

KLAUSUR 1 Programmierpraktikum 2BK11 27.11.2006 Zeit: 45 Minuten
Gruppe A Name, Vorname:

Hilfsmittel:
keine

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß auf das Aufgabenblatt 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 (keine Warnungen vom Compiler).
- 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.

Bemerkungen:

Der Compiler darf keine Warnungen erzeugen !!!

AUFGABEN

1) Ein Schüler will unbedingt die Probe (für eine Gleichung machen) und will dabei den Wert folgenden mathematischen Terms berechnen:

$$\frac{17}{5} \cdot \frac{423}{23}$$

Schreiben Sie ein C-Programm, das den Wert dieses Terms berechnet und ihn auf dem Bildschirm ausgibt.

Geben Sie 2 Möglichkeiten (in demselben Programm) an, wie man dies programmtechnisch realisieren kann.

2) Schreiben Sie ein C-Programm, das den Nettopreis einer Ware berechnet:

Der Bruttopreis und der Prozentsatz soll über Tastatur eingegeben werden.

Auf dem Bildschirm soll dann der Prozentwert und der Nettopreis ausgegeben werden.

3) Schreiben Sie ein Programm, das das zu einer Zahl zugehörige ASCII-Zeichen auf dem Bildschirm ausgibt. Welches ASCII-Zeichen bekommt man von der Zahl 321 ?

Begründen Sie.

KLAUSUR 1 Programmierpraktikum 2BK11 28.11.2006 Zeit: 45 Minuten
Gruppe 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 Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!
- Bei praktischen Arbeiten am Computer bitte folgendes beachten:
- NUR **ablauffähige** C-Programme werden bewertet ! **Jede** Aufgabe als eigenes Projekt (keine Warnungen vom Compiler).
- 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.

Bemerkungen:

Der Compiler darf keine Warnungen erzeugen !!!

AUFGABEN

- 1) a) Schreiben Sie ein C-Programm, das den Mittelwert von 2 über Tastatur eingegebenen Noten (Fließkommazahlen) berechnet.
b) Für die Zeugniserstellung muss dieser Notendurchschnitt abgerundet werden (z.B. wird aus dem Mittelwert von 3,9 die der Zeugnisnote 3).
c) In den Bemerkungen auf dem Zeugnis soll noch ausgegeben werden, wie groß der im Zeugnis abgeschnittene Nachkommateil des Mittelwerts ist (im Beispiel oben also 0,9).

2) Das Volumen einer Kugel berechnet sich wie folgt:

$$V = \frac{4}{3} \cdot \pi \cdot r^3$$

Schreiben Sie das dazu passende C-Programm. Das Volumen muss in einer Variablen mit dem Datentyp float gespeichert werden. Die Konstante pi muss mit einer define-Direktive realisiert werden.

3) In einer Hardwareumgebung mit sehr wenig Arbeitsspeicher muss schonend mit dieser Ressource umgegangen werden. Deshalb darf das folgende Programm nur 3 Variablen (mit den Namen z1, z2, z3) verwenden. Schreiben Sie das folgende "quick and dirty" C-Programm (EVA-Prinzip muss nicht eingehalten werden):

- a) Geben Sie 3 Zahlen ein.
- b) Speichern Sie die Summe dieser 3 Zahlen in einer Variablen und geben die Summe auf dem Bildschirm aus.
- c) Speichern Sie den Mittelwert dieser 3 Zahlen in einer Variablen und geben den Mittelwert auf dem Bildschirm aus.
- d) Speichern Sie das Produkt dieser 3 Zahlen in einer Variablen und geben das Produkt auf dem Bildschirm aus.
- e) Geben Sie alle 3 über Tastatur eingegebenen Zahlen auf dem Bildschirm aus

KLAUSUR 1 Programmierpraxis 2BKI1 Nachtermin 1 Zeit: 45 Minuten

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.
- Programme dürfen beim Kompilieren keine Warnungen und keine Fehlermeldungen bringen.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

Schreiben Sie ein C-Programm, das den Ersatzwiderstand einer Parallelschaltung von 3 Widerständen berechnet (Datentyp jeweils float).

