

C-ÜBUNGSAUFGABEN VERZWEIGUNGEN 1

Vorbemerkungen:

Bevor ein Programm am Rechner eingetippt werden darf, müssen folgende Voraussetzungen erfüllt sein:

V1) Der Algorithmus muß vom Entwickler als Struktogramm dargestellt werden.

Dabei soll rechts in geklammerter Form der Eingabe- Verarbeitungs- und Ausgabeteil gekennzeichnet werden.

V2) Dann muß der Entwickler einen Tester (Schüler in der Klasse) suchen, der prüft, ob die Syntax des Struktogramms korrekt ist. Falls die Syntax korrekt ist, muß der Tester ein ausführliches Testprotokoll des Struktogramms erstellen, wobei er den Test mit seiner Unterschrift bestätigt.

Der Test findet ohne den Entwickler statt (am besten einen großen Abstand halten).

Nach dem Test erhält der Entwickler das Testprotokoll. Steht dort, daß der Algorithmus falsch ist, muß der Entwickler sein Struktogramm verbessern und danach wieder einen Tester suchen.

Falls das Struktogramm irgendwann vom Tester als korrekt eingestuft wurde, wird es dem Lehrer zur Unterschrift vorgelegt und für die Programmierung freigegeben.

V3) Dann muß dieses Struktogramm in die Programmiersprache C umgesetzt werden.

Dabei soll (jeweils als Kommentar) der Eingabe- Verarbeitungs- und Ausgabeteil gekennzeichnet werden.

Bemerkungen für die Programmierer

B1)

Am Anfang des Programms muss für den Anwender eine vollständige Programminformation (Programminfo) auf dem Bildschirm ausgegeben werden, in der beschrieben steht, wie der Anwender das Programm zu bedienen hat und was das Programm macht (z.B. kann dies durch die Angabe eines Beispiels ergänzt werden).

B2)

Es widerspricht nicht dem EVA-Prinzip, wenn z.B. im Eingabeteil in einer Variablen (selbsterklärende Variablennamen wählen, wie z.B. **beenden**, **zustand**, **fehler**, **error**, **status**, usw.) bestimmte Zustände abgespeichert werden (z.B: ob eine Zahl keine Note ist) und diese dann im Ausgabeteil abgeprüft werden. Bevor die nächste Zahl eingegeben wird, muß dann der Wert dieser Variablen abgeprüft werden.

B3)

In jedem Programm das reservierte Wort **return** nur genau einmal (am Schluß) verwenden. Wenn das Programm z.B. sofort nach einer Eingabe beendet werden soll, bitte B2) beachten.

1) Eine "einfache" Aufgabe

Welche Ausgabe erscheint auf dem Bildschirm ?

Begründen Sie Ihre Entscheidung!

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

int main()
{
    double summe = 0;
    double y = 0.1;
    summe = summe + y;
    summe = summe + y;
    summe = summe + y;
    summe = summe + y;
    summe = summe + y;
    summe = summe + y;
    summe = summe + y;
    summe = summe + y;
    summe = summe + y;

