

Übungsaufgaben zur Klausurvorbereitung

Aufgabe 1

Geben Sie das Ergebnis der folgenden Operationen an:

```
1 int k;  
2 double x = 4.2, y = 1.9, z;  
3 char ch1 = 12, ch2 = 7;  
4 int array[12];  
5 int *ptr1;  
6 int *ptr2;
```

Der in ptr2 gespeicherte Wert ist 0xB000.

Der in ptr1 gespeicherte Wert ist 0xB010.

- | | |
|--------------------------|--------|
| a) k = x + y; | k = |
| b) z = (int)x + (int)y; | z = |
| c) k = (int)(x + y); | k = |
| d) k = sizeof(array[0]); | k = |
| e) k = sizeof(array); | k = |
| f) ch1 = ch1 >> 1; | ch1 = |
| g) ch2 = ch2 << 3; | ch2 = |
| h) ch2 = ch2 - ch1; | ch2 = |
| i) k = ptr2 - ptr1; | k = |
| j) ptr1++; | ptr1 = |

Aufgabe 2

Geben Sie den Inhalt der Variablen p, q, r, s, t und des Arrays data nach Ausführung der Funktion func an:

```
1 int func(int x, int &y, int z, int *py, double *pdata) {  
2     x += 5;  
3     y *= 3;  
4     z -= 2;  
5     *py += 4;  
6     *(pdata + 1) += 3.5;  
7     return (z + 5);  
8 }  
9  
10 int main() {  
11     int p = 15, q = 16, r = 17, s = 18, t = 19;  
12     double data[] = {2.0, 3.0, 4.0, 5.0, 6.0};  
13     p = func(q, r, s, &t, data);
```

```

14     q = p;
15     r = s;
16     s = t;
17     t = (int) data[2];
18     data[0] = s;
19     data[1] = t;
20     data[2] = data[4];
21     data[3] = data[0];
22     return 0;
23 }

```

Aufgabe 3

Gegeben ist ein Integer-Array.

- a) Schreiben Sie eine allgemeingültige Funktion, die in einem Integer-Array nach einem bestimmten Wert sucht. Ist der Wert vorhanden, soll der entsprechende Index über den Rückgabewert zurückgegeben werden, anderenfalls soll -1 zurückgegeben werden. Prototyp der Funktion:

```
1 int find(int* pointer, int size, int value);
```

- b) Binden Sie die Funktion durch Deklaration in ein übergeordnetes Modul (`main()`) ein und rufen Sie die Funktion mit einem Beispielwert auf, z. B. Array {5, 15, 25, 35, 45} und Wert 25.

Aufgabe 4

- a) Schreiben Sie eine Funktion, die ein dynamisch allokiertes Integer-Array mit einer gegebenen Grösse erstellt und mit Zufallswerten zwischen 1 und 100 füllt. Der Prototyp der Funktion lautet:

```
1 int* createRandomArray(int size);
```

Hinweis: Verwenden Sie `rand()` für Zufallswerte (angenommen, `srand(time(NULL))` wurde aufgerufen).

- b) Binden Sie die Funktion in ein übergeordnetes Modul (`main()`) ein. Allokieren Sie ein Array mit 50 Elementen, rufen Sie die Funktion auf, geben Sie die ersten 5 Werte aus und geben Sie den Speicher anschließend frei.

Hinweis: Programmieren in ANSI-C oder C++ bleibt Ihnen überlassen.

Aufgabe 5

Deklariieren Sie eine Klasse `Vector` für Vektoren in der Ebene (u, v) . Definieren Sie innerhalb der Klasse:

- den Standardkonstruktor,
- den Initialisierungskonstruktor,

- den Destruktor.

Definieren Sie außerhalb der Klasse folgende Memberfunktionen:

- Addition zweier Vektoren (`add`),
- Operatorfunktion zur Verwendung des `+`-Operators zum Addieren zweier Vektoren.

Erzeugen Sie im Hauptprogramm mehrere Objekte vom Typ `Vector`, die die implementierten Konstruktoren nutzen, und rufen Sie alle Memberfunktionen auf. Die Operatorfunktion rufen Sie einmal mit dem Operator und einmal mit der vollständigen Notation auf.