

Mohamad Algoabra, Matrikelnummer : 219204154

Bashar Almukdad, Matrikelnummer : 219204329

1. Erläutern Sie die Regeln für erlaubte Variablennamen in C. Der Name foobar ist erlaubt, der Name foo&bar dagegen nicht – warum?

Die Regeln für Nennen die Variablen in C:

- Der Name muss mit einem Gross- oder Kleinbuchstaben oder mit einem Unterstrich (_) beginnen.
- Der name kann auch Zahlen beeinhaltend allerdings nicht am Anfang.
- Darüber hinaus Groß- und Kleinschreibung haben unterschiedliche Bedeutung. Also sind die Variablen name , Name und NAME gelten als unterschiedliche Variablen .
- Außerdem gibt es in C eine Liste sogenannter *Schlüsselwörter*. Diese Wörter dürfen nicht von Variablen benutzt werden. Wie zum Beispiel (if ,do ,while).
- Leerzeichen is verboten in den Namen ,und Der Name darf keine Symbole enthalten (\$\$\$& ...).

Daher ist die Variable **foo&bar** falsch, weil sie ein Symbol enthält, das eine andere Bedeutung haben kann.

2. Könnte es sein, dass sehr lange Variablennamen in C aus bestimmten Gründen problematisch sind? Was sagt der ANSI-Standard bzw. das Kernighan-Ritchie Buch hierzu?

- Die Länge der Variablennamen können ja in vielen Fällen problematisch sein Beispielsweise, wenn der Programmierer sie falsch schreibt ,taucht aus diesem Grund später Fehler auf ,es wäre also besser ,wenn wir unsere Variablen nicht so lange Namen zu geben
- Und zweitens ist die Compiler selbst , die Compiler stellt oft Begrenzungen für die Namen der Variablen. Bei Variablen werden nur die ersten 32 Zeichen zur Unterscheidung gelesen werden. Das heisst, dass man Variablennamen, die länger als 32 Zeichen sind, vermeiden sollte, da sie zu Verwechslungen führen können.

3. Schreiben Sie ein Programm, das Beträge von Euro in US Dollar umrechnet. Der aktuelle Währungskurs ist: • 1 Euro entspricht 1.10 USD Überlegen Sie sich, mit welcher Formel Sie einen Betrag von Euro in USD umrechnen können. Ihr Programm soll so lange Beträge einlesen, bis der Benutzer eine 0 eingegeben hat – dann soll das Programm beendet werden. Wenn der Benutzer einen anderen Wert eingibt, dann soll die Umrechnung in USD und die Ausgabe des Wertes erfolgen.

```
#include <stdio.h>

int main(){
    float e,u;
    printf("Bitte geben Sie einen Betrag in Euro ein (Abbruch mit 0)\n");
    scanf("%f", &e);
    while(e != 0) {
        u = e*1.10;
        printf("Der Betrag entspricht in USD: %f\n",u);
        printf("Bitte geben Sie einen Betrag in Euro ein (Abbruch mit 0)\n");
        scanf("%f", &e);
    }
    return 0;
}
```

4. Schreiben Sie ein Programm, das eine positive ganze Zahl in das binäre Zahlensystem übersetzt. Dabei dürfen Sie die Binärstellen in umgedrehter Reihenfolge ausgeben (d.h., niederwertigste Stelle zuerst), so dass Sie etwa für die Zahl 1310 die Ausgabe 10112 erhalten (in normaler Reihung wäre die Ausgabe 11012).

```
#include <stdio.h>

int main(){
    int dec, bin, x=2;
    printf("Bitte geben Sie eine positive ganze Zahl:");
    scanf("%d",&dec);
    if(dec == 0){
        printf("0");
    }
    else{
        while (dec != 0) {
            if(dec%x == 0){
                bin=0;
            }
            else{
                bin=1;
                dec = dec -(dec%x);
            }
            printf("%d", bin);
            x = x*2;
        }
    }
    return 0;
}
```

5.