    if(summe==1.0) {
        printf("summe im if-Teil=%f\n", summe);
    }
    else{
        printf("summe im else-Teil=%f\n", summe);
    }
    return 0;
}
```

2) Absolutwert (Betrag)

Schreiben Sie ein Programm, das zu einer über Tastatur eingegebenen Zahl den Absolutwert (Betrag) ausgibt. Ein Absolutwert ist die Größe einer Zahl, ohne Berücksichtigung ihres Vorzeichens.

Beispiel:

7,5 --> 7,5

-3 ---> 3

3) Zahlen sortieren

a) Schreiben Sie ein Programm, das zwei über Tastatur eingegebene Zahlen der Größe nach sortiert auf dem Bildschirm ausgibt.

b) Machen Sie das Gleiche wie bei a), außer daß statt Zahlen Zeichen verwendet werden. Die Sortierung erfolgt nach der ASCII-Tabelle.

4) Mittelwert

a) Über Tastatur sollen 2 Zahlen eingegeben werden.

Wenn nach der Eingabe aller 2 Zahlen festgestellt wird, daß alle 2 Zahlen Noten sind (also die Zahl größer gleich 1 und kleiner gleich 6 sind), muß der Mittelwert der 2 Noten auf dem Bildschirm ausgegeben werden.

Sonst muß ausgegeben werden, welche Zahlen keine Note sind.

Tipp:

Fehler bei der Eingabe können z.B. in der integer-Variablen "fehler" so verwaltet werden:

fehler = 0 : Keine Fehler bei der Eingabe (d.h. alle 2 Zahlen sind Noten)

fehler = 1 (=01) : 1. Zahl ist eine Note und 2. Zahl ist keine Note.

fehler = 10: 1. Zahl ist keine Note und 2. Zahl ist eine Note.

fehler = 11: 1. Zahl ist keine Note und 2. Zahl ist keine Note.

D.h. Eine 1 in der Zahl bedeutet welche Zahl keine Note ist !!

b) Schreiben Sie ein Programm, das den Mittelwert zweier Noten (z.B. in Mathematik) berechnet:

Sie geben über Tastatur zwei reelle Zahlen ein.

Wenn eine der Zahlen keine Note ist, (also die Zahl kleiner als 1 oder größer als 6 ist), soll das Programm sofort beendet werden und die Meldung "Die zuletzt eingegebene Zahl ist keine Note" auf dem Bildschirm ausgegeben werden.

Wenn alle zwei Zahlen Noten sind, (also die Zahlen größer gleich eins und kleiner gleich sechs sind), soll die Meldung "Der Mittelwert ihrer Noten ist: ..." und der Mittelwert der Noten auf dem Bildschirm ausgegeben werden.

Siehe Bemerkung B2)

5) Mittelwert

a) Schreiben Sie ein Programm, das den Mittelwert dreier Noten (z.B. in Computertechnik) berechnet. Siehe letzte Aufgabe Teil a)

b) Schreiben Sie ein Programm, das den Mittelwert dreier Noten (z.B. in Computertechnik) berechnet. Siehe letzte Aufgabe Teil b)

6) Mittlere Zahl

Schreiben Sie ein Programm, das die "mittlere" (d.h. zweitkleinste bzw. zweitgrößte) von 3 über Tastatur eingegebenen, ganzen Zahlen bestimmt und auf dem Bildschirm ausgibt.

7) Anzahl gleicher Zeichen

a) Schreiben Sie ein Programm, das ermittelt, ob von 3 über Tastatur eingegebenen Zeichen genau 2 davon gleich sind.

Das Ergebnis ('Genau 2 Zeichen sind gleich' bzw. 'Nicht genau 2 Zeichen sind gleich') soll auf dem Bildschirm ausgegeben werden.

b) Machen Sie das Gleiche wie bei a), außer daß statt Zeichen Zahlen verwendet werden.

8) Anzahl gleicher Zeichen

a) Schreiben Sie ein Programm, das berechnet, wieviele gleiche Zeichen in einer Folge von drei über Tastatur eingegebenen Zeichen (gespeichert in den Variablen x, y, z) vorkommen.

b) Machen Sie das Gleiche wie bei a), außer daß statt Zeichen Zahlen verwendet werden.

9) Anzahl gerader Zahlen

Schreiben Sie ein Programm, in dem drei ganze Zahlen über Tastatur eingegeben werden. Das Programm soll die Anzahl der geraden Zahlen (unter diesen drei eingegebenen) berechnen und auf dem Bildschirm ausgeben.

