

Name, Vorname:

Bemerkungen:

Die Zahlenbasis eines Zahlenwerts (Adresse, Inhalt, Wert) wird in einem Programm wie üblich angegeben (z.B. wird in C die Hexadezimalzahl 12 durch 0x12 angegeben). Außerhalb von Programmen muß eine hexadezimale und eine binäre Zahl durch ein H bzw. ein B angegeben werden. Folgt einer Zahl (außerhalb eines Programms) nicht B oder H, so ist es eine Dezimalzahl. Beispiel: 13H, 00010011B

1) Wie kann man (zumindest mit einer bestimmten Wahrscheinlichkeit) bei einer seriellen Datenübertragung einen Datenübertragungsfehler feststellen ?

2) Sie sollen Daten über die erste serielle Schnittstelle des einen Computers A (25-Pin Stecker) zur zweiten seriellen Schnittstelle des Computers B (25-Pin Stecker) übertragen. Die Schnittstellenparameter sind: 5760 Baud, 8 Datenbits, 2 Stopbits, ungerade Parität.

a) Wie nennt man das dazu benötigte Kabel ?

b) Wie groß ist der Wirkungsgrad der Übertragung ? (genaues Rechenergebnis !)

c) Wie groß ist der Datendurchsatz der Übertragung ? (genaues Rechenergebnis !)

d) Wie lange braucht die Übertragung einer 9600 Byte großen Datei ?

e) Auf welche Werte (und in welcher Reihenfolge) müssen die Inhalte welcher Portadressen gesetzt werden um die obigen Schnittstellenparameter für Computer A einzustellen ? Machen Sie dies in einer zweispaltigen Tabelle.

f) Schreiben Sie einen Programmteil (in C), der die obigen Schnittstellenparameter für den Computer B einstellt.

3) Sie wollen über die serielle Schnittstelle vom Computer A (25 Pin Stecker) über ein selbstgebasteltes Kabel zum Computer B (25 Pin Stecker) die Buchstaben des deutschen Alphabets versenden.

a) Wieviel Leitungen braucht das Kabel minimal ?

b) Welche Pins müssen durch das Kabel verbunden werden ?

Lösungen:

1) Durch das Festlegen der Parität (gerade bzw. ungerade).

2)

a) Nullmodemkabel

$$b) \text{Wirkungsgrad} = \frac{8}{1+8+1+2} = \frac{2}{3}$$

$$c) \text{Datendurchsatz} = \frac{\frac{2}{3} * 5760}{8} = 480 \text{ Byte/s}$$

d) 9600 Byte : 480 Byte / s = 20 Sekunden

e)

Port	Inhalt
0x3FB	0x80
0x3F8	0x14
0x3F9	0x0
0x3FB	0xF

f)

```
r = _outp(0x2fb, 0x80);
```

```
r = _outp(0x2f8, 0x14);
```

```
r = _outp(0x2f9, 0x0);
```

```
r = _outp(0x2fb, 0xF);
```

3)

a) 1 Leitung

b) Pin 2 (Computer A) -----> Pin 3 (Computer B)

Name, Vorname:

Hilfsmittel:
Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programnteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1) Betrachten Sie den folgenden Programmausschnitt:

```
...
int main(){
    int zahl = 4;
    int quad = 8;
    —————→ 1
    quadrat(zahl, &quad);
    —————→ 2
    printf("%d hoch 2 = %d\n", zahl, quad);
    return 0;
}

void quadrat(int z, int *zq){
    *zq = z * z;
}
```

Durch die Deklaration der Variablen zahl und quad werden die folgenden Zellen

	Adresse	Inhalt
zahl	08151	?
quad	04711	?

im Arbeitsspeicher reserviert.

- a) Welchen Wert hat der Inhalt der Variablen zahl und quad an der Stelle 1 im Programm ?
- b) Welchen Wert hat z und zq beim Aufruf von quadrat(zahl, &quad) ?
- c) Was bewirkt die Anweisung `*zq = z * z` an welcher Adresse im Arbeitsspeicher ?
- d) Welchen Wert hat der Inhalt der Variablen zahl und quad an der Stelle 2 im Programm ?

2) Sie wollen ein Programm schreiben, das abhängig von der Eingabe entweder die Summe oder die Differenz oder das Produkt oder den Quotienten zweier Zahlen liefert.

Sie wollen dazu die Funktion tr (wie Taschenrechner) benutzen, die Sie aber wegen Arbeitsüberlastung von einem "Programmierknecht" implementieren (programmieren) lassen. Entwerfen Sie für den "Programmierknecht" eine **Beschreibung** (Leistungsbeschreibung) dieser Funktion mit dem im Unterricht verwendeten Schema. (Kein Programm !!!)

3) Betrachten Sie den folgenden Programmausschnitt:

```
...
int main(){
    float r;
    float u;
    r = 2;
    u = 3;
    → 1
    berechne_umfang (r, u);
    → 2
    printf("Radius= %f, Umfang= %f", r, u);
}

void berechne_umfang(float radius, float umfang){
    umfang = 2 * 3.14 * radius ;
}
```

Durch die Deklaration der Variablen r und u werden die folgenden Zellen

	Adresse	Inhalt
r	0120	?
u	0130	?

im Arbeitsspeicher reserviert.

a) Welchen Wert hat der Inhalt der Variablen r und u an der Stelle 1 im Programm ?

b) Welchen Wert hat radius und umfang beim Aufruf von berechne_umfang (r, u) ?

c) Was bewirkt die folgende Anweisung im Arbeitsspeicher an der Adresse 0130 ?

umfang = 2 * 3.14 * radius;

d) Welchen Wert hat der Inhalt der Variablen r und u an der Stelle 2 im Programm ?

4) Programmieren Sie anhand der folgenden Beschreibung die dazugehörige Funktion:

```
/*
**
**  int ersatz(float R1, float R2, int mod, float *RG)  **
**
**
**
**#*****
/*
```

Parameter:

(i) float R1>0: erster Widerstandswert
(i) float R2>0: zweiter Widerstandswert
(i) int mod:
10: Parallelschaltung
20: Hintereinanderschaltung

(o) float *Rg: Gesamtwiderstand

Return:

(o) 0: Parallelschaltung oder Hintereinanderschaltung wurde
berechnet (mod ist 10 oder 20)
-1: mod ist weder 10 noch 20

Beschreibung:

Berechnet in Abhängigkeit vom Modus mod (10 bedeutet eine
Parallelschaltung, 20 bedeutet eine Hintereinanderschaltung),
den Widerstandswerten R1 und R2 den Ersatzwiderstand
(Gesamtwiderstand) Rg der Widerstandsschaltung.

*/

Lösungen

1)

a) zahl: 4, quad: 8

b) z: 4, zq: 04711

c) In den Inhalt der Adresse 04711 wird der Wert 16 geschrieben.

d) zahl: 4, quad: 16

2)

```
/*  
**  
** double tr (double z1, double z2, int mod)   
**  
*#  
*/
```

Parameter:

```
(i) double z1:           erste Zahl  
(i) double z2!=0, wenn mod=4:  zweite Zahl  
(i) int mod∈{1;2;3;4}:  
    1: berechnet Summe z1+z2  
    2: berechnet Differenz z1-z2  
    3: berechnet Produkt z1*z2  
    4: berechnet Quotient z1/z2
```

Return:

Ergebnis der gewünschten Operation (Summe, oder Differenz oder Produkt oder Quotient).

