

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 den Funktionswert der Funktion $y = x^3 + 4/x + 1$ an der Stelle x und geben Sie diesen reellen Wert von y auf die Konsole aus. Schreiben Sie das Programm so, dass für den Funktionswert y immer definierte Werte berechnet werden (Hinweis: $25 / 0$ ist z.B. nicht definiert). (20P)

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

int main()
{
    double x;
    do
    {
        cout << "Wert: ";
        cin >> x;
        if(x != 0.0)
        {
            double y = x * x * x + 4 / x + 1.0;
            cout << "y = " << y << endl;
        }
    }
    while(x != 0.0);

    return 0;
}
```

3. Aufgabe: Funktionen und Schleifen

a) Erklären Sie die **wesentlichen** Unterschiede zwischen den folgenden Funktionsaufrufen in **wenigen** und **kurzen** Sätzen (bei Romanen kann es Punktabzug geben!): (12 P)

1. void functionCall1(int iInValue)
2. void functionCall2(int& riInOutValue)
3. void functionCall3(int* piInOutValue)
4. int functionCall4()

zu 1. Aufruf mittels call by value, d.h. der Wert wird als Kopie übergeben

zu 2. Aufruf mittels call by reference in C++-Form, d.h. die Variable selbst wird übergeben

zu 3. Aufruf mittels call by reference, C-Form, d.h. ein Zeiger auf eine Variable wird übergeben. Mittels Dereferenzierung des Zeigers kann dann der Variablenwert gelesen und/oder geändert werden. Auch Arrays lassen sich in dieser Form übergeben.

zu 4. Die Funktion besitzt keine Parameter, sondern nur einen Rückgabewert vom Typ int.

b) Welche Schleifenarten gibt es in C/C++? Geben Sie kleine Beispiele als Programm**fragmente** (keine Funktionen, kein Hauptprogramm, keine Ein- oder Ausgabe!). (6 P)

for-schleife

```
int x = 0;
for(int li=0; li < 10; ++li)
{
    x += li;
}
```

while-Schleife

```
int x = 0;
while(x < 10)
{
    x += 2;
}
```

do-while-Schleife

```
int x = 0;
do
{
    x += 4;
}
while(x < 10);
```

- Lösen Sie die Aufgaben bitte auf dem Blatt -

4. Aufgabe: Array/Feld, Indizierung

a) Erklären Sie den Unterschied zwischen einem Array und einer Struktur (kein Roman – kann Punktabzug geben!). (4 P)

- ein Array besteht aus eine Anzahl von Variablen/Feldern desselben Datentyps. Der Zugriff auf einzelne Felder erfolgt mit dem Index-Operator [].
- eine Struktur besteht aus mehreren Variablen, die einen unterschiedlichen Datentyp besitzen können. Der Zugriff erfolgt mit dem Punkt-Operator .

b) 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 Ein- oder Ausgaben schreiben!). Bibliotheken dürfen nicht verwendet werden! (8 P)

```
double polynom(double* a, double dInX, int iInSize)
{
    double dResult = *a;
    double x = dInX;
    for(int li=1; li < iInSize; ++li)
    {
        dResult = a[li] * x;
        x *= dInX;
    }
    return dResult;
}
```

```
// Aufruf der Funktion polynom
```

```
double a[4] = {1.5, 2.1, 7.2, 4.2};
double dResult = polynom(a, 2.0, 4);
```

- Lösen Sie die Aufgaben bitte auf dem Blatt -

5. Aufgabe: Zeichenketten

a) Schreiben Sie eine Funktion `strlen`, die die Länge einer mit 0 terminierten Zeichenkette bestimmt. Die Zeichenkette soll als Parameter übergeben werden. Der Funktionswert soll die Länge zurückgeben (kein Hauptprogramm, keine Ein- oder Ausgabe!). (8 P)

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

b) Schreiben Sie eine Funktion `lower`, die eine Zeichenkette `acString` mit Großbuchstaben in Kleinbuchstaben ändert. Alle anderen Zeichen sollen unverändert bleiben. (kein Hauptprogramm, keine Ein- oder Ausgabe!). (15 P)

`lower("THIS IS AN EXAMPLE!")` → "this is an example!"

Hinweis: `char cOffset = 'a' - 'A';`

```
void lower(char* acInString)
{
    char cOffset = 'a' - 'A';
    int iLen = 0;
    while(acInString[iLen] != 0)
    {
        char cCurrent = acInString[iLen];
        if(cCurrent >= 'A' && cCurrent <= 'Z')
        {
            acInString[iLen] = cCurrent + cOffset;
        }
        iLen++;
    }
}
```

6. Aufgabe: Algorithmus

Was macht die nachfolgende Funktion unknown? Was wird auf die Konsole ausgegeben? (newstr dient nur dem Anlegen einer beschreibbaren Zeichenkette)

Bitte beschreiben Sie die Funktionsweise möglichst abstrakt – Romane geben Abzug! (15 P)

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

char* newstr(char* acInString)
{
    char* acValue = new char[1000];
    int iValue = 0;
    while(acInString[iValue] != 0 && iValue < 999)
    {
        acValue[iValue] = acInString[iValue];
        iValue++;
    }
    acValue[iValue] = 0;
    return acValue;
}

int unknown(char* acInParam1, char acInParam2, char acInParam3, int& riOutValue)
{
    riOutValue = 0;
    int iValue = 0;
    while(acInParam1[riOutValue] != 0)
    {
        if(acInParam1[riOutValue] == acInParam2)
        {
            acInParam1[riOutValue] = acInParam3;
            iValue++;
        }
        riOutValue++;
    }

    return iValue;
}

int main()
{
    int iValue;
    char* acValue = newstr("Was tut dieses Programm?");

    int iResult = unknown(acValue, ' ', '_', iValue);

    cout << "Ergebnis: " << acValue << ", " << iResult << "/" << iValue << endl;
    delete [] acValue;

    return 0;
}
```

Die Funktion unknown ersetzt in der übergebenen Zeichenkette alle Zeichen, die gleich dem acInParam2 sind durch das Zeichen in acInParam3. Der Parameter riOutValue enthält nach dem Aufruf die Länge der Zeichenkette. Der Rückgabewert gibt an, wieviele Zeichen ersetzt wurden.

- Lösen Sie die Aufgaben bitte auf dem Blatt -