10) Zerlegen

Zwei positive, ganze Zahlen sollen eingelesen werden. Die größere soll nach der kleineren zerlegt werden, und das Ergebnis ist auszugeben. Die Ausgabe soll darüber Auskunft geben, ob die kleinere Zahl die größere Zahl teilt oder nicht.

Beispiel:

Eingabe: $a = 5$, $b = 14$

Ausgabe: $14 = 5 \cdot 2 + 4$, 14 ist nicht durch 5 teilbar.

11) Lösungsmenge bestimmen

Schreiben Sie ein Programm, das die Lösungsmenge der Gleichung:

$a \cdot x = b$ (a, b sind beliebige, reelle, über Tastatur eingegebene Zahlen)

ermittelt und auf dem Bildschirm ausgibt.

12) Kryptologie

Zur Information:

Eine Bank hat 5 Bankdirektoren.

Der Tresor der Bank kann nur dann geöffnet werden, wenn mindestens 3 Bankdirektoren anwesend sind und ihren Geheimcode dem Bankcomputer mitteilen. Aus dem jeweiligen Geheimcode der Bankdirektoren baut dann der Bankcomputer den Superschlüssel, der den Tresor öffnet.

Wie kann dies realisiert werden ?

Lösung:

I) Schlüssel erzeugen

Zur Erzeugung des Schlüssels (der an alle 5 Bankdirektoren vergeben wird), wird im Programm "Schlüssel_erzeugen" Folgendes gemacht:

Schritt1:

Man gibt eine Parabel der folgenden Form vor:

$$y = ax^2 + bx + c$$

Konkret: Man gibt Werte für a, b und c vor.

Beispiel:

$$a = -1, b = 2, c = 3$$

Schritt2:

Dann bestimmt man 5 verschiedene Punkte der Parabel und teilt jedem Bankdirektoren davon genau einen mit.

Konkret: Man gibt 5 verschiedene x-Koordinaten x_1, x_2, x_3, x_4, x_5 vor

Beispiel:

$$x_1 = 1 \quad x_2 = 2 \quad x_3 = 3 \quad x_4 = 4 \quad x_5 = 5$$

Daraus werden dann die y-Koordinaten berechnet (mit $a = -1, b = 2, c = 3$)

$$y_1 = ax^2 + bx + c = -1 \cdot 1^2 + 2 \cdot 1 + 3 = 4$$

$$y_2 = ax^2 + bx + c = -1 \cdot 2^2 + 2 \cdot 2 + 3 = 3$$

$$y_3 = ax^2 + bx + c = -1 \cdot 3^2 + 2 \cdot 3 + 3 = 0$$

$$y_4 = ax^2 + bx + c = -1 \cdot 4^2 + 2 \cdot 4 + 3 = -5$$

$$y_5 = ax^2 + bx + c = -1 \cdot 5^2 + 2 \cdot 5 + 3 = -12$$

Das ergibt dann die 5 Schlüssel (Zahlenpaare):

$$(1,4), (2,3), (3,0), (4,-5), (5,-12)$$

II) Schlüssel testen

Zum Testen der Schlüssel wird im Programm "Login" Folgendes gemacht:

Schritt1:

Es gibt die Möglichkeit 3, 4 oder 5 Schlüssel (Zahlenpaare) einzugeben, je nachdem wie viele Bankdirektoren vor dem Rechner sitzen, der den Tresor öffnen soll.

Dies muß im Programm ausgewählt werden.

Dann wird getestet, ob diese Schlüssel Punkte der Parabel mit der Funktionsgleichung $y = ax^2 + bx + c$ sind.

Beispiel:

$a = -1, b = 2, c = 3$ und 3 Bankdirektoren geben ihre Schlüssel $(1,4), (2,3), (3,0)$ ein.

Dann testet das Programm, ob Folgendes gilt:

$$-1 \cdot 1^2 + 2 \cdot 1 + 3 = 4 \quad \text{und} \quad -1 \cdot 2^2 + 2 \cdot 2 + 3 = 3 \quad \text{und} \quad -1 \cdot 3^2 + 2 \cdot 3 + 3 = 0$$

Dies ist wahr, also gibt das Programm aus: "Der Tresor wird geöffnet"

13) Spiele mit Wahrscheinlichkeiten (Glücksspiele)

1.1) Münze - Würfel

- a) Schreiben Sie ein Programm, das eine Münze simuliert, d.h. "Zahl" oder "Wappen" bzw. 0 oder 1 auf dem Bildschirm ausgibt.
- b) Schreiben Sie ein Programm, das einen Würfel simuliert, d.h. eine der Zahlen 1,2,3,4,5,6 auf dem Bildschirm ausgibt.

Beispiel für die Verwendung einer Zufallszahl:

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

int main(){
    int zufallszahl;
    // srand bitte nur EINMAL aufrufen !!!!!
    srand((unsigned)time( NULL ));
    // Zufallszahl (ganze Zahl) zwischen 0 und RAND_MAX
    zufallszahl=rand();
    return(0);
}
```