Beschreibung:

Berechnet in Abhängigkeit vom Modus mod:

Summe $z1+z2$, wenn $mod = 1$

Differenz $z1-z2$, wenn $mod = 2$

Produkt $z1*z2$, wenn $mod = 3$

Quotient $z1/z2$, wenn $mod = 4$

*/

Bemerkung zum Begriff Zusicherung:

Die Angabe beim Parameter z1:

$z2 \neq 0$, wenn $mod=4$

und die Angabe:

$mod \in \{1;2;3;4\}$

nennt man Zusicherung. Das bedeutet, daß unter diesen Voraussetzungen der Programmierer dieser Funktion für die Korrektheit der Berechnungen dieser Funktion **garantiert**.

Je weniger Zusicherungen der Programmierer macht, desto größer wird der programmtechnische Aufwand für ihn, dest mehr "Intelligenz" muss er in die Funktion packen.

3)

a) $r = 2, u = 3$

b) radius = 2, umfang = 3

c) nichts

d) $r = 2, u = 3$

4)

```
int tr(double r1, double r2, int mod, double *rg){
    int r;
    if(mod==10){
        *rg=1/(1/r1+1/r2);
        r=0;
    }
    else if(mod==20){
        *rg=r1+r2;
        r=0;
    }
    else
        r=-1;
    return(r);
}
```

KLAUSUR 1 SAE E2FI Nachtermin Zeit: 60 Minuten
Gruppe A Name, Vorname:

Hilfsmittel:

Bücher und Skripte eigener Wahl.

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

Bemerkung:

- 1) Die Aufgaben müssen nicht am Computer gemacht werden.
- 2) Für jede Funktion muss vor der Implementierung der Funktion eine **Beschreibung** (Leistungsbeschreibung) mit dem im Unterricht verwendeten Schema gemacht werden.

AUFGABEN

- 1)
 - a) Schreiben Sie die Funktion *volumen*, die das Volumen eines Quaders bestimmt.
 - b) Geben Sie ein Beispiel eines möglichen Funktionsaufrufs
- 2)
 - a) Schreiben Sie die Funktion *mittelwert*, die **entweder** den arithmetischen Mittelwert oder den harmonischen Mittelwert zweier Zahlen bestimmt und außerdem zurückliefert, ob diese gleich groß sind. Ein Output davon soll über return realisiert werden.
 - b) Geben Sie ein Beispiel eines möglichen Funktionsaufrufs
- 3)
 - a) Schreiben Sie die Funktion *feldmax*, die von zwei eindimensionalen Feldern (der Länge 2) in einem dritten eindimensionalen Feld (der Länge 2) in der ersten Zelle die Summe der beiden Zellen des ersten Feldes und in der zweiten Zelle des dritten Feldes die Differenz der beiden Zellen des zweiten Feldes abspeichert.
 - b) Geben Sie ein Beispiel eines möglichen Funktionsaufrufs

Beispiel:

1. Feld: [10, 20]
2. Feld: [30, 40]]
3. Feld: [10 + 20, 30 - 40]

Name, Vorname:

Hilfsmittel:

Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) Ein Wagen wird durch seinen Inhabernamen, die Anzahl der Türen und den Hubraum beschrieben.

Realisieren Sie dies durch eine Struktur in der Programmiersprache C.

2) Initialisieren Sie eine Variable mit dem Datentyp Wagen mit von Ihnen ausgesuchten Daten.

3) In einem Verwaltungsprogramm soll über Tastatur die Anzahl der Wagen in einem Fuhrpark eingegeben und unter anderem von diesen Wagen jeweils der Hubraum eingegeben werden können.

Realisieren Sie das C-Programm.

4) Schreiben Sie die Funktion printWagen, die als Parameter eine Variable mit dem Datentyp Wagen bekommt und den Namen, den Hubraum und die Anzahl der Türen des Wagens auf dem Bildschirm ausgibt.

5) Schreiben Sie die Funktion changeWagen, die u.a. als Parameter eine Variable mit dem Datentyp Wagen bekommt und den Hubraum des Wagens abändert.

Zusatzaufgaben (nach der Klausur):

1) Ein Wagen wird durch seinen Inhabernamen, die Anzahl der Türen und den Hubraum beschrieben.

Eine Person wird durch ihre Personalnummer und den Nachnamen beschrieben.

Eine Garage wird durch ihren Besitzer (Person), die in ihr stehenden Wagen und den monatlichen Mietbetrag beschrieben.

Realisieren Sie Person und Wagen jeweils durch eine C-Struktur.

Realisieren Sie auch Garage unter Zuhilfenahme der Strukturen Person und Wagen.

2) Initialisieren Sie eine Variable mit dem Datentyp Garage mit von Ihnen ausgesuchten Daten.

3) In einem Verwaltungsprogramm soll über Tastatur die Anzahl der Garagen in einem Ort und von diesen Garagen jeweils die Garagenbesitzer (speziell: Personalnummer und Nachnamen), die Autos (speziell: Inhabernamen, die Anzahl der Türen und den Hubraum) in der Garage und die Garagenmieten eingegeben werden können.

Benutzen Sie gleich (oder auch erst später) die folgenden Teilaufgaben.

4) Die in dem vorigen Verwaltungsprogramm eingegebenen Daten sollen für jede Garage auf dem Bildschirm ausgegeben werden.

Benutzen Sie dafür gleich (oder auch erst später) die in einigen nachfolgenden Teilaufgaben erstellten Funktionen.

5) Schreiben Sie die Funktion printWagen, die als Parameter eine Variable mit dem Datentyp Wagen bekommt und den Inhabernamen, die Anzahl der Türen und den Hubraum des Wagens auf dem Bildschirm ausgibt.

6) Schreiben Sie die Funktion setWagen, die u.a. als Parameter eine Variable mit dem Datentyp Wagen bekommt und den Inhabernamen, die Anzahl der Türen und den Hubraum des Wagens setzt (abändert).

7) Schreiben Sie die Funktion printPerson, die als Parameter eine Variable mit dem Datentyp Person bekommt und die Personalnummer und den Nachnamen einer Person auf dem Bildschirm ausgibt.

8) Schreiben Sie die Funktion setPerson, die u.a. als Parameter eine Variable mit dem Datentyp Person bekommt und die Personalnummer und den Nachnamen einer Person setzt (abändert).

9) Schreiben Sie die Funktion printGarage, die als Parameter eine Variable mit dem Datentyp Garage bekommt und den Garagenbesitzer (speziell: Personalnummer und Nachnamen), die Autos (speziell: Inhabernamen, die Anzahl der Türen und den Hubraum) in der Garage und die Garagenmiete auf dem Bildschirm ausgibt.

