

Aufgabe 1: Fehler erkennen

Dies sind Multiple Choice Aufgaben. Es kann mehrere richtige Antworten geben. Falsch gesetzte Kreuze geben Punktabzug, jedoch kann eine Frage keine negative Anzahl Punkte verursachen. Voll ausgemahlte Kästchen gelten als nicht angekreuzt, wenn sich daneben ein leeres neu gemaltes Kästchen befindet.

Aufgabe 1.1: Frage 1

Was führt zu einem Speicherzugriff auf fremde Speicherbereiche?

- Der Zugriff auf den Index 5 eines 5 elementigen Arrays.
- Die fehlende Initialisierung eines Chars.
- Rechenfehler bei der Pointerarithmetik

Aufgabe 1.2: Frage 2

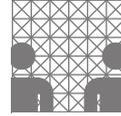
Welchen Wert haben uninitialisierte Variablen?

- Sie sind immer mit 0 vorinitialisiert.
- Der Compiler verlangt, dass Variablen initialisiert werden.
- Der Wert den Variablen vor der Initialisierung haben ist nicht eindeutig festgelegt.

Aufgabe 1.3: Frage 3

Welche Aktionen bei der dynamischen Speicherverwaltung führen zu undefiniertem Verhalten?

- Die Anwendung von `free()` auf einen Nullpointer.
- Die doppelte Freigabe eines Pointers.
- Die Nutzung eines Pointers nach seiner Freigabe.



Aufgabe 2: Fehler beheben

Markieren Sie in folgendem Quelltext alle vorkommenden Speicherfehler und geben Sie einen Lösungsweg an, wie man diesen Speicherfehler beheben kann.

```
1
2 #include <stdio.h>
3
4 int main(int argc, char *argv[])
5 {
6     int wortanzahl = 0;
7     int wortlaenge1 = 0;
8     int wortlaenge2 = 0;
9     char *string1=NULL;
10    char *string2=NULL;
11
12    printf("Bitte gib an, ob du ein oder zwei Worte schreiben moechtest:\t");
13    scanf("%i", &wortanzahl);
14    if((wortanzahl < 1)|| (wortanzahl > 2))
15    {
16        printf("Fehler: du sollst 1 oder 2 eingeben\n");
17        return 1;
18    }
19    printf("Bitte gib an, wie lang dein erstes Wort ist:\t");
20    scanf("%i", &wortlaenge1);
21    string1 = (char *)malloc(sizeof(char)*wortlaenge1);
22    printf("Bitte gib nun dein erstes Wort ein:\t");
23    scanf("%s", string1);
24    if(wortanzahl > 1)
25    {
26        printf("Bitte gib an, wie lang dein zweites Wort ist:\t");
27        scanf("%i", &wortlaenge2);
28        string2 = (char *)malloc(sizeof(char)*wortlaenge2);
29        printf("Bitte gib nun dein erstes Wort ein:\t");
30        scanf("%s", string2);
31        printf("Die Katze sagt %s und %s\n",string1,string2);
32        free(string1);
33        return 0;
34    }
35    printf("Die Katze sagt %s\n",string1);
36    return 0;
37    free(string1);
38
39 }
```