Wurde ein Widerstandwert kleiner oder gleich Null eingegeben, muß das Programm eine entsprechende Meldung auf den Bildschirm bringen (im Ausgabeteil !).

Dies muß mit dem EVA-Prinzip realisiert werden.

Bemerkung:

$$\frac{1}{R_G} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

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 bedeutet Priorität in der Programmiersprache C ? (2P)

Geben Sie ein Beispiel.

2) Was bedeutet Assoziativität in der Programmiersprache C ? (2P)

Geben Sie ein Beispiel.

3) Was gibt der folgende Programmausschnitt auf dem Bildschirm aus ? Begründen Sie! (16P)

```
int x=10;
int a=0;
int b=0;
int c=0;
if(x=2)
    printf("Ausgabe1\n");
if(a==(b==c))
    printf("Ausgabe2\n");
if(123)
    printf("Ausgabe3\n");
if(5-3/4==5+3/4)
    printf("Ausgabe4\n");
if(x=(2&&8))
    printf("Ausgabe5\n");
    printf("Ausgabe6\n");
if(x||b)
    printf("Ausgabe7\n");
if((x=2)&&8)
    printf("Ausgabe8\n");
```

4) a) Erscheint beim Kompilieren des folgenden syntaktisch korrekten Programmausschnitts eine Warnung ? Begründen Sie genau! (8P)

b) Welchen Wert hat f ?

```
...
float f;
f = 3/5*(3.5 + 4 * 2.5);
...
```

5) Ersetzen Sie folgenden syntaktisch korrekten Programmausschnitt durch mehrere einseitige Verzweigungen: (6P)

Programmausschnitt:

```
...  
if(a==b && b==c || a+b==c)  
    x=1;  
else  
    x=2;
```

6) Sie werden beauftragt ein C-Programm (Taschenrechner) zu schreiben, das abhängig von der Eingabe eines Zeichens (+, -, *, /) entweder die Summe oder die Differenz oder das Produkt oder den Quotienten zweier Zahlen liefert.

Beachten Sie bitte das EVA-Prinzip und berücksichtigen Sie, dass der Anwender auch andere Zeichen eingeben kann, als +, -, *, /. (16P)

Lösungen:

1) Priorität gibt den Vorrang eines Operators in einem Ausdruck an.

Beispiel: Punkt vor Strich.

2) Assoziativität gibt an, in welcher Reihenfolge (von links nach rechts oder von rechts nach links) Operatoren mit der gleichen Priorität abgearbeitet werden.

Beispiel: * wird von links nach rechts abgearbeitet.

3)

a) Ausgabe1: Der Wert des Ausdrucks (Zuweisungsausdrucks) $x=2$ ist 2. Der Wert 2 wird als wahr interpretiert.

b) keine Ausgabe2: Der Wert des Ausdrucks $b==c$ ist 1, also wahr, da b und c jeweils 0 ist und damit b gleich groß wie c ist. Der Wert des Ausdrucks $a==(b==c)$, also konkret $a==1$, ist 0, also falsch, da a gleich 0 ist. Der Wert 0 wird als falsch interpretiert.

c) Ausgabe3: 123 wird als wahr interpretiert.

d) Ausgabe4: $5-3/4$ ist 5, da $3/4$ Null ist. $5+3/4$ ist 5, da $3/4$ Null ist. Also ist $5-3/4$ gleich $5+3/4$. Also hat der Ausdruck $5-3/4==5+3/4$ den Wert 1 und wird als wahr interpretiert.

e) Ausgabe5: 2 und 8 wird jeweils als wahr interpretiert. Deshalb ist $2 \ \&\& \ 8$ wahr und hat den Wert 1. Der Wert des Zuweisungsausdrucks $x=(2 \ \&\& \ 8)$ ist deshalb 1 und wird als wahr interpretiert.

f) Ausgabe6: Nach `if(x=(2&&8))` steht keine Klammer {
Deshalb besteht der if-Körper der dieser if-Anweisung nur aus einer Anweisung. Die dieser Anweisung folgende Anweisung wird damit auf jeden Fall ausgeführt.