10) Schreiben Sie die Funktion setGarage, die u.a. als Parameter eine Variable mit dem Datentyp Garage bekommt und den Garagenbesitzer (speziell: Personalnummer und Nachnamen), die Autos (speziell: Inhabernamen, die Anzahl der Türen und den Hubraum) in der Garage und die Garagenmiete setzt (abändert).

Lösung:

```
#include "stdafx.h"
#include "stdio.h"
#include <malloc.h>

void printBesitzer(struct garage w);
void changeBesitzer(struct garage *w, int nr);
void printWagen(struct wagen w);
void changeWagen(struct wagen *w, int hub);

struct wagen{
    char besitzer[20];
    double hubraum;
    int anzahl;
};

struct person{
    int pnr;
    char name[20];
};

struct garage{
    struct person inhaber;
    struct wagen fahrzeug[2];
    double miete;
};

void main(){
    int anz,i, hub;
    struct garage *gp;
    struct wagen *wp;
    struct garage
    g1={{100,"Maier"}, {"Bauer",123,4,"Schulze",124,5},567};

    printf("Anzahl der Wagen eingeben\n");
    scanf("%d",&anz);
    wp = (struct wagen*) malloc(anz*sizeof(struct wagen));
    for(i=0;i<anz;i++){
        printf("Hubraum des Wagens eingeben\n");
        scanf("%d",&hub);
        (*(wp+i)).hubraum = hub;
    }

    printf("Anzahl der Garagen eingeben\n");
    scanf("%d",&anz);
    gp = (struct garage*) malloc(anz*sizeof(struct garage));
    for(i=0;i<anz;i++){
        printf("Personalnummer des Garagenbesitzers
        eingeben\n");
        scanf("%d",&hub);
        (*(gp+i)).inhaber.pnr = hub;
    }

    changeBesitzer(gp, 123);
    changeBesitzer(gp+1, 124);
```

```

    changeBesitzer(gp+2, 125);

    printBesitzer(*gp);
    printBesitzer(*(gp+1));
    printBesitzer(*(gp+2));

    printWagen(*wp);

    free(gp);
    free(wp);
}

void printBesitzer(struct garage g){
    printf("Garagenname= %s\n",g.inhaber.name);
    printf("Personalnummer= %d\n",g.inhaber.pnr);
}

void changeBesitzer(struct garage *w, int nr){
    (*w).inhaber.pnr = nr;
}

void printWagen(struct wagen w){
    printf("Wagenart= %s\n",w.besitzer);
    printf("Hubraum= %f\n",w.hubraum);
    printf("Anzahl der Türen= %d\n",w.anzahl);
}

void changeWagen(struct wagen *w, int hub){
    (*w).hubraum = hub;
}

```

Name, Vorname:

Hilfsmittel:

Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) Ein Zimmer wird durch seine Länge, Breite, Höhe und seine Nutzung (Schlafzimmer, Wohnzimmer, usw.) beschrieben.

Eine Person wird durch ihre Personalnummer und den Nachnamen beschrieben.

Eine Wohnung wird durch seine maximal 4 Zimmer, seinen Besitzer und seinen Mieter beschrieben.

Realisieren Sie jeweils Person und Zimmer durch eine Struktur.

Realisieren Sie auch Wohnung unter Zuhilfenahme der Strukturen Person und Zimmer.

2) Initialisieren Sie eine Variable mit dem Datentyp Wohnung mit von Ihnen ausgesuchten Daten.

3) In einem Verwaltungsprogramm soll ein Verwaltungsbeamter über Tastatur die Anzahl der Wohnungen in einem Stadtteil eingeben können und unter anderem von diesen Wohnungen die Personalnummern der Wohnungseigentümer eingeben können.

Realisieren Sie das C-Programm.

4) Schreiben Sie die Funktion printMieter, die als Parameter eine Variable mit dem Datentyp Wohnung bekommt und den Namen und die Personalnummer des Mieters auf dem Bildschirm ausgibt.

5) Schreiben Sie die Funktion changeMieter, die u.a. als Parameter eine Variable mit dem Datentyp Wohnung bekommt und die die Personalnummer eines Mieters einer Wohnung abändert.

Lösung:

```
#include "stdafx.h"
#include "stdio.h"
#include <malloc.h>

void printMieter(struct wohnung w);
void changeMieter(struct wohnung *w, int nr);

struct zimmer{
    double l;
    double b;
    double h;
    char art[20];
};

struct person{
    int pnr;
    char name[20];
};

struct wohnung{
    struct person besitzer;
    struct person mieter;
    struct zimmer raum[4];
};

void main(){
    int anz,i, nr;
    struct wohnung *pw;
    struct wohnung w1 =
    {{100,"Maier"},{200,"Schulz"},{1,2,3,"w",10,20,30,"s"}};
    printf("Anzahl der Wohnungen eingeben\n");
    scanf("%d",&anz);

    pw = (struct wohnung*) malloc(anz*sizeof(struct wohnung));
    for(i=0;i<anz;i++){
        printf("Pnr des Mieters dieser Wohnung eingeben\n");
        scanf("%d",&nr);
        (*(pw+i)).mieter.pnr = nr;
        // (pw+i)->mieter.pnr = nr;
    }

    changeMieter(pw, 123);
    changeMieter(pw+1, 124);
    changeMieter(pw+2, 125);

    printMieter(*pw);
    printMieter(*(pw+1));
    printMieter(*(pw+2));

    free(pw);
}

void printMieter(struct wohnung w){
    printf("Mietername= %s\n",w.mieter.name);
    printf("Personalnummer= %d\n",w.mieter.pnr);
}
```

```
void changeMieter(struct wohnung *w, int nr){  
    (*w).mieter.pnr = nr;  
}
```

Name, Vorname:

Hilfsmittel:

Skripte und Bücher eigener Wahl

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programnteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!
- Bei praktischen Arbeiten am Computer bitte folgendes beachten:
- NUR **ablauffähige** C-Programme werden bewertet ! **Jede** Aufgabe als eigenes Projekt
- Der Name, Vorname und die Aufgabennummer (z.B: Aufgabe 1) muß als **Kommentar** in jedes Programm mitaufgenommen werden.
- Name der Quelldatei nach folgendem Schema vergeben: Gruppe_Rechnernummer_Nachname_Vorname_Aufgabennr.
Beispiel: A_12_Mustermann_Erika_nr3
- Die Programme müssen ausgedruckt werden.
- Die entsprechenden Quelldateien müssen auf Diskette abgespeichert und abgegeben werden.

AUFGABEN

Bemerkung: Jede Teilaufgabe entspricht einem Programmteil. Die ganze Aufgabe soll durch **ein** Projekt realisiert werden.

1) Entwickeln Sie ein Programm, das den Umfang und den Flächeninhalt von Rechtecken berechnet. Das Programmdesign und die Lösung soll **objektorientiert** sein.

Das Programm enthält u.a. eine Eingabemethode scan und eine Ausgabenmethode print.

a) Erzeugen Sie die Klasse Rechteck mit den zugehörigen Attributen und Methoden.

b) Rufen Sie in ... main ... in der folgenden Reihenfolge die Methoden auf.:

rechteck1.print ..., rechteck2.print ..., rechteck1.scan ..., rechteck2.scan ..., rechteck1.print ..., rechteck2.print ...

