 1)! & \text{sonst} \end{cases}$$

Rekursive Funktion:

```
long fakul_rekursiv ( int x )
{
    long fakul;

    if ( x == 1 )
    {
        fakul = 1;
        cout << "Fakultaet von (" << x << " ) = " << fakul << endl;
    }
    else
    {
        fakul = x * fakul_rekursiv ( x - 1 );
        cout << "Fakultaet von (" << x << " ) = " << fakul << endl;
    }

    return fakul;
}
```

8 Punkte

Hauptprogramm:

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

int main()
{
    int n;
    long fakul;

    cout << "Fuer welche Zahl soll die Fakultaet berechnet werden? ";
    cin >> n;
    fakul = fakul_rekursiv ( n );
    cout << "Fakultaet von " << n << " = " << fakul << endl;

    return 0;
}
```

6 Punkte