g) Ausgabe7: Der Wert von x ist 2 und der Wert von b ist 0. Damit wird 2 als wahr und 0 als falsch interpretiert. Also hat der Ausdruck $x \ || \ b$ den Wert 1 und ist damit wahr.

h) Ausgabe8: Der Wert des Ausdrucks (Zuweisungsausdrucks) $x=2$ ist 2. Der Wert 2 wird als wahr interpretiert. 2 und 8 wird jeweils als wahr interpretiert. Deshalb ist $2 \ \&\& \ 8$ wahr und hat den Wert 1. Der Wert des Zuweisungsausdrucks $(x=2) \ \&\& \ 8$ ist deshalb 1 und wird als wahr interpretiert.

4) Es wird zuerst $3/5$ berechnet (Assoziativität von links nach rechts). Der Wert ist 0.

4 ist integer, 2.5 ist double. Also wird 4 in double 4.0 umgewandelt. Das Ergebnis 10.0 ist damit double. 3.5 ist double und damit ist $3.5 * 10.0$ auch double. Dieser Wert wird mit integer 0 multipliziert, wobei deshalb vorher die integer 0 in double 0.0 umgewandelt wird und den Wert double 0.0 ergibt.

Da double in float abgespeichert wird, wird die double Zahl 0.0 in float umgewandelt. Dabei kann ein Datenverlust entstehen. Deshalb gibt der Compiler eine Warnung aus.

5)

```
if(a==b && b==c || a+b==c)
    x=1;
if(!(a==b && b==c || a+b==c))
    x=2;
```

6)

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

int main(){
    double zahl1, zahl2;
    double ergebnis;
    char rechenzeichen = 's';    // schlecht

    // Eingabeteil
    printf("Bitte die 1. Zahl eingeben\n");
    scanf("%lf", &zahl1);
    fflush(stdin);

    printf("Bitte die 2. Zahl eingeben\n");
    scanf("%lf", &zahl2);
    fflush(stdin);

    printf("Bitte das Rechenzeichen eingeben\n");
    scanf("%c", &rechenzeichen);
    fflush(stdin);

    //Verarbeitungsteil
    if(rechenzeichen == '+'){
        ergebnis = zahl1 + zahl2;
        rechenzeichen = 'g';    // gut
    }

    if(rechenzeichen == '-'){
        ergebnis = zahl1 - zahl2;
        rechenzeichen = 'g';    // gut
    }

    if(rechenzeichen == '*'){
        ergebnis = zahl1 * zahl2;
        rechenzeichen = 'g';    // gut
    }

    if(rechenzeichen == '/'){
        ergebnis = zahl1 / zahl2;
        rechenzeichen = 'g';    // gut
    }

    // Ausgabeteil
    if(rechenzeichen == 'g'){
        printf("Ergebnis = %f\n", ergebnis);
    }
    else{
        printf("Bitte richtiges Rechenzeichen eingeben\n");
    }

    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.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) a) Es soll über Tastatur eine **ganze** Zahl eingegeben werden.

Ist diese Zahl eine Schulnote zwischen 1 und 4 (je einschließlich) wird die Meldung "Prüfung bestanden" ausgegeben.

Ist diese Zahl die Schulnote 5 oder 6 wird die Meldung "Prüfung nicht bestanden" ausgegeben.

Sonst wird die Meldung "Diese Zahl ist keine Note" ausgegeben.

Realisieren Sie dies durch ein C-Programm (Programmausschnitt) nur mit switch- Anweisung (keine if-else-Anweisung, keine if-Anweisung). (5P)

b) Erklären Sie die if-else-if-Ketten an einem Beispiel (Programmausschnitt). (5P)

c) Was kann man mit einer if-else-Anweisung, aber nicht mit einer switch-Anweisung ? (4P)

d) Was kann man mit einer switch-Anweisung, aber nicht mit einer if-else-Anweisung ? (2P) (gibt es so einen Fall?)