Begründen Sie, warum beim ersten Aufruf von rechteck1.print ...und rechteck2.print ... Daten auf dem Bildschirm ausgegeben werden, obwohl noch keine Daten eingegeben wurden.

c) Aus Testgründen werden die über Tastatur eingegebenen Rechteckseiten des Objekts rechteck1 verdoppelt und in dem Objekt rechteck2 gespeichert. Geben Sie die Daten von rechteck2 nach der Verdopplung aus. Berechnen Sie, um das Wievielfache sich dann der Umfang erhöht und geben Sie dieses Ergebnis auf dem Bildschirm aus.

d) Erweitern Sie das Programm so, daß jeweils mehrere Rechtecke verwaltet (d.h. einlesen und ausgeben der jeweiligen Daten dieser Rechtecke) werden können. Benutzen Sie dazu ein Feld, in dem hintereinander alle eingelesenen Rechtecke gespeichert werden.

(Bemerkung: Die Feldlänge soll vom Programmierer ohne aufwändige Bemühungen an genau einer Stelle leicht zu verändern sein und muß dann neu compiliert werden. Die Feldlänge soll zuerst einmal den Wert 3 haben).

e) Zu Testgründen werden die über Tastatur im Aufgabenteil d) eingegebenen Rechteckseiten eines jeden Rechtecks verdoppelt und in einem neuen Feld abgespeichert. Geben Sie die Daten nach der Verdopplung aus. Berechnen Sie, um das Wievielfache sich dann der Flächeninhalt eines jeden Rechtecks erhöht und geben Sie diese Ergebnisse auf dem Bildschirm aus.

Name, Vorname:

Hilfsmittel:

Skripte und Bücher eigener Wahl

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!
- Bei praktischen Arbeiten am Computer bitte folgendes beachten:
- NUR **ablauffähige** C-Programme werden bewertet ! **Jede** Aufgabe als eigenes Projekt
- Der Name, Vorname und die Aufgabennummer (z.B: Aufgabe 1) muß als **Kommentar** in jedes Programm mitaufgenommen werden.
- Name der Quelldatei nach folgendem Schema vergeben: Gruppe_Rechnernummer_Nachname_Vorname_Aufgabennr.
Beispiel: A_12_Mustermann_Erika_nr3
- Die Programme müssen ausgedruckt werden.
- Die entsprechenden Quelldateien müssen auf Diskette abgespeichert und abgegeben werden.

AUFGABEN

Bemerkung: Jede Teilaufgabe entspricht einem Programmteil. Die ganze Aufgabe soll durch **ein** Projekt realisiert werden.

1) Entwickeln Sie ein Programm, das den Umfang und den Flächeninhalt von Kreisen berechnet. Das Programmdesign und die Lösung soll **objektorientiert** sein.
Das Programm enthält u.a. eine Eingabemethode scan und eine Ausgabenmethode print.

a) Erzeugen Sie die Klasse Kreis mit den zugehörigen Attributen und Methoden.
Erzeugen Sie in main zwei Kreise k1 und k2.

b) Rufen Sie in ... main ... in der folgenden Reihenfolge die Methoden auf.:
k1.print ..., k2.print ..., k1.scan ..., k2.scan ..., k1.print ..., k2.print ...
Begründen Sie, warum beim ersten Aufruf von k1.print ... und k2.print ... Daten auf dem Bildschirm ausgegeben werden, obwohl noch keine Daten eingegeben wurden.

c) Aus Testgründen wird der über Tastatur eingegebene Radius des Objekts k1 verdoppelt und in dem Objekt k2 gespeichert (dies soll in einem Ausdruck geschehen).
Geben Sie die Daten von k2 nach der Verdopplung aus. Berechnen Sie, um das Wievielfache sich dann der Umfang erhöht und geben Sie dieses Ergebnis auf dem Bildschirm aus.

d) Erweitern Sie das Programm main so, daß jeweils mehrere Kreise (in einer Schleife) verwaltet (d.h. einlesen und ausgeben der jeweiligen Daten dieser Kreise) werden können.
Benutzen Sie dazu ein Feld, in dem hintereinander alle eingelesenen Kreise gespeichert werden.

(Bemerkung: Die Feldlänge soll vom Programmierer ohne aufwändige Bemühungen an genau einer Stelle leicht zu verändern sein und muß dann neu kompiliert werden. Die Feldlänge soll zuerst einmal den Wert 2 haben).

e) Zu Testgründen wird der über Tastatur im Aufgabenteil d) eingegebene Radius eines jeden Kreises verdoppelt und in einem neuen Feld abgespeichert. Geben Sie die Daten nach der Verdopplung aus. Berechnen Sie, um das Wievielfache sich dann der Flächeninhalt eines jeden Kreises erhöht und geben Sie diese Ergebnisse auf dem Bildschirm aus.

KLAUSUR 3 SAE E2FI Nachtermin Zeit: 60 Minuten

Name, Vorname:

Hilfsmittel:

Skripte und Bücher eigener Wahl

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!
Bei praktischen Arbeiten am Computer bitte folgendes beachten:
- NUR **ablauffähige** C-Programme werden bewertet ! **Jede** Aufgabe als eigenes Projekt
- Der Name, Vorname und die Aufgabennummer (z.B: Aufgabe 1) muß als **Kommentar** in jedes Programm mitaufgenommen werden.
- Name der Quelldatei nach folgendem Schema vergeben: Gruppe_Rechnummer_Nachname_Vorname_Aufgabennr.
Beispiel: A_12_Mustermann_Erika_nr3
- Die Programme müssen ausgedruckt werden.
- Die entsprechenden Quelldateien müssen auf Diskette abgespeichert und abgegeben werden.

AUFGABEN

Bemerkung: Jede Teilaufgabe entspricht einem Programmteil. Die ganze Aufgabe soll durch **ein** Projekt realisiert werden.

1) Entwickeln Sie ein Programm, das die Artikel eines Kaufladens verwaltet. Das Programmdesign und die Lösung soll **objektorientiert** sein.

Das Programm enthält u.a. eine Eingabemethode scan und eine Ausgabenmethode print. Ein Artikel enthält der Einfachheit halber nur die Attribute Artikelbezeichnung, Einkaufspreis und Preisfaktor. Der Verkaufspreis ist kein Attribut, sondern berechnet sich als Produkt aus dem Einkaufspreis und dem Preisfaktor.

a) Erzeugen Sie die Klasse Artikel mit den zugehörigen Attributen und Methoden.

b) Angenommen (diese Annahme gilt nur für diesen Aufgabenteil), die Klasse Artikel würde als weiteres Attribut ein Objekt u des Klassentyps Xyz enthalten und der Programmierer würde vergessen im Konstruktor der Klasse Artikel das Attribut u zu initialisieren. Was würde geschehen ?

c) Rufen Sie in ... main ... in der folgenden Reihenfolge die Methoden auf.:
artikel1.scan ..., artikel 2.scan ..., artikel 1.print ..., artikel 2.print ...

d) Von seinen Informanten hat der Eigentümer des Kaufladens mitbekommen, daß ein Konkurrent (der genau die gleichen Artikel vertreibt) die Artikel zwar um das 1,2 - fache einkauft, dafür aber den Preisfaktor auf das 0,8 - fache reduziert..
Deshalb werden aus Testgründen der über Tastatur eingegebene Einkaufspreis und der Preisfaktor des Objekts artikel1 wie gerade beschrieben abgeändert und in dem Objekt artikel2 gespeichert. Geben Sie die Daten von artikel2 nach der Veränderung aus. Berechnen Sie, um das Wievielfache sich dann der Verkaufspreis erhöht und geben Sie dieses Ergebnis auf dem Bildschirm aus.

e) Erweitern Sie das Programm so, daß jeweils mehrere Artikel verwaltet (d.h. einlesen und ausgeben der jeweiligen Daten dieser Artikel) werden können. Benutzen Sie dazu ein Feld, in dem hintereinander alle eingelesenen Artikel gespeichert werden.

