

# }U Info | 22.11.03

Created on: 03-11-2022 10:47

## Prüfungsaufgabe S2019.3

### 1. Fehler im Code Finden

```
01: void mul_ten(int arr, int N, int mul, int M) {  
02:     int m = 0;  
03:     for ( n == 1; n <= N; ++n) {  
04:         tmp = arr[n];  
05:         if ( tmp % 10) { // we found a multiple of 10  
06:             if ( m <= M ) // the array mul is big enough  
07:                 mul[++m] = tmp  
08:             else  
09:                 return -1  
10:             endif  
11:         }  
12:         return n;  
13:     }
```

### 2. Begriffe erklären

- Besprechen wir nicht in der Übung, ist recht trivial

### 3. Output der Funktion geben

```
int func (int m, int &n) {  
    return (++m) + (n--);  
}
```

Give the **output** of

```
int f = 1, g = 2;  
int k = func(f, g);  
cout << f << g << k << endl;
```

and **briefly explain** your reasoning.

# Lösungen

## 1.

Error	Line	Correction	Error type
1	1	The keyword “void” should be replaced by “int” (the function is supposed to return an integer result)	syntactic
2	1	The “[ ]” are missing after <code>arr</code> and <code>mul</code> (these are arrays)	syntactic
3	3	The variable “n” was not declared (replace “n” by “int n”)	syntactic
4	3	The “==” symbol should be replaced by “=” (assignment, not comparison)	syntactic/semantic
5	3	The <code>for</code> -loop should go from 0 to <code>N-1</code> (not from 1 to <code>N</code> )	semantic
6	4	The variable “tmp” was not declared (replace “tmp” by “int tmp”)	syntactic
7	5	The condition should read “!(tmp % 10)” instead (if <code>tmp</code> is a multiple of 10, the rest of the division by 10 is non-zero, which maps to <code>false</code> and not to <code>true</code> ; note that the parentheses are necessary here as otherwise, the “!” would be carried out before the “%”)	semantic
8	6	The condition “m <= M” should be replaced by “m < M” (the array index goes from 0 to <code>M-1</code> , so that element <code>M</code> is already outside the array; this the error cumulates with the next one, where the code may even try to write an element <code>M+1</code> .)	semantic
9	7	The “++m” will increase <code>m</code> before accessing the element, <i>i.e.</i> will start at element 1, not 0 (should be “m++” instead)	semantic
10	7	The “;” symbol is missing	syntactic
11	9	The “;” symbol is missing	syntactic
12	10	“endif” is not a keyword in C++	syntactic
13	11	A “}” is needed here to close the <code>for</code> -loop	syntactic
14	12	The value of “m” should be returned, not that of “n” (given the previous correction, “n” is actually out of scope at this point, <i>i.e.</i> undefined)	semantic

## 3.

The program will print the final values of `f`, `g`, and `k`, namely 1, 1, and 4. The variable `f` is passed to `func` *by value*, so its value in the calling program is unaffected by what happens to the corresponding local variable called `m` in `func`. As a result, the final value of `f` is still 1. In contrast, the variable `g` is passed *by reference*, so that the corresponding variable `n` in `func` corresponds to the same memory location. As a result, the final value of `g` is also 1, because it is decremented in `func` by the `n--`. Finally, in `func`, the result of `++m` with a *prefix* operator is the value of `m` *after* incrementing (*i.e.* 2), whereas that of `n--` with a *suffix* operator is the value of `n` *before* decrementing (*i.e.* 2). The sum (*i.e.* 4) is returned by `func` and stored into the variable `k`.

## Ex 3

- Fragerunde bzgl C++
- Es ist super wichtig, das ihr diese Task gut versteht. Diese Task umfasst fast alles, was ihr in C++ für die Prüfung können müsst.

- Der Sprung von B auf C, ist nur sehr wenig code. Falls ihr euch also sicher in der Theorie zu der Task fühlt, könnt ihr euch gerne an auch an Task C versuchen.