



## 2. Aufgabe: Grundlagen

Schreiben Sie ein Hauptprogramm, das eine Schleife enthält und so lange reelle Werte in die Variable  $x$  von der Konsole einliest, bis der Wert 0 eingegeben wird. Bestimmen Sie innerhalb der Schleife mit diesem Wert den Funktionswert der Funktion

$$f(x) = x^3 - 5/(x + 4) + 3$$

und geben Sie diesen reellen Wert auf die Konsole aus. Schreiben Sie das Programm so, dass für den Funktionswert immer definierte Werte berechnet werden (Hinweis:  $25 / 0$  ist z.B. nicht definiert) – Beachten Sie: Außer der Ein-/Ausgabe-Standardbibliothek darf keine Bibliothek verwendet werden. (20P)

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

int main()
{
    double dValue = 0.0;
    do
    {
        cout << "Wert: ";
        cin >> dValue;

        if(dValue != 0.0 && dValue != -4.0)
        {
            double dResult = dValue * dValue * dValue +
                5.0 / (dValue - 4.0) + 3;
            cout << "Ergebnis f(" << dValue << ") = " << dResult << endl;
        }
    }
    while(dValue != 0.0);

    return 0;
}
```

### 3. Aufgabe: Variablen

a) Was versteht man unter einer Variablendeklaration im Unterschied zu einer Variablendefinition? (4 P)

Eine Variablendefinition ist eine Variablendeklaration, bei der die Variable einen Anfangswert erhält. Man spricht hier von einer Initialisierung der Variablen.

b) Bestimmen Sie den Wert der nachfolgenden Variablen: (6 P)

- int x = 47 / 2; x == 23

- double d = 3.1415926536;  
int x = d; x == 3

- double d = 12.67 ;  
double x = (int)((d \* 10.0) + 0.5) / 10.0); x == 12.0

### 4. Aufgabe: Funktionen

a) Schreiben Sie eine Funktion, die zwei reelle Parameter a und b besitzt und in der die math. Formel  $a + 2b + ab$  berechnet wird. Geben Sie das Ergebnis dieser Berechnung als Funktionswert zurück. (5 P)

```
double f(double a, double b)
{
    return a + 2 * b + a * b;
}
```

b) Wie nennt man die oben verwendete Parameterübergabe, wenn nur der Wert des Parameters verwendet wird? (1 P):

call by value

c) Wie nennt man eine Parameterübergabe, wenn die Variable selbst übergeben wird? (1 P):

call by reference

4 d) Schreiben Sie eine kleine Funktion, bei der ein Polynom  $a$  vom Grad 3 in der Form eines Arrays übergeben und der Wert an der Stelle  $x$  berechnet wird ( $a_0 + a_1 * x^1 + a_2 * x^2 + a_3 * x^3$ ). Übergeben Sie der Funktion das Polynom als Array, sowie den Wert für  $x$ . Das Ergebnis soll als reelle Zahl in Form eines Funktionswertes zurück gegeben werden. Geben Sie anschließend ein Programmfragment an, bei dem die von Ihnen definierte Funktion aufgerufen wird (d.h. eine Anweisung reicht und bitte kein Hauptprogramm oder irgendwelche Ausgaben schreiben!). Bibliotheken dürfen nicht verwendet werden! (8 P)

```
double polynom(double* a, double x)
{
    return a[0] + a[1] * x + a[2] * x * x + a[3] * x * x * x;
}

double    adPolynom[] = {2.5, 3.7, 8.2, -1.2};

polynom(adPolynom, 5.0;
```

e) Aus welchen Teilen besteht eine Funktion und wie ist sie aufgebaut bzw. gibt es Besonderheiten? (10 P)

Funktionskopf, Funktionsrumpf

Funktionskopf ist aufgebaut aus:

1. Datentyp des Rückgabewertes
  - a. Hat eine Funktion keinen Rückgabewert, wird als Datentyp void angegeben.
2. Name der Funktion
3. Parameterliste eingeschlossen in runden Klammern
  - a. eine Parameterliste enthält keinen, einen oder beliebig viele Parameter,
  - b. enthält die Parameterliste mehr als einen Parameter, werden die Parameter durch ein Komma getrennt,
  - c. die Definition eines Parameters entspricht einer Variablendefinition,
  - d. Parameter können wie Variablen verwendet werden.

Schema des Aufbaus eines Funktionskopf:

Datentyp Funktionsname (Parameterliste)

## 5. Aufgabe: Array/Feld, Indizierung, Zeichenketten

a) Gegeben ist eine Zeichenkette `char* acString`. Die Zeichenkette schließt mit einer 0 als Kennzeichnung für das Ende der Zeichenkette. Schreiben Sie eine Funktion, die die Länge einer solchen Zeichenkette bestimmt und als Funktionsergebnis zurück gibt. (10 P)

```
int strlen(char* acInString)
{
    int iLen = 0;
    while(acInString[iLen] != 0)
    {
        iLen++;
    }
    return iLen;
}
```

b) Schreiben Sie eine Funktion `contains`, die zählt, wie oft ein geg. Zeichen in einer Zeichenkette vorhanden ist. Verwenden Sie die unten angegebene Funktion und implementieren Sie im Funktionsrumpf, die gestellte Spezifikation. Kein Hauptprogramm, keine Ein-/Ausgabe! (8)

```
unsigned int contains(char* acString, char cSearchedChar)
{
    unsigned int uiCount = 0;
    int iLen = 0;
    while(acInString[iLen] != 0)
    {
        if(acInString[iLen] == cSearchedChar)
        {
            uiCount++;
        }
        iLen++;
    }

    return uiCount;
}
```



## 6. Aufgabe: Algorithmus

Was berechnen die nachfolgenden Funktionen unknown1 und unknown2?

Bitte beschreiben Sie die Funktionsweise möglichst abstrakt. (15 P)

*Hinweis: Testen Sie den Algorithmus und beobachten Sie die Variablenwerte anhand des geg. Hauptprogramms. Welche Zahlenmenge könnte hier eine Rolle spielen, die in C/C++ nicht als Datentyp existiert?*

```
struct Tripel // Datum
{
    int    miVar1;           // Jahr
    int    miVar2;           // Monat
    int    miVar3;           // Tag
};

bool unknown1(Tripel sInTripel) // true: Schaltjahr, false: kein Schaltjahr
{
    return sInTripel.miVar1 % 4 == 0 &&
        (sInTripel.miVar1 % 100 != 0 || sInTripel.miVar1 % 400 == 0);
}

int unknown2(Tripel sInTripel) // Anzahl vergangener Tage im angegebenen Jahr
{
    int aiVar1[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30};

    int iVar2 = 0;
    for(int li=0; li < sInTripel.miVar2; ++li)
    {
        iVar2 += aiVar1[li];
    }
    iVar2 += sInTripel.miVar3;

    if(sInTripel.miVar2 > 2 && unknown1(sInTripel))
    {
        iVar2++;
    }
    return iVar2;
}

void main()
{
    Tripel oTripel1;
    oTripel1.miVar1 = 2013;
    oTripel1.miVar2 = 2;
    oTripel1.miVar3 = 13; // Datum 13.02.2013

    Tripel oTripel2;
    oTripel2.miVar1 = 2014;
    oTripel2.miVar2 = 4;
    oTripel2.miVar3 = 11; // Datum 11.04.2014

    int i1 = unknown2(oTripel1); // i1 == 44
    int i2 = unknown2(oTripel2); // i2 == 101
}
```

- Bitte beachten Sie auch die Rückseite -
- Lösen Sie die Aufgaben bitte auf dem Blatt -