Anderes Beispiel für die Verwendung einer Zufallszahl:

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

int main(){
    double zufallszahl;
    // srand bitte nur EINMAL aufrufen !!!!!
    srand((unsigned)time(NULL));
    // Zufallszahl ist eine Fließkommazahl zwischen 0 und 1
    // also z.B. 0,3947
    zufallszahl = (double)rand()/RAND_MAX;
    printf("zufallszahl zwischen 0 und 1: %f\n",zufallszahl);
    return(0);
}
```

1.2) Zufallsspiel

Schreiben Sie ein Programm (Glücksspiel), das eine Zufallszahl (zwischen 10 und 99) erzeugt und auf dem Bildschirm ausgibt. Sind in der Zufallszahl die 2 Ziffern gleich groß, dann hat der Spieler gewonnen, sonst verloren.

1.3) Spiel Schere - Stein - Papier

Ein Spieler (Anwender) soll gegen den Computer Schere-Stein-Papier spielen.

Der Spieler setzt eine der 3 Figuren Schere, Stein oder Papier.

Dann setzt der Computer durch Zufall auch eine der 3 Figuren.

Spielausgang:

Stein - Schere --> Stein gewinnt

Stein - Papier --> Papier gewinnt

Stein - Stein --> unentschieden

Schere - Papier --> Schere gewinnt

Schere - Schere --> unentschieden

Papier - Papier --> unentschieden

14) Lösungsmenge bestimmen (schwieig)

Schreiben Sie ein Programm, das die Lösung(en) der quadratischen Gleichung:

$a x^2 + b x + c = 0$ (a, b, c sind beliebige, reelle, über Tastatur eingegebene Zahlen)

ermittelt und auf dem Bildschirm ausgibt.

Wurzel berechnen mit $\text{erg} = \text{sqrt}(\text{zahl})$, wobei erg und zahl vom Datentyp double sind.

Zusätzlich muss noch `#include <math.h>` am Programmanfang included werden

15) Logische Operatoren

Lösen Sie alle vorigen Aufgaben, indem Sie (statt logischen Operatoren in den Bedingungen) verschachtelte Verzweigungen benutzen.

16)

Lösen Sie die Aufgabe

```
if (B1 && B2)
```

```
    A1
```

```
else
```

```
    A2
```

durch eine verschachtelte if-Anweisung.

17)

Lösen Sie die Aufgabe

