

Programmieren in C, Testat 3

1. Das folgende Programm kann ohne Fehler kompiliert werden:

```
1  #include <stdio.h>
2  int main(void){
3      int *x;
4      *x = 100;
5      printf("\n *x = %d \n &x = %p \n sizeof(x) = %lu \n",
6             *x, (void*)x, sizeof(x));
7      return 0;
8  }
```

Was passiert, wenn Sie es ausführen. Erklären Sie die Fehlermeldung und korrigieren Sie das Programm, indem Sie höchstens zwei neue Programmzeilen einfügen. Erklären Sie dann die Programmausgabe im Detail.

2. Ein Programmieranfänger hat das nachfolgende Programm geschrieben. Nach seinen eingefügten Kommentaren zu urteilen wollte er aber wohl eine andere Logik implementieren. Finden Sie seinen logischen Fehler und beheben Sie ihn.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(void){
4      double a[3]={1.1,2.2,3.3}; // Koordinaten eines 3D-Punktes
5      double *b = (double*)malloc(3*sizeof(double)); // Speicher für einen 3D-Punkt
6      b = a; // Kopiere Punkt a nach b !
7      b[2] = 0; // Projiziere Punkt b in die (x,y)-Ebene !
8      printf("\n b = (%.21f,%.21f,%.21f) ",b[0],b[1],b[2]);
9      printf("\n a = (%.21f,%.21f,%.21f) \n",a[0],a[1],a[2]);
10     return 0;
11 }
```

English translation

1. The following program can be compiled without errors and warnings:

```
1  #include <stdio.h>
2  int main(void){
3      int *x;
4      *x = 100;
5      printf("\n *x = %d \n &x = %p \n sizeof(x) = %lu \n",
6             *x, (void*)x, sizeof(x));
7      return 0;
8  }
```

What happens when you execute it. Explain the error message and correct the program by inserting at most two new program lines at appropriate positions. Finally explain the program output in detail.

2. A beginning programmer wrote the following program. According to his comments it seems that he wanted to implement something else. Find and explain his logical error and correct the program accordingly.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(void){
4      double a[3]={1.1,2.2,3.3}; // coordinates of a point a in 3D
5      double *b = (double*)malloc(3*sizeof(double)); // memory for a 3D point b
6      b = a; // copy point a to b
7      b[2] = 0; // project b into the (x,y)-plane
8      printf("\n b = (%.2lf,%.2lf,%.2lf) ",b[0],b[1],b[2]);
9      printf("\n a = (%.2lf,%.2lf,%.2lf) \n",a[0],a[1],a[2]);
10     return 0;
11 }
```