- **Erläutern Sie, was das Programm macht.**

Das Programm konvertiert die Zahl vom Dezimalsystem in ein beliebiges vom Benutzer gewähltes Zahlensystem zwischen 2 und 10 d.h. alle Zahlensysteme zwischen dem Binärsystem und Dezimalsystem. wobei W ist das Zahlensystem, in dem der Benutzer die Zahl konvertieren möchte (z.B. 2 für das Binärsystem, 8 für das Oktalsystem...usw), und X ist die Zahl in Dezimalzahl.

Das Programm ist aber nicht vollständig, so dass es einige Fehler gibt, zuerst im Fall der Einführung von 0 oder 1 in der Variable "w", das Programm wird in eine Endlosschleife treten, weil die 0 und 1 den Wert fest macht und das zur Endlosschleife führt → wegen dieser Schleife (**$\text{while}(y \leq x) y = y * w;$**).

Es gibt auch einen Sonderfall, wenn $w = 1$ und $x = 0$ eine unendliche Reihe von Nullen erzeugen, weil die Bedingung in der Schleife (**$\text{while}(y > 0)$**) immer wahr wird. Außerdem funktioniert das Programm mit negativen Zahlen nicht.

des Weiteren die Zahlensysteme mit Basis größer als 10 haben keine speziellen Zahlen oder Symbol für Zahlen zwischen derselben Basis und zehn, was zu einem Konvertierungsfehler führt.

- **Erläutern Sie, wie es das Programm macht – d. h., welchen Algorithmus das Programm verwendet, um seine Aufgabe zu erfüllen.**

Der Algorithmus konvertiert die Zahlen vom Dezimalsystem in andere Systeme, die lautet: x sei eine Zahl im Dezimalsystem und w die Basis, in dessen Zahlensystem die Zahl x konvertiert wird.

- Sei x die angegebene Dezimalzahl.
- Sei w die Basis der Zahl, in die man konvertieren möchte.
- Sei z die Zahl, die anfangs leer ist, und in die konvertiert wird.
- Sei y eine Zahl, die mit einem Wert von 1 beginnt.
- Die Anweisung ($y = y * w$) lässt sich wiederholen, bis y größer als x wird.
- Danach wird der letzte Wert von y durch w geteilt, und als neuer Wert von y abgespeichert.
- Wiederholen, bis y kleiner oder gleich 0 wird.

Teilen x durch y und lassen Sie das Ergebnis z sein
Dann das Ergebnis von z erscheint auf dem Bildschirm und subtrahiert $y * z$ von der eingegebenen Dezimalzahl x , und als neuer Wert von y abgespeichert.
Danach wird der letzte Wert von y durch w geteilt, und als neuer Wert von y abgespeichert.

6. In Aufgabe 6 von Aufgabenblatt A01 sollten Sie ein Struktogramm für einen Algorithmus zur Berechnung der Sinusfunktion auf Basis einer Reihenentwicklung angeben. Nun ist Ihre Aufgabe, diesen Algorithmus zu implementieren. Schreiben Sie ein Programm, das einen Winkelwert von der Standardeingabe in eine double-Variable einliest, den zugehörigen Sinuswert als double-Zahl mit Hilfe Ihres Algorithmus berechnet, und anschließend ausgibt. Anmerkungen:

- Die Reihenentwicklung erwartet, dass der Winkel in Bogenmaß angegeben wird.
- Für das Einlesen eines Wertes in eine double-Variable x können sie den Aufruf `scanf("%lf",&x);` verwenden.
- Wenn Sie ihren Algorithmus geschickt formulieren, benötigen Sie für die Reihenentwicklung lediglich eine einzige Schleife.

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

```
int main(){
    double i, n, x, sum, v;
    printf("Schreiben Sie der Wert vom Winkel: ");
    scanf ("%lf",&x);
    x=x*3.14159265358/180;
    v=x;
    sum=x;
    n = 10 ;

    for(i=1;i<=n;i++)
    {
        v=(v*(-1)*x*x)/(2*i*(2*i+1));
        sum=sum+v;
    }

    printf(" Sin(%f) = %.9lf",x,sum);

    return 0;
}
```