

14 / 20  
13

## Minitest 2

Minh Khue Pham (2579036)

18.05.2021 13:50:07 - 14:10:06

8

### Multiple Choice (8 Punkte)

Beantworten Sie die folgenden Fragen mit **wahr** oder **falsch**. Für jede richtige Antwort bekommen Sie 1 Punkt. Für falsche Antworten bekommen Sie 0 Punkte. Wenn Sie keine Antwort geben bekommen Sie 0.5 Punkte.

Answer the following questions with **true** or **false**. For every correct answer you obtain 1 point. For incorrect answers you get 0 points. If you do not give an answer you receive 0.5 points.

1. Mit *malloc* können neue Behälter angelegt werden.

1. *malloc* can be used to create new storage.

#### Response:

- ☒ wahr / true
- ☐ falsch / false

2. Alle Komponenten eines *structs* haben den gleichen Typ.

2. All components of a *struct* have the same type.

#### Response:

- ☐ wahr / true
- ☒ falsch / false

3. *int foo(int x);* ist eine gültige Funktionsdeklaration in C.

3. *int foo(int x);* is a valid function declaration in C.

#### Response:

- ☒ wahr / true
- ☐ falsch / false

4. Der *sizeof* Operator gibt die Größe eines Typs in Bytes zurück.

4. The *sizeof* operator returns the size of a type in bytes.

**Response:**

- ☒ wahr / true
- ☐ falsch / false

5. Mit *fscanf* können Werte auf der Konsole ausgegeben werden.  
5. *fscanf* can be used to output values to the console.

**Response:**

- ☐ wahr / true
- ☒ falsch / false

6. Der Ausdruck  $x -= 5$  ist eine Kurzschreibweise für  $x = 5 - x$   
6. The expression  $x -= 5$  is a shorthand for  $x = 5 - x$

**Response:**

- ☐ wahr / true
- ☒ falsch / false

7. Jeder Ausdruck kann R-ausgewertet werden.  
7. Every expression has an R-value.

**Response:**

- ☒ wahr / true
- ☐ falsch / false

8. Eine Zeichenkette wird in C durch eine Reihung aus *chars* dargestellt.  
8. A string in C is represented by a *char* array.

**Response:**

- ☒ wahr / true
- ☐ falsch / false

## 2 Funktion Vervollständigen (6 Punkte)

Die unten gezeigte Funktion *filter* erhält eine Reihung *array* der Länge *length*, sowie einen Dateipfad *output* als Argumente. Diese Funktion soll alle positiven Zahlen aus *array* mit einem Leerzeichen voneinander getrennt in eine Datei am Pfad *output* schreiben. Ein Leerzeichen vor der ersten oder hinter der letzten Zahl darf bestehen.

Vervollständigen Sie die Funktion indem Sie die mit Kommentaren versehenen Stellen vervollständigen. Tun Sie dies durch Eintragen der Antwort in das zugehörige Textfeld.

The function below receives an *array* of size *length* as input, as well as a file path *output*. The function shall write all positive numbers of *array* into a file located at the file path *output*. The numbers must be separated by a single space. A space before the first or after the last number is allowed.

Complete the function definition by filling out the gaps indicated by comments. Do this by filling out the corresponding text boxes below.

```

1  #include <stdio.h>
2
3  void filter(int* array, int length, char* output) {
4      FILE* file = fopen(/* TODO: Lücke 1 / Gap 1 */, "w");
5      if (!file) return;
6
7      for (int i = 0; i != length; ++i) {
8          if (array[i] > 0) {
9              fprintf(/* TODO: Lücke 2 / Gap 2 */);
10         }
11     }
12
13     /* Datei schließen / Close file */
14     /* TODO: Lücke 3 / Gap 3 */
15 }

```

- 0 Lücke 1 / Gap 1:  *file*
- 2 Lücke 2 / Gap 2:  *file*
- 0 Lücke 3 / Gap 3:  *void* *fclose(file);*

#### 4 Zeiger (6 Punkte)

Welche Ausgabe liefern die Funktionen *f*, *g* und *h*? (Jeweils 2 Punkte)

What is the output of *f*, *g* and *h*? (2 points each)

```
void f() {  
    int a = 42;  
  
    int* p = malloc(sizeof(int));  
    *p = 5;  
    a = *p;  
    free(p);  
  
    printf("%d", a);  
}
```

```
void g() {  
    int a = 2;  
  
    int* x = &a;  
    int* y = x;  
    *x = 30;  
    *y = 20;  
  
    printf("%d", a);  
}
```

```

void h() {
    int a[2];
    a[0] = 0;
    a[1] = 7;

    int* p = &a[0];
    *(p + 1) = 33;

    printf("%d", a[1]);
}

```

Ausgabe für *f* / Output for *f* : 5 ✓  
 Ausgabe für *g* / Output for *g*: 2 ✓  
 Ausgabe für *h* / Output for *h*: 33 ✓