2) Was gibt der folgende Programmausschnitt auf dem Bildschirm aus ? Begründen Sie! (16P)

```
int x=10;
int a=0;
int b=0;
int c=0;
if(x=2)
    printf("Ausgabe1\n");
if(a==b==c)
    printf("Ausgabe2\n");
if(123)
    printf("Ausgabe3\n");
if(5-3/4==5+3/4)
    printf("Ausgabe4\n");
if(x=(2&&8))
    printf("Ausgabe5\n");
    printf("Ausgabe6\n");
if(x||b)
    printf("Ausgabe7\n");
if((x=2)&&8)
    printf("Ausgabe8\n");
```


3) Es soll der Ersatzwiderstand einer Parallelschaltung von 3 Widerständen berechnet werden. Wurde ein Widerstandwert kleiner oder gleich Null eingegeben, muß das Programm sofort beendet werden und eine entsprechende Meldung auf den Bildschirm ausgegeben werden. (insbesondere darf dann nicht mehr ein weiterer Widerstand eingegeben und der Ersatzwiderstand berechnet werden).

Dies muß mit dem EVA-Prinzip realisiert werden.

Außerdem muß in der Ausgabe noch angegeben werden, welcher Widerstandswert (z.B. 1. Widerstandwert) falsch eingegeben wurde.

Erstellen Sie das dazugehörige Struktogramm.

(18P)

Bemerkungen:

a)
$$\frac{1}{R_G} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

b) Es widerspricht nicht dem EVA-Prinzip, wenn z.B. im Eingabeteil in bestimmten Variablen bestimmte Zustände abgespeichert werden (z.B: ob der 3. Widerstandswert kleiner oder gleich 0 ist) und diese dann im Ausgabeteil abgeprüft werden.

Lösungen:

1)

a)

```
int main(){
    int note;
    printf("Bitte eine ganze Note eingeben\n");
    scanf("%c", &note);

    switch(note){    // int oder char
        case 1:
        case 2:
        case 3:
        case 4:
            printf("Prüfung bestanden \n");
            break;    // nicht abweisend

        case 5:
        case 6:
            printf("Prüfung nicht bestanden \n");
            break;

        default:    // falls kein case-Fall zutrifft
            printf("Dies ist keine Note \n");
            break;
    }
    return 0;
}
```

b)

```
int main(){
    int note;
    printf("Bitte eine ganze Note eingeben\n");
    scanf("%c", &note);

    if(note==1){
        printf("Note sehr gut\n");
    }
    else if(note==2){
        printf("Note gut\n");
    }
    else if(note==3){
        printf("Note befriedigend\n");
    }
    else if(note==4){
        printf("Note ausreichend\n");
    }
    else if(note==5){
        printf("Note mangelhaft\n");
    }
    else if(note==6){
        printf("Note ungenügend\n");
    }
    else{
        printf("Dies ist keine Note\n");
    }
}
```

c)

