

Aufgabe 1 (Fehlersuche).**(6 Punkte)**

Die folgende Funktion soll in einer Liste von Zahlen jede Stelle duplizieren.

D.h. aus einer Liste [1, 2, 3, 4, 5] wird z.B. [1, 1, 2, 2, 3, 3, 4, 4, 5, 5].

Das Programm funktioniert nicht richtig. Erläutern Sie den/die Fehler und eine Lösungsmöglichkeit.

```
void verdoppele(vector<int> & liste)
{
    for (int i=0; i<liste.size(); i++)
    {
        liste.push_back(liste[list.size()-1]);
        for (int j=liste.size()-1; j>i+1; j--)
        {
            liste[j] = liste[j-1];
        }
        liste[i+1] = liste[i];
    }
}
```

Lösung:

Das Programm läuft in eine Endlosschleife, weil sich in jedem Schleifendurchlauf die Länge der Liste erhöht.

D.h. man müsste die Länge der Liste vorab in einer Variable speichern. Außerdem muss man anschließend die Schrittweite korrigieren, damit das gerade duplizierte Element nicht noch ein weiteres Mal dupliziert wird. Zuletzt muss die Schleife doppelt so lange laufen.

```
void verdoppele(vector<int> & liste)
{
    int s = liste.size();
    for (int i=0; i<2*s; i+=2)
    {
        liste.push_back(liste[list.size()-1]);
        for (int j=liste.size()-1; j>i+1; j--)
        {
            liste[j] = liste[j-1];
        }
        liste[i+1] = liste[i];
    }
}
```

Aufgabe 2 (Wert- und Referenzparameter).**(4 Punkte)**

Geben Sie die Ausgabe des folgenden Programms an:

```
void bar(int x) {
    x--;
}

void foo(vector<int> & x) {
    for (int i=0; i<3; i++) x[i]++;
    for (int i=0; i<3; i++) bar(x[i]);
}

int main() {
    vector<int> liste = { 2,3,4 };
    for (int i=0; i<3; i++) cout << liste[i] << " ";
    cout << endl;
    foo(liste);
    for (int i=0; i<3; i++) cout << liste[i] << " ";
    cout << endl;
    return 0;
}
```

Lösung:

Die Ausgabe ist 2 3 4 3 4 5

Aufgabe 3 (Rekursion).**(5 Punkte)**

Erklären Sie, was die folgende Funktion berechnet:

```
int bar(int x, int y) {  
    if (y<0) return -bar(x,-y);  
    if (0 <= x && x < y) return x;  
    if (x < 0) return bar(x+y,y);  
    if (x >= y) return bar(x-y,y);  
}
```

Lösung:

Die Funktion berechnet x modulo y .

Aufgabe 4 (Programmverständnis).

(6 Punkte)

Geben Sie die Ausgabe des folgenden Programms an:

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

int bar(int s) {
    return s + 2;
}

void foo(char * s) {
    int i=2;
    for (i = *s; s[0] != 'b'; i++) {
        s[i] = 103;
    }
    *s = bar(i);
}

int main() {
    char c = 'b';
    foo(&c);
    cout << c << endl;
}
```

Geben Sie dabei für jede Zuweisung an und für die Funktionsköpfe an, welche Variable welchen Wert bekommt.

Lösung:

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

int bar(int s) {           // s: Ascii-Code von 'b'
    return s + 2;          // return Ascii-Code von 'd'
}

void foo(char * s) {       // s: Pointer auf main::c
    int i=2;               // i: 2
    for (i = *s; s[0] != 'b'; i++) {
        s[i] = 103;
    }
    *s = bar(i);           // main::c: Ascii-Code von 'd'
}

int main() {
    char c = 'b';          // c: Ascii-Code von 'b'
    foo(&c);
    cout << c << endl;
}
```

}

Aufgabe 5 (Datentypen).**(3 Punkte)**

Definieren Sie einen Datentyp **Rechnung**, der es ermöglicht, Datensätze zu Rechnungen mit Name, Kunden- und Rechnungsnummer, Betrag und Datum zu speichern.

Lösung:

```
struct Rechnung {  
    string name;  
    string knr;  
    string rnr;  
    int betrag;  
    string datum;  
};
```