(Bemerkung: Die Feldlänge soll vom Programmierer ohne aufwändige Bemühungen an genau einer Stelle leicht zu verändern sein und muß dann neu compiliert werden. Die Feldlänge soll zuerst einmal den Wert 3 haben).

f) Zu Testgründen werden die über Tastatur im Aufgabenteil e) eingegebenen Artikel so wie im Aufgabenteil d) verändert und in einem neuen Feld abgespeichert. Geben Sie die Daten nach der Veränderung aus. Berechnen Sie, um das Wievielfache sich dann der Verkaufspreis eines jeden Artikels erhöht und geben Sie diese Ergebnisse auf dem Bildschirm aus.

Name, Vorname:

Hilfsmittel:

Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

- 1) Was ist ein Standard-Konstruktor ?
- 2) Wann (unter welchen Umständen) erzeugt der Compiler den Standard-Konstruktor ?
- 3) Wann wird ein Konstruktor aufgerufen ?
- 4) Eine Klasse A enthalte u.a. das (eingelagerte) Objekt b (der Klasse B).
Was passiert, wenn der Programmierer im Rumpf des Konstruktors A das Objekt b nicht initialisiert ?
- 5) Was ist ein Destruktor ?
- 6) Wie viele Destrukturen gibt es pro Klasse ?
- 7) Wann wird ein Destruktor aufgerufen ?
- 8) Die Klasse Kuh wird wie folgt implementiert:

```
class Kuh{
    private:
        double gewicht;
        double milchleistung;
    public:
        ...
};
```

 - a) Erzeugen (implementieren) Sie einen Konstruktor für die Klasse Kuh.
 - b) Legen Sie ein Feld mit 4 Elementen an. Jedes Element muß ein Objekt der Klasse Kuh sein.
 - c) In main wird das Objekt k der Klasse Kuh angelegt. Kann man durch die folgende Anweisung in main auf das Attribut gewicht im Objekt k zugreifen ? Begründen Sie!
`k.gewicht = 13;`

- 9) a) Warum gibt es im folgenden Programm eine Fehlermeldung des Kompilers ?
b) Welche 2 Möglichkeiten gibt es, diese zu beseitigen ?

```
#include "stdafx.h"
#include <stdio.h>

class Motor{
private:
    double hubraum;
    double leistung;
public:
    Motor(double h, double l){
        setMotor(h,l);
    }

    void setMotor(double h, double l){
        hubraum=h;
        leistung=l;
    }
};

class Auto{
private:
    Motor m;
    double gewicht;

public:
    Auto(double h, double l, double g){
        m.setMotor(h, l);
        gewicht = g;
    }
};

void main(){
    Auto a1(300,40,500);
}
```

Lösungen:

- 1) Ein parameterloser Konstruktor
- 2) Wenn der Programmierer keinen Konstruktor erzeugt hat.
- 3) Beim Instanzieren (anlegen) eines Objekts.
- 4) Dann wird der Standard-Konstruktor von B aufgerufen.
- 5) Ein Destruktor ist in gewisser Weise das Gegenteil eines Konstruktors, weil evtl. vorher im Konstruktor angeforderte Ressourcen (z.B. allozierter Speicherplatz, geöffnete Dateien, usw.) im Destruktor wieder freigegeben werden müssen.
- 6) Einen Destruktor (ohne Parameter).
- 7) Am Ende einer Funktion (Methode) und am Ende eines Blocks.

8) Die Klasse Kuh wird wie folgt implementiert:

```
class Kuh{
    private:
        double gewicht;
        double milchleistung;
    public:
        ...
};
```

a)

```
Kuh::Kuh(double g, double m){
    gewicht = g;
    milchleistung = m;
}
```

b)

```
Kuh kuhstall[4] = {Kuh(9,8), Kuh(3,4), Kuh(2,7), Kuh(5,6)};
```

c)

Nein, weil gewicht private ist. Auf private Variablen darf von aussen nicht zugegriffen werden.

9) a) In der Klasse Auto befindet sich das eingelagerte Objekt m (der Klasse Motor).

Da im Konstruktor Auto nicht -für das Objekt m - an den Konstruktor Motor weitergereicht wird, wird der Standard-Konstruktor von Motor aufgerufen. Da dieser nicht existiert, gibt es eine Fehlermeldung.

b)

b1) Einen Standard-Konstruktor in Motor implementieren.

b2) Im Konstruktor Auto an den Konstruktor Motor für m weiterleiten:

```
Auto(double h, double l, double g):m(h,l){
    ...
}
```

Name, Vorname:

Hilfsmittel:

Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programnteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1) In einer schon geöffneten Datei befinden sich lauter Vornamen, die durch ein Leerzeichen getrennt sind. Diese sollen alle ausgelesen und auf dem Bildschirm ausgegeben werden. Schreiben Sie für das Auslesen und Ausgeben (auf dem Bildschirm) den entsprechenden Programmausschnitt.

2) a) Was ist eine Klassenvariable und welchen Sinn hat es, diese in einem Programm zu benutzen ?

b) In einer Klassenarbeit gibt es eine maximale Punktezahl (maxPunkte, Datentyp integer), von der jeder Schüler eine bestimmte Punktezahl (punkte, Datentyp integer) erreicht. Jeder Schüler hat einen Namen (nachname, Datentyp string). Die Klasse hat eine bestimmte Anzahl Schüler (schuelerAnz, Datentyp integer).

Es sollen 2 Objekte der Klasse "Klassenarbeit" erzeugt werden. Benutzen Sie dazu Klassenvariable(n).

Schreiben Sie dazu den entsprechenden Programmausschnitt.

Benutzen Sie dazu u.a. folgende Methoden (Datentyp und formale Parameter bitte selbst bestimmen):

```
...setMaxPunkte...  
...getMaxPunkte...  
...setPunkte...  
...getPunkte...  
...setNachname...  
...getNachname...  
...setSchuelerAnz...  
...getSchuelerAnz...
```

c) Zusatzaufgabe (gibt zusätzliche Punkte):

Die Gesamtpunktzahl aller erreichten Punkte aller Schüler soll auch berechnet werden.

Wie kann man dies am elegantesten lösen

3) Es soll ein Verwaltungsprogramm für Klassenarbeiten entwickelt werden, das sämtliche erreichte Punkte der verschiedenen Schüler einer Klasse speichert.