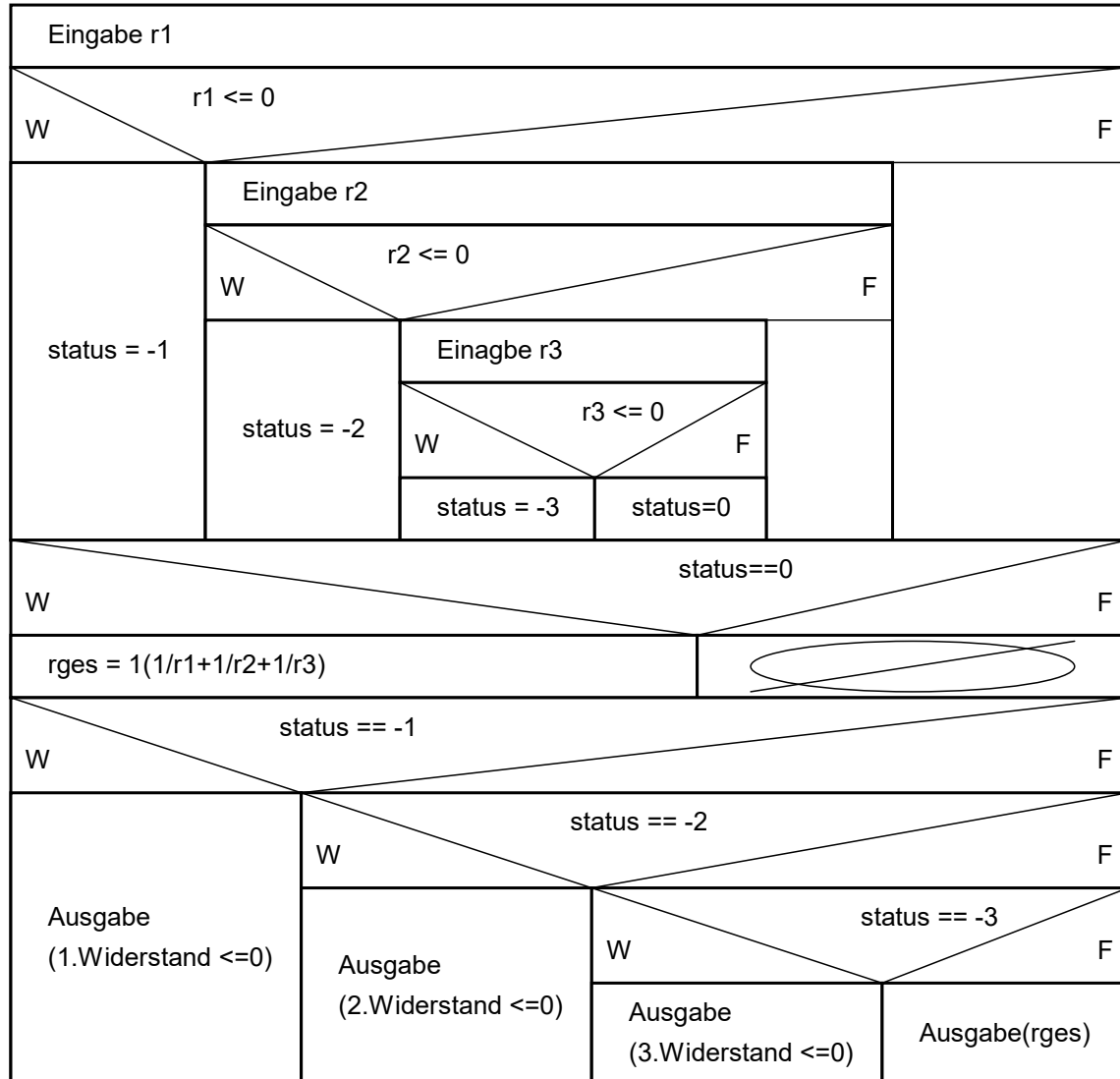
switch: nur für ganzzahlige oder char Datentypen verwendbar

d) nichts

2)
siehe Lösungen der letzten Klassenarbeit !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

3)

Widerstandsberechnung



KLAUSUR 3 Programmiertheorie 2BKI1 1.3.2007 Zeit: 45 Minuten

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) Erstellen Sie ein Struktogramm, in dem zwei ganze Zahlen (Zähler und Nenner) über Tastatur eingegeben werden. Außerdem soll zusätzlich noch folgendes gemacht werden:

a) 35P

Voraussetzung:

Es muss angenommen werden (ohne abzu prüfen), dass die zwei ganzen Zahlen jeweils > 0 sind.

Der ggt (größter gemeinsamer Teiler) von diesen zwei Zahlen muss nach folgender Vorschrift berechnet werden:

Zuerst die zwei Zahlen jeweils durch 1 teilen, dann jeweils feststellen, ob 1 die erste Zahl und die zweite Zahl teilt.

Dann das gleiche für 2 machen, dann das gleiche für 3 machen, usw. bis die Zahl durch die geteilt werden soll, gleich dem Nenner ist.