```
if (B1)
```

```
    A1
```

```
if (B2)
```

```
    A2
```

```
if (B3)
```

```
    A3
```

durch eine if-else-Anweisung.

18) Zahlen sortieren

Schreiben Sie ein Programm, das drei Zahlen z_1 , z_2 , z_3 wie folgt sortiert (der Größe nach).

- Sortiere zuerst die z_1 und z_2 derart, dass in z_1 die kleinere und in z_2 die größere gespeichert ist.
- Sortiere dann z_2 und z_3 derart, dass in z_2 die kleinere und in z_3 die größere gespeichert ist.
- Da jetzt aber z_2 wieder kleiner als z_1 sein kann, muss man wieder z_1 und z_2 sortieren, derart, dass in z_1 die kleinere und in z_2 die größere gespeichert ist.

19) Dreiecksberechnungen

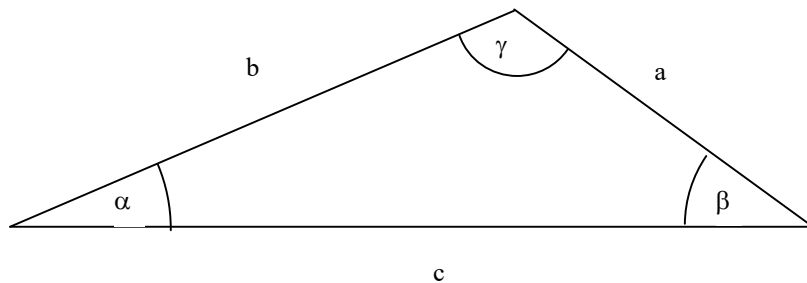
Schreiben Sie ein Programm, das die Winkel in einem Dreieck berechnet.

Sie geben über Tastatur die drei Seitenlängen ein.

Die Seitenlängen müssen die eines "echten" Dreiecks sein, d.h:

alle Seitenlängen müssen größer als 0 sein, das Dreieck darf auf keiner Geraden liegen und die Seitenlängen müssen einem Dreieck entsprechen (z.B. entsprechen die Seitenlängen 1, 2, 1000 keinem Dreieck).

Tipp:



Das Quadrat einer Seite ist gleich der Summe der Quadrate der anderen (anliegenden) Seiten abzüglich dem doppelten Produkt der anderen (anliegenden) Seiten multipliziert mit dem Kosinus des eingeschlossenen Winkels der anderen (anliegenden) Seiten.

Beispiele:

$$a^2 = b^2 + c^2 - 2bc \cdot \cos(\alpha) \implies \cos(\alpha) = \frac{b^2 + c^2 - a^2}{2bc}$$

$$b^2 = a^2 + c^2 - 2ac \cdot \cos(\beta) \implies \cos(\beta) = \frac{a^2 + c^2 - b^2}{2ac}$$

$$c^2 = a^2 + b^2 - 2ab \cdot \cos(\gamma) \implies \cos(\gamma) = \frac{a^2 + b^2 - c^2}{2ab}$$

20) Bußgeldkatalog

Schreiben Sie ein Programm (Bussgeldkatalog), das für das unten angezeigte Vergehen die entsprechenden Sanktionen auf dem Bildschirm ausgibt.

Ein Ausschnitt aus dem Bußgeldkatalog

Innerhalb geschlossener Ortschaften:

bis 10 km/h 15,- EUR
11-15 km/h 25,- EUR
16-20 km/h 35,- EUR
21-25 km/h 50,- EUR, 1 Punkt
26-30 km/h 60,- EUR, 3 Punkte
31-40 km/h 100,- EUR, 3 Punkte, 1 Monat Fahrverbot
41-50 km/h 125,- EUR, 4 Punkte, 1 Monat Fahrverbot
51-60 km/h 175,- EUR, 4 Punkte, 2 Monate Fahrverbot
61-70 km/h 300,- EUR, 4 Punkte, 3 Monate Fahrverbot
über 70 km/h 425,- EUR, 4 Punkte, 3 Monate Fahrverbot

Außerhalb geschlossener Ortschaften (z.B. Landstraße, Autobahn):

bis 10 km/h 10,- EUR
11-15 km/h 20,- EUR
16-20 km/h 30,- EUR
21-25 km/h 40,- EUR, 1 Punkt
26-30 km/h 50,- EUR, 3 Punkte
31-40 km/h 75,- EUR, 3 Punkte
41-50 km/h 100,- EUR, 3 Punkte, 1 Monat Fahrverbot
51-60 km/h 150,- EUR, 4 Punkte, 1 Monat Fahrverbot
61-70 km/h 275,- EUR, 4 Punkte, 2 Monate Fahrverbot
über 70 km/h 375,- EUR, 4 Punkte, 3 Monate Fahrverbot

21) Ersatzwiderstand

1) Erstellen Sie ein Struktogramm zu dem Programm, das den Ersatzwiderstand von **maximal** drei parallel geschalteten Widerständen berechnet. Die Widerstände (Einheit: Ohm) müssen über Tastatur eingegeben werden. Wenn für einen Widerstand ein Wert kleiner oder gleich 0 eingegeben wird, wird der Ersatzwiderstand der bisher eingegebenen Widerstände berechnet und ausgegeben (bzw. eine entsprechende Meldung, wenn kein Ersatzwiderstand existiert). Danach wird das Programm **beendet** (ohne auf eine weitere Eingabe zu warten). D.h. es kann kein Ersatzwiderstand bzw. der Ersatzwiderstand von 1, 2 oder 3 parallelen Widerständen berechnet werden.

22) Taschenrechner

Es soll ein Taschenrechner (er soll die 4 Grundrechenarten beherrschen) programmiert werden.