Das Programm soll für verschiedene Schülerklasse mit verschiedenen Anzahl Schülern funktionieren.

a) Welchen Nachteil hat der Datentyp Feld (array) zur Lösung des Problems ?

b) Schreiben Sie einen Programmausschnitt, der nur - die vom Lehrer jeweils eingegebenen - erreichten Punkte eines Schülers dynamisch speichert.

c) Schreiben Sie einen Programmausschnitt, der die jeweils vom Lehrer eingegebenen Objekte (Nachnamen, erreichten Punkte) eines Schülers in einem dynamischen Feld vom Datentyp Klassenarbeit speichert.

Bemerkung:

Die Klasse "Klassenarbeit" und deren Methoden können von der vorigen Aufgabe (ohne sie nochmals neu zu implementieren bzw. abzuschreiben, übernommen, bzw. als bekannt vorausgesetzt werden. Derjenige, der die vorige Aufgabe nicht bearbeitet hat, kann trotzdem die Klasse bzw. die Methoden darin verwenden.

Name, Vorname:

Hilfsmittel:

2 Seiten selbstverfasstes Manuskript

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

Es soll ein Verwaltungsprogramm für einen Milchbauern entwickelt werden, in dem die Milchleistung (in Liter pro Tag), der aktuelle Preis pro Liter Kuhmilch und das Gewicht einer Kuh gespeichert werden.

Dazu muß eine Klasse Kuh erzeugt werden.

a) Was ist die Klassenvariable ?

b) Erzeugen Sie die Klasse Kuh mit den entsprechenden Attributen und den zugehörigen Methoden.

c) Schreiben Sie einen Programmausschnitt, der die jeweils vom Bauern eingegebenen Daten einer Kuh in einem dynamischen Feld vom Datentyp Kuh speichert und danach auf dem Bildschirm ausgibt.

Name, Vorname:

Hilfsmittel:
Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programnteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1) Betrachten Sie den folgenden Programmausschnitt:

```
...
int main(){
    int zahl = 4;
    int quad = 8;
    —————> 1
    quadrat(zahl, &quad);
    —————> 2
    printf("%d hoch 2 = %d\n", zahl, quad);
    return 0;
}

void quadrat(int z, int *zq){
    *zq = z * z;
}
```

Durch die Deklaration der Variablen zahl und quad werden die folgenden Zellen

	Adresse	Inhalt
zahl	08151	?
quad	04711	?

im Arbeitsspeicher reserviert.

- a) Welchen Wert hat der Inhalt der Variablen zahl und quad an der Stelle 1 im Programm ?
- b) Welchen Wert hat z und zq beim Aufruf von quadrat(zahl, &quad) ?
- c) Was bewirkt die Anweisung *zq = z * z an welcher Adresse im Arbeitsspeicher ?
- d) Welchen Wert hat der Inhalt der Variablen zahl und quad an der Stelle 2 im Programm ?

2) Sie wollen ein Programm schreiben, das abhängig von der Eingabe entweder die Summe oder die Differenz oder das Produkt oder den Quotienten zweier Zahlen liefert.

Sie wollen dazu die Funktion tr (wie Taschenrechner) benutzen, die Sie aber wegen Arbeitsüberlastung von einem "Programmierknecht" implementieren (programmieren) lassen. Entwerfen Sie für den "Programmierknecht" eine **Beschreibung** (Leistungsbeschreibung) dieser Funktion mit dem im Unterricht verwendeten Schema. (Kein Programm !!!)

3) Betrachten Sie den folgenden Programmausschnitt:

```
...
int main(){
    float r;
    float u;
    r = 2;
    u = 3;
    → 1
    berechne_umfang (r, u);
    → 2
    printf("Radius= %f, Umfang= %f", r, u);
}

void berechne_umfang(float radius, float umfang){
    umfang = 2 * 3.14 * radius ;
}
```

Durch die Deklaration der Variablen r und u werden die folgenden Zellen

	Adresse	Inhalt
r	0120	?
u	0130	?

im Arbeitsspeicher reserviert.

a) Welchen Wert hat der Inhalt der Variablen r und u an der Stelle 1 im Programm ?

b) Welchen Wert hat radius und umfang beim Aufruf von berechne_umfang (r, u) ?

c) Was bewirkt die folgende Anweisung im Arbeitsspeicher an der Adresse 0130 ?

umfang = 2 * 3.14 * radius;

d) Welchen Wert hat der Inhalt der Variablen r und u an der Stelle 2 im Programm ?

4) Programmieren Sie anhand der folgenden Beschreibung die dazugehörige Funktion:

```
/*
**
**  int ersatz(float R1, float R2, int mod, float *RG)
**
**
**#
*/
```

Parameter:

- (i) float R1>0: erster Widerstandswert
- (i) float R2>0: zweiter Widerstandswert
- (i) int mod:
 - 10: Parallelschaltung
 - 20: Hintereinanderschaltung

(o) float *Rg: Gesamtwiderstand

Return:

- (o) 0: Parallelschaltung oder Hintereinanderschaltung wurde berechnet (mod ist 10 oder 20)
- 1: mod ist weder 10 noch 20

Beschreibung:

Berechnet in Abhängigkeit vom Modus mod (10 bedeutet eine Parallelschaltung, 20 bedeutet eine Hintereinanderschaltung), den Widerstandswerten R1 und R2 den Ersatzwiderstand (Gesamtwiderstand) Rg der Widerstandsschaltung.

*/

Lösungen

1)

a) zahl: 4, quad: 8

(2P)

b) z: 4, zq: 04711

(1P + 4P)

c) In den Inhalt der Adresse 04711 wird der Wert 16 geschrieben.

(1 P)

d) zahl: 4, quad: 16

(4 P)

2)

(13 P)

```
/* **** */
/**                                           **/
/** double tr (double z1, double z2, int mod) **/
/**                                           **/
/*# **** */
/*
```

Parameter:

(i) double z1: erste Zahl

(i) double z2!=0, wenn mod=4: zweite Zahl

(i) int mode{1;2;3;4}:

1: berechnet Summe z1+z2

2: berechnet Differenz z1-z2

3: berechnet Produkt z1*z2

4: berechnet Quotient z1/z2

Return:

Ergebnis der gewünschten Operation (Summe, oder Differenz oder Produkt oder Quotient).

Beschreibung:

Berechnet in Abhängigkeit vom Modus mod:

Summe z1+z2, wenn mod = 1

Differenz z1-z2, wenn mod = 2

Produkt z1*z2, wenn mod = 3

Quotient z1/z2, wenn mod = 4

*/

Bemerkung zum Begriff Zusicherung:

Die Angabe beim Parameter z1:

z2!=0, wenn mod=4

und die Angabe:

mode{1;2;3;4}

nennt man Zusicherung. Das bedeutet, daß unter diesen Voraussetzungen der Programmierer dieser Funktion für die Korrektheit der Berechnungen dieser Funktion **garantiert**.

Je weniger Zusicherungen der Programmierer macht, desto größer wird der programmtechnische Aufwand für ihn, dest mehr "Intelligenz" muss er in die Funktion packen.