Beispiel:

zaehler = -6

nenner = 4

1 teilt -6 und 1 teilt 4 \implies 1 ist vorläufiger ggt

2 teilt -6 und 2 teilt 4 \implies 2 ist vorläufiger ggt

3 teilt -6 und 3 teilt nicht 4 \implies 2 ist immer noch vorläufiger ggt

4 teilt nicht -6 und 4 teilt 4 \implies 2 ist immer noch vorläufiger ggt

Bemerkung:

Wer diese Aufgabe nicht lösen kann, kann unabhängig davon Aufgabe b) lösen.

Siehe Aufgabe b)

b)

15P

Voraussetzung:

Zähler und Nenner dürfen beliebige ganze Zahlen sein (auch negative Zahlen bzw. gleich 0).

Zähler und Nenner müssen maximal gekürzt und das Ergebnis (genauso wie in den folgenden Beispielen) auf dem Bildschirm ausgegeben werden.

Beispiele:

$$18 / 24 = 3 / 4$$

$$-9 / 6 = -3 / 2$$

$$70 / -90 = -7 / 9$$

$$-24 / -30 = 4 / 5$$

$$0 / -5 = 0$$

$$0 / 8 = 0$$

Bemerkung:

In dieser Aufgabe muss (und darf) nicht mehr der Teil des Struktogramms von Aufgabe a) verwendet (und eingebaut) werden, der den ggt zweier Zahlen berechnet.

Stattdessen soll in dem Struktogramm ein Ausdruck der Form:

erg = berechne_ggt (z1, z2)

als eine Anweisung benutzt werden, in der z1, z2 die Zahlen bedeuten, die jeweils > 0 sind und erg der größte gemeinsame Teiler ist.

Statt z1, z2 und erg müssen natürlich selbstgewählte, selbsterklärendere Variablennamen verwendet werden!

KLAUSUR 3 Programmiertheorie 2BKI1 Nachtermin 1 Zeit: 30 Minuten

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.
- Programme dürfen beim Kompilieren keine Warnungen und keine Fehlermeldungen bringen.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1)

a) Eine natürliche Zahl n ($n \geq 2$) heißt Primzahl, wenn sie nur durch 1 und sich selbst teilbar ist. Beispiele für Primzahlen: 2, 3, 5, 7, 11, 13,

Erstellen Sie ein Struktogramm, das bestimmt, ob eine ganze Zahl n ($n \geq 2$) eine Primzahl ist: Sie geben über Tastatur eine ganze Zahl n ein. Auf dem Bildschirm soll dann ausgegeben werden, ob diese Zahl eine Primzahl ist oder nicht.

Falls eine ganze Zahl < 2 eingegeben wird, muss das Struktogramm entsprechend reagieren.

Tipp:

Feststellen, ob 1 die Zahl n teilt, dann
feststellen, ob 2 die Zahl n teilt, dann
feststellen, ob 3 die Zahl n teilt, usw.

b) Führen Sie ein paar Tests durch.

KLAUSUR 4 Programmiertheorie 2BKI1 14.6.2007 Zeit: 60 Minuten

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)

12P

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.

- Welchen Wert hat der Inhalt der Variablen zahl und quad an der Stelle [1] im Programm ?
- Welchen Wert hat z und zq beim Aufruf von quadrat(zahl, &quad) ?
- Was bewirkt die Anweisung `*zq = z * z` an welcher Adresse im Arbeitsspeicher ?
(konkreten Wert der Adresse und deren Inhalt angeben!)
- Welchen Wert hat der Inhalt der Variablen zahl und quad an der Stelle [2] im Programm ?

2)

13P

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)

12P

Betrachten Sie den folgenden Programmausschnitt:

```
...
int main() {
    float r;
    float u;
    r = 2;
    u = 3;
    → 
    berechne_umfang (r, u);
    → 
    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
<code>r</code>	0120	?
<code>u</code>	0130	?

im Arbeitsspeicher reserviert.

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

4)

13P

Programmieren Sie anhand der folgenden Beschreibung die dazugehörige Funktion:

```

/*****
/**
/**  int ersatz(double r1, double r2, int mod, double *rg)  **/
/**
/**
/**#*****/
/*

```

Parameter:

```

(i) double r1>0:    erster Widerstandswert
(i) double r2>0:    zweiter Widerstandswert
(i) int mod:         10: Parallelschaltung
                    20: Reihenschaltung
(o) double *rg:     Gesamtwiderstand

```

Return:

```

(o) 0: Parallelschaltung oder Reihenschaltung 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 Reihenschaltung), den Widerstandswerten r1 und r2 den Ersatzwiderstand (Gesamtwiderstand) rg der Widerstandsschaltung.