- a/A: Berechnet die Summe
- b/B: Berechnet die Differenz
- c/C: Berechnet das Produkt
- d/D: Berechnet den Quotient
- sonst: Programmende

Dieser soll auf 2 Arten (der Anwender kann dies auswählen) betrieben werden können: als Erwachsenentaschenrechner oder als Kindertaschenrechner. Der Kindertaschenrechner soll bei Eingabe einer negativen Zahl bzw. eines negativen Ergebnisses sofort eine entsprechende Meldung bringen und dann das Programm sofort beenden.

Außerdem soll der Taschenrechner (egal ob als Erwachsenentaschenrechner oder als Kindertaschenrechner genutzt) bei Division durch Null sofort eine entsprechende Meldung bringen und dann das Programm sofort (ohne Rechnung) beenden.

23) Mengenlehre

Schreiben Sie ein Programm, das die Schnittmenge, die Vereinigungsmenge und die Differenzmenge zweier Zahlenmengen A und B berechnet und angibt ob A von B Teilmenge ist bzw. umgekehrt.

Über Tastatur werden die n ($n = 2$) Elemente der Menge A eingegeben. Dann werden über Tastatur die m ($m = 2$) Elemente der Menge B eingegeben.

Es soll berechnet und dann ausgegeben werden:

- a) $A \cap B$
- b) $A \cup B$
- c) $A \setminus B$
- d) Ist A Teilmenge von B ?
- e) Ist B Teilmenge von A ?

Bemerkungen:

b1) Die Elemente der Menge A müssen jeweils verschieden sein. Das Programm muß dies abprüfen. Die Elemente der Menge B müssen verschieden jeweils sein. Das Programm muß dies abprüfen.

b2) Lösen Sie die Aufgabe für andere Werte von n bzw. m.

24) Wochentag bestimmen

Implementieren Sie ein Programm, das für einen Tag (1..31), den Monat (1..12) und das Jahr den zugehörigen Wochentag bestimmt und auf dem Bildschirm ausgibt.

Folgendes Verfahren berechnet diesen Wochentag für ein Datum, das zwischen März 1900 und Februar 2100 liegt.

Zuerst muss die Gesamtanzahl der Wochentage von 1900 bis zum gegebenen Datum errechnet werden (s.u.). Dies ist die Hauptschwierigkeit. Wenn diese Zahl berechnet ist, muß sie nur noch mit Rest durch 7 geteilt werden. Das ergibt den Wochentag, wobei 0 der Sonntag ist.

Die Gesamtanzahl Wochentage berechnet sich schrittweise wie folgt:

Vom gegebenen Jahr 1900 abziehen und dies mit 365 multiplizieren.

Die fehlenden Schalttage hinzuaddieren $(\text{jahr} - 1900) / 4$. Wenn das Jahr selbst ein Schaltjahr ist, dann ist für Januar und Februar ein Tag zuviel berechnet. In diesem Fall Eins abziehen.

Pro Monat die entsprechende Anzahl Tage pro Monat hinzuaddieren (beim Februar immer 28 Tage, da das Schaltjahr schon berücksichtigt ist). Noch den Tag dazu.

Beispiele:

Ostersonntag: 23. April 1916, 8. April 2007, 4. April 2010

Aschermittwoch: 1. März 2006, 21. Februar 2007, 17. Februar 2010

Gründungstag der BRD (Montag): 23. Mai 1949

25) Schnittpunkt zweier Geraden (Achtung: viele Fallunterscheidungen!)

Zuerst müssen 2 verschiedene Punkte eingegeben werden (bzw. die zugehörigen Koordinaten). Diese legen eindeutig eine Gerade g_1 fest.

Dann müssen wieder 2 verschiedene Punkte eingegeben werden (bzw. die zugehörigen Koordinaten). Diese legen wieder eindeutig eine Gerade g_2 fest.

Bestimmen Sie die Schnittpunkte dieser Geraden g_1 und g_2 .

26) Innerer Punkt eines Dreiecks (schwierig)

$A(x_A | y_A)$, $B(x_B | y_B)$, $C(x_C | y_C)$ sind die 3 Eckpunkte eines Dreiecks in einem zweidimensionalen Koordinatensystem. Ein Punkt P heißt innerer Punkt des Dreiecks ABC, wenn er sich innerhalb des Dreiecks befindet (P darf auch nicht auf den Seiten des Dreiecks liegen).

a) Geben Sie ein Kriterium für einen inneren Punkt an.