3)

a) $r = 2, u = 3$

(2P)

b) radius = 2, umfang = 3

(2P)

c) nichts

(4P)

d) $r = 2, u = 3$

(1P + 4P)

4)

(13P)

```
int tr(double r1, double r2, int mod, double *rg){
    int r;
    if(mod==10){
        *rg=1/(1/r1+1/r2);
        r=0;
    }
    else if(mod==20){
        *rg=r1+r2;
        r=0;
    }
    else
        r=-1;
    return(r);
}
```

Name, Vorname:

Hilfsmittel:

Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1) Es soll ein kleines Demoprogramm zum Testen der Klasse Rechteck und deren Methoden entwickelt werden.

Dazu muß eine Klasse "Rechteck" erzeugt werden.

a) Erzeugen Sie die Klasse "Rechteck".

Diese Klasse hat genau die folgenden Attribute (Datentyp bitte selbst bestimmen) und sonst keine anderen:

seiteA, seiteB

Diese Klasse hat genau die folgenden Methoden (Datentyp und formale Parameter bitte selbst bestimmen) und sonst keine anderen:

...setSeiteA...

...setSeiteB...

...getSeiteA...

...getSeiteB...

...berechneFlaeche... (berechnet die Fläche des Rechtecks)

...vergroessere... (vergroessert alle zwei Seiten um einen bestimmten Faktor)

...printAll... (gibt die zwei Seitenlängen und die Fläche des Rechtecks auf dem Bildschirm aus)

Bemerkung:

Die Methoden müssen Zusicherungen machen (z.B. dürfen die Seitenlängen des Rechtecks nicht negativ werden)

b) Schreiben Sie einen Programmausschnitt (Demoprogramm), in dem ein Rechteck angelegt wird (der Variablenname des Rechtecks sei "tisch").

b1) Dann sollen zwei Zahlen über Tastatur eingegeben werden.

b2) Mit den entsprechenden Methoden sollen diese Zahlen die zwei Seitenlängen des Rechtecks festlegen.

b3) Mit der entsprechenden Methode sollen die zwei Seiten des Rechtecks verdoppelt werden.

b4) Mit der entsprechenden Methode sollen die zwei Seiten und die Fläche auf dem Bildschirm ausgegeben werden.

b5) Mit der entsprechenden Methode soll der Umfang des Rechtecks berechnet, in einer Variable abgespeichert und dann auf dem Bildschirm ausgegeben werden.

Lösungen:

```
#include "stdafx.h"
#include "stdio.h"

class Rechteck{
    private:
        double seiteA;
        double seiteB;

    public:
        void setSeiteA(double s);
        void setSeiteB(double s);
        double getSeiteA();
        double getSeiteB();
        // void scanSeiteA();
        // void scanSeiteB();
        double Rechteck::berechneFlaeche();
        // double Rechteck::berechneUmfang();
        void Rechteck::vergroessereSeiten(double faktor);
        void Rechteck::printAll();
}; // class Rechteck

void Rechteck::setSeiteA(double s){
    if(s>0){ // !!! Zusicherung machen !!!
        seiteA=s;
    }
}

void Rechteck::setSeiteB(double s){
    if(s>0){ // !!! Zusicherung machen !!!
        seiteB=s;
    }
}

double Rechteck::getSeiteA(){
    return(seiteA);
}

double Rechteck::getSeiteB(){
    return(seiteB);
}

// void Rechteck::scanSeiteA(){
//     double temp;
//     printf("Bitte Seite A eingeben\n");
//     scanf("%lf",&temp);
//     if(temp>0){ // !!! Zusicherung machen !!!
//         seiteA=temp;
//     }
// }

// void Rechteck::scanSeiteB(){
//     double temp;
//     printf("Bitte Seite B eingeben\n");
//     scanf("%lf",&temp);
//     if(temp>0){ // !!! Zusicherung machen !!!
//         seiteB=temp;
//     }
// }
```

<pre>double Rechteck::berechneFlaeche() { return(seiteA*seiteB); } // double Rechteck::berechneUmfang() { // return(2*getSeiteA()+2*getSeiteB()); // } void Rechteck::vergroessereSeiten(double faktor) { if(faktor>0){ // !!! Zusicherung machen !!! seiteA=seiteA*faktor; seiteB=seiteB*faktor; } } void Rechteck::printAll() { printf("Seite A=%f\n",seiteA); printf("Seite B=%f\n",seiteB); printf("Flaeche=%f\n",berechneFlaeche()); } int main(){ Rechteck tisch; double seite1, seite2, umfang; printf("1. Seitenlaenge eingeben\n"); scanf("%lf",&seite1); printf("2. Seitenlaenge eingeben\n"); scanf("%lf",&seite2); // alternativ möglich: // tisch.scanSeiteA(); // tisch.scanSeiteB(); tisch.setSeiteA(seite1); tisch.setSeiteB(seite2); tisch.vergroessereSeiten(2); tisch.printAll(); umfang=2*(tisch.getSeiteA()+tisch.getSeiteB()); printf("Umfang=%f\n",umfang); return 0; }</pre>	<p>4P</p> <p>6P</p> <p>6P</p> <p>1P</p> <p>1P</p> <p>1P</p> <p>1P</p> <p>2P</p> <p>2P</p> <p>2P</p> <p>2P</p> <p>4P</p> <p>1P</p>
---	---

Bemerkungen:

1) Statt die Seiten des Rechtecks im Hauptprogramm einzugeben, ist es besser eigene Methoden in der Klasse Rechteck zu basteln (siehe oben):

```
void Rechteck::scanSeiteA();
void Rechteck::scanSeiteB();
```

2) Statt den Umfang umständlich mit:

```
umfang=2*(tisch.getSeiteA()+tisch.getSeiteB());
```

im Hauptprogramm auszurechnen, ist es besser eigens eine Methode in der Klasse Rechteck zu basteln (siehe oben):

```
double Rechteck::berechneUmfang() {
```

Name, Vorname:

Hilfsmittel:

Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1)

Ein Zylinder wird durch seinen Kreis (Grundkreis) und seine Höhe beschrieben.

Ein Kreis wird durch seinen Radius, die x-Koordinate und die y-Koordinate seines Mittelpunkts beschrieben.

Es soll ein kleines Demoprogramm zum Testen der Klasse Zylinder und deren Methoden entwickelt werden. Dazu muß eine Klasse "Zylinder" erzeugt werden.

Die Deklarationen der Klassen "Zylinder" und "Kreis" sind wie folgt gegeben:

(es dürfen keine weitere Datenmember und Methodenmember zu den Klassen hinzugefügt werden).

```
class Kreis{
    private:
        double x;
        double y;
        double r;

    public:
        Kreis(double xx, double yy, double rr);
        void setX(double xx);
        void setY(double yy);
        double getX();
        double getY();
        void setRadius(double rr);
        double getRadius();
        double berechneFlaeche();
}; // class Kreis
```



```

class Zylinder{
    private:
        Kreis k;
        double h;

    public:
        Zylinder::Zylinder(double xx, double yy, double rr,
                           double hh);

        void setX(double xx);
        void setY(double yy);
        double getX();
        double getY();
        void setRadius(double rr);
        double getRadius();
        void setHoehe(double hh);
        double getHoehe();
        double berechneVolumen();
        void scan();
        void print();
}; // class Zylinder

```

a) Implementieren Sie die zugehörigen Methoden.