*/

Lösungen

- 1) 12P
a) zahl: 4, quad: 8 1P + 1P
b) z: 4, zq: 04711 1P + 2P
c) In den Inhalt der Adresse 04711 wird der Wert 16 geschrieben. 4P
d) zahl: 4, quad: 16 1P + 2P

2) 13P
/*****/
/** **/
/** 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:

- (o) 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

*/

Jede fehlende Zusicherung: -2 P

z2!=0 ist keine richtige Zusicherung, da dann z.B. auch 3 * 0 verboten würde: -2 P

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, desto mehr "Intelligenz" muss er in die Funktion packen.

- 3) 12P
a) $r = 2, u = 3$ 1P + 1P
b) radius = 2, umfang = 3 1P + 2P
c) nichts 4P
d) $r = 2, u = 3$ 3P

4) 13P

```
int ersatz(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 5 Programmiertheorie 2BKI1 28.6.2007 Zeit: 45 Minuten*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 !!!!

fopen	Zugriff	Datei existiert schon	Datei existiert noch nicht
r	R	Datei wird zum Lesen geöffnet	ergibt Fehler
w	W	Datei wird zum Schreiben geöffnet, Inhalt wird gelöscht	erzeugt Datei, (Modus: siehe Spalte links)
r +	R/W	Datei wird zum Lesen und Schreiben geöffnet.	ergibt Fehler
w +	R/W	Datei wird zum Lesen und Schreiben geöffnet, Inhalt wird gelöscht	erzeugt Datei, (Modus: siehe Spalte links)
a	W	Datei wird nur zum Schreiben im Anfügemodus geöffnet.	erzeugt Datei (Modus: siehe Spalte links)
a+	R/W	Datei wird zum Lesen und Schreiben im Anfügemodus geöffnet.	erzeugt Datei (Modus: siehe Spalte links)

AUFGABEN

1) Was ist ein Dateizeiger ? **3P**

2) Was bedeutet "eine Datei befindet sich im Anfügemodus" ? **3P**

3) Was liefert fopen zurück, wenn ein Fehler passiert? **3P**

4) Was machen folgende Anweisungen und welche Werte hat erg jeweils? **9P**
Begründen Sie!

a) `erg = fseek(stream, 5, SEEK_SET);`

b) `erg = fseek(stream, 0, SEEK_CUR);`

c) `erg = fseek(stream, 3, SEEK_END);`

5) Eine Datei soll zum Lesen und Schreiben geöffnet werden (kein Anfügemodus). **1P**
Außerdem soll es einen Fehler geben, wenn die Datei nicht existiert. Welchen Modus müssen Sie wählen?

6) Eine Datei soll zum Lesen und Schreiben im Anfügemodus geöffnet werden. **1P**
Außerdem soll es keinen Fehler geben, wenn die Datei nicht existiert, sondern sie soll dann zum Lesen und Schreiben im Anfügemodus geöffnet werden.
Welchen Modus müssen Sie wählen?

7) Wenn die Datei "test1.txt" noch nicht existiert soll sie zum Lesen und Schreiben **10P**
(kein Anfügemodus) geöffnet und angelegt werden. Wenn diese Datei dagegen schon existiert, soll sie sicherheitshalber zum Lesen und Schreiben im Anfügemodus geöffnet werden.
Erstellen Sie den dazugehörigen Programmteil (Programmausschnitt, kein vollständiges Programm).

8) **20P**
a) Beschreiben Sie die fett gedruckten Anweisungen des folgenden syntaktisch korrekten Programms
b) Was gibt das Programm auf dem Bildschirm aus?
c) Begründen Sie.

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

int main() {
    FILE *stream;
    int erg;
    char mychar='x';

    stream = fopen("C:\\textmode.txt", "w+" );
    erg = fseek(stream, 0, SEEK_SET);
    fprintf(stream, "%c", '\n');
    erg = fseek(stream, 0, SEEK_SET);
    fscanf(stream, "%c", &mychar);
    printf("Ausgabe=%d\n", mychar);
    erg = fclose(stream);

    stream = fopen("C:\\textmode.txt", "rb+" );
    erg = fseek(stream, 0, SEEK_SET);
    fscanf(stream, "%c", &mychar);
    printf("Ausgabe=%d\n", mychar);
    erg = fclose(stream);
    return 0;
}
```

