

1. (12 Punkte) **Fehler finden**

Im folgenden Programm, das die Anzahl der ungeraden Zahlen in einem Array sowohl ausgeben als auch als Rückgabewert zurückgeben soll, haben sich sechs Fehler eingeschlichen. Markieren Sie die jeweilige Stelle geeignet und schreiben eine laufende Nummer dazu. Darunter beschreiben Sie kurz, wo der Fehler dabei liegt und wie es richtig lauten müsste (jeweils 1 Punkt pro korrekt beschriebenen Fehler und 1 Punkt für die richtige Variante).

```
1  int countOdd (int arr[], int elems) {  
2      int odd;  
3  
4      for (int i = 0; i <= elems; i++) {  
5          if ((arr[i] % 2) == 0) {  
6              odd + 1;  
7          }  
8      }  
9  
10     printf("%s odd numbers are in the array\n", odd);  
11  
12 }
```

1.) _____

2.) _____

3.) _____

4.) _____

5.) _____

6.) _____

2. (10 Punkte) **Rekursives Array-Maximum**

In dieser Funktion sollen Sie den größten geraden Wert eines Arrays auf rekursive Art ermitteln. Implementieren Sie dazu eine Funktion maxevenrec, die eine Referenz auf ein Array, dessen Länge, sowie einen Startindex als Parameter erhält, und die größte gerade Zahl im Array zurückgibt. Ist keine gerade Zahl vorhanden, so soll 0 zurückgegeben werden.

```
1  unsigned int maxevenrec (unsigned int arr[], int index, int size) {  
2  
3  
4  
5  
6  
7  
8  
9  
10 }
```

3. (5 Punkte) **Ausdrücke und Datentypen**

Betrachten Sie folgendes:

```
1 int a = 10;  
2 double arr[15] = { 0 };
```

Welchen Typ hat der jeweilige Ausdruck? (1 Punkt pro korrektem Typ)

- a) Der Ausdruck `"Hallo Welt!"[0]` _____
- b) Der Ausdruck `10 / a` _____
- c) Der Ausdruck `arr + 1` _____
- d) Der Ausdruck `arr[1]` _____
- e) Der Ausdruck `10 / *arr` _____

4. (5 Punkte) **Ausdrücke und Werte**

Betrachten Sie folgendes:

```
1 int a = 9;  
2 double arr[15] = { 1 };  
3  
4 printf("%p", arr); // Ergibt 1000  
5 printf("%d", sizeof(double)); // Ergibt 8
```

Welchen Wert haben folgende Ausdrücke (1 Punkt pro korrektem Wert):

- a) Der Ausdruck `arr[1] = a / 2` _____
- b) Der Ausdruck `a++` _____
- c) Der Ausdruck `arr + 1` _____
- d) Der Ausdruck `arr[1]` _____
- e) Der Ausdruck `sizeof(arr)` _____

5. (10 Punkte) **Codeanalyse**

Betrachten Sie folgendes Programm:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  char *qux (char foo[], int bar) {
5      for (int i = 0; foo[i]; i++) {
6          foo[i] += bar;
7          if (foo[i] > 'Z')
8              foo[i] -= 26;
9      }
10
11     return foo;
12 }
13
14 int main (void) {
15     char str[] = "WELT";
16     printf("%s\n", str + 1);
17     printf("%c\n", *str - 1);
18     printf("Code: %s\n", qux(str, 5));
19     printf("Orig: %s\n", str);
20     printf("length: %lu\n", strlen(str));
21     printf("sizeof: %lu\n", sizeof(str));
22     printf("Int: %d\n", str[4]);
23 }
```

Was ist die Ausgabe des Programms?