Benutzen Sie Constraints (Zusicherungen) falls dies sinnvoll ist.

b) Im Hauptprogramm soll ein Zylinder "tonne" mit den folgenden Eigenschaften erzeugt werden:

x-Koordinate des Mittelpunkts: 1

y-Koordinate des Mittelpunkts: 2

Radius des Kreises: 3

Höhe des Zylinders: 4

c) Mit der (einer einzigen) entsprechenden Methode sollen sämtliche Eigenschaften (auch das Volumen) des Zylinders "tonne" auf dem Bildschirm ausgegeben werden.

d) Mit den entsprechenden (obigen) Methoden sollen sämtliche 4 Eigenschaften des Zylinders "tonne" verdoppelt werden.

e) Mit der (einer einzigen) entsprechenden Methode sollen sämtliche Eigenschaften (auch das Volumen) des nun veränderten Zylinders "tonne" auf dem Bildschirm ausgegeben werden.

Name, Vorname:

Hilfsmittel:

Keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) Geben Sie die syntaktischen Fehler und die Fehler, die zur Laufzeit entstehen können, an.

```
#include "stdafx.h"
#include <stdio.h>

int main() {
    int i = 3;
    int v[3]={10,9,15,8};
    double w[i];

    v[2] = 10;
    w[i+1]=v[3];
    printf("r=%d\n",v[i-4]);
    return 0;
}
```

2) Es müssen die Umsätze (grösser oder gleich Null) einer Firma in ein Feld eingelesen werden.

Die Umsätze müssen über Tastatur eingegeben werden. Wenn für einen Umsatz ein Wert kleiner als Null eingegeben wird, wird die Eingabe beendet. Dann wird der Gesamtumsatz (die Summe dieser Umsätze) berechnet und auf dem Bildschirm ausgegeben.

Erstellen Sie dazu ein **Struktogramm**, kein C-Programm !!

Bemerkung:

Es dürfen keine nicht reservierten Zellen eines Feldes überschrieben werden.

Lösung

1)

```
int main(){
    int i = 3;
    int v[3]={10,9,15,8};    // zu viele Elemente
    double w[i];             // i muss Konstante sein

    v[2] = 10;
    w[i+1]=v[3];             // v[3] nicht reservierter Speicher
    printf("r=%d\n",v[i-4]); // v[i-4] nicht reservierter Speicher
    return 0;
}
```

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- Programme dürfen beim Kompilieren keine Warnungen und keine Fehlermeldungen bringen.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

Bemerkung:

Alle für eine Funktion "wichtigen" Informationen müssen als **formale Parameter** realisiert werden, **nicht** als lokale Variablen.

1) Schreiben Sie eine Funktion, die das Volumen eines Quaders berechnet.
Erzeugen Sie dazu auch einen Aufruf im Hauptprogramm.

2) Schreiben Sie eine Funktion, die den Benzinverbrauch pro 100 km berechnet.
Erzeugen Sie dazu auch einen Aufruf im Hauptprogramm.

3) Schreiben Sie eine Funktion die einen Taschenrechner simuliert:
Abhängig vom Zeichen ('+', '-', '*', '/') und zwei Zahlen soll die jeweilige Rechenoperation gemacht werden.
Erzeugen Sie dazu auch einen Aufruf im Hauptprogramm.

4) Schreiben Sie die Funktion *high*, die die Zahl n^n einer ganzen Zahl n bestimmt.
Erzeugen Sie dazu auch einen Aufruf im Hauptprogramm.

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- Programme dürfen beim Kompilieren keine Warnungen und keine Fehlermeldungen bringen.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programnteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

Bemerkung:

- a) Alle für eine Funktion "wichtigen" Informationen müssen als **formale Parameter** realisiert werden, **nicht** als lokale Variablen.
- b) Innerhalb einer Funktion darf keine Ausgabe auf dem Bildschirm gemacht werden.

1) Schreiben Sie eine Funktion, die das Minimum dreier Zahlen (Datentyp jeweils integer) berechnet.

Erzeugen Sie dazu auch einen Aufruf in einem syntaktisch korrekten Hauptprogramm.

2) Schreiben Sie eine Funktion, die den Ersatzwiderstand einer Parallelschaltung von 2 Widerständen berechnet (Datentyp jeweils double).

Bei einem Widerstandwert kleiner oder gleich Null, soll am Rückgabewert erkennbar sein, welcher Widerstand kleiner oder gleich Null war.

Erzeugen Sie dazu auch einen Aufruf in einem syntaktisch korrekten Hauptprogramm.

Bemerkung:

$$\frac{1}{R_G} = \frac{1}{R_1} + \frac{1}{R_2}$$

3) Definition:

Die Fakultät einer natürlichen Zahl n ist wie folgt definiert:

$$n! = n * (n-1) * (n-2) * \dots * 1$$

Beispiel: $4! = 4 * 3 * 2 * 1$, $1! = 1$ und per Definition: $0! = 1$

Schreiben Sie die Funktion *fak*, die von einer natürlichen Zahl n deren Fakultät (also n !) bestimmt. Erzeugen Sie dazu auch einen Aufruf in einem syntaktisch korrekten Hauptprogramm.

4) Ein Anfangskapital K wird zu einem bestimmten Zinssatz p eine bestimmte Anzahl Jahre n auf einem Konto angelegt, ohne die Zinsen abzuheben.

Schreiben Sie eine Funktion, die das Endkapital K_n nach n Jahren berechnet.

Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- Programme dürfen beim Kompilieren keine Warnungen und keine Fehlermeldungen bringen.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

Bemerkung:

- a) Alle für eine Funktion "wichtigen" Informationen müssen als **formale Parameter** realisiert werden, **nicht** als lokale Variablen.
- b) Innerhalb einer Funktion darf keine Ausgabe auf dem Bildschirm gemacht werden.

1) Schreiben Sie eine Funktion, die das Maximum dreier Zahlen (Datentyp jeweils integer) berechnet.

Erzeugen Sie dazu auch einen Aufruf in einem syntaktisch korrekten Hauptprogramm.

2) Schreiben Sie die Funktion *brutto_netto*, die aus einem Betrag, der Mehrwertsteuer in Prozent (z.B. 16) und einem Modus **entweder** den Bruttopreis oder den Nettopreis bestimmt.

3) Definition:

Die Quersumme einer ganzen Zahl ist die Summe seiner Ziffern.

Beispiele:

Die Quersumme von 367 ist: $3 + 6 + 7 = 16$

Die Quersumme von 1996 ist: $1 + 9 + 9 + 6 = 25$

Schreiben Sie eine Funktion, die von einer ganzen Zahl n ($n \geq 0$) die Quersumme berechnet.

4) Schreiben Sie eine Funktion, die einen Bruch maximal kürzt:

Beispiel: $18 / 24 = 3 / 4$