Lösungen

- 1) Ein Dateizeiger zeigt auf die aktuelle Position in der Datei, von der ab gelesen oder geschrieben wird.
- 2) Daten werden beim Schreiben immer, egal wo sich der Dateizeiger befindet, am Dateiende eingefügt.

3) NULL

4)

a) `erg = fseek(stream, 5, SEEK_SET);`

verschiebt relativ zum Dateianfang um 5 Bytes in Richtung Dateiende;

`erg = 0`, falls Datei mindestens aus 5 Bytes besteht.

b) `erg = fseek(stream, 0, SEEK_CUR);`

verschiebt relativ zur aktuellen Position um 0 Bytes in Richtung Dateianfang

`erg = 0`;

c) `erg = fseek(stream, 3, SEEK_END);`

verschiebt relativ zum Dateiende die aktuellen Position um 3 Bytes in Richtung Dateiende;

`erg` wird negativ, da beim Schreiben über das Dateiende hinaus ein Fehler gemacht wird

5) `r+`

6) `a+`

7)

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

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

```
int main() {
    FILE *stream;
    int erg;

    stream = fopen("C:\\test1.txt", "r" );
    // Datei existiert noch nicht
    if(stream==NULL)
        // Dann wird die Datei im R/W Modus angelegt
        stream = fopen("C:\\test1.txt", "w+" );
    else{ // Datei existiert schon
        // also wird sie zuerst geschlossen
        erg=fclose(stream);
        // um sie dann im gewünschten Modus wieder öffnen zu können
        stream = fopen("C:\\test1.txt", "a+" );
    }
    fprintf(stream, "%c", 'x');
}
```

8)
10
13

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

int main(){
    FILE *stream;
    int erg;
    char mychar='x';
    // Datei wird im Textmodus geöffnet bzw. neu angelegt
    stream = fopen("C:\\textmode.txt", "w+" );
    // Dateizeiger wird an den Dateianfang verschoben
    erg = fseek(stream, 0, SEEK_SET);
    // 0xD und 0xA wird reingeschrieben, da Textmodus
    fprintf(stream, "%c", '\n');
    // Dateizeiger wird an den Dateianfang verschoben
    erg = fseek(stream, 0, SEEK_SET);
    // Beim Auslesen wird 0xD unterdrückt (weggeworfen), da Textmodus
    // also steht 0xA in mychar
    fscanf(stream, "%c", &mychar);
    // 10 wird ausgegeben, der Dezimalwert von 0xA
    printf("Ausgabe=%d\n", mychar);
    // Datei wird geschlossen
    erg = fclose(stream);

    // Die schon angelegte Datei wird im Binärmodus
    // zum Lesen und Schreiben geöffnet
    stream = fopen("C:\\textmode.txt", "rb+" );
    // Dateizeiger wird an den Dateianfang verschoben
    erg = fseek(stream, 0, SEEK_SET);
    // Beim Auslesen wird das erste Zeichen 0xD ausgelesen, da Binärmodus
    // also steht 0xD in mychar
    fscanf(stream, "%c", &mychar);
    // 13 wird ausgegeben, der Dezimalwert von 0xD
    printf("Ausgabe=%d\n", mychar);
    // Datei wird geschlossen
    erg = fclose(stream);
    return 0;
}
```