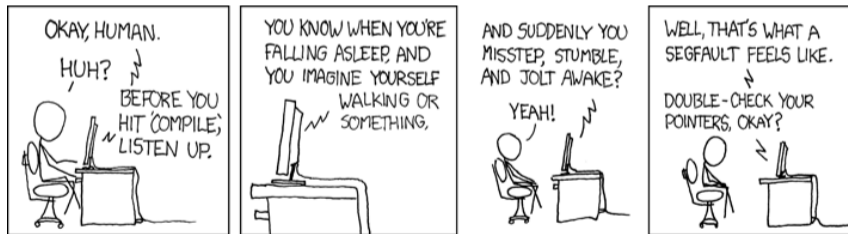


# Informatik für Mathematiker und Physiker

## Exercise Session 13



<https://xkcd.com/371/>

Find mistakes in the following code and suggest fixes:

```
1 // PRE: len is the length of the memory block that starts at array
2 void test1(int* array, int len) {
3     int* fourth = array + 3;
4     if (len > 3) {
5         std::cout << *fourth << std::endl;
6     }
7     for (int* p = array; p != array + len; ++p) {
8         std::cout << *p << std::endl;
9     }
10 }
```

Find mistakes in the following code and suggest fixes:

```
1 // PRE: len is the length of the memory block that starts at array
2 void test1(int* array, int len) {
3     //int* fourth = array + 3;    // ERROR
4     if (len > 3) {
5         int* fourth = array + 3;    // OK
6         std::cout << *fourth << std::endl;
7     }
8     for (int* p = array; p != array + len; ++p) {
9         std::cout << *p << std::endl;
10    }
11 }
```

Even if the pointer is not dereferenced, it must point into a memory block or to the element just after its end.

Find mistakes in the following code and suggest fixes:

```
1 // PRE: len >= 2
2 int* fib(unsigned int len) {
3     int* array = new int[len];
4     array[0] = 0; array[1] = 1;
5     for (int* p = array+2; p < array + len; ++p) {
6         *p = *(p-2) + *(p-1); }
7     return array; }
8 void print(int* array, int len) {
9     for (int* p = array+2; p < array + len; ++p) {
10         std::cout << *p << " ";
11     }
12 }
13 void test2(unsigned int len) {
14     int* array = fib(len);
15     print(array, len);
16 }
```

```
1 // PRE: len >= 2
2 int* fib(unsigned int len) {
3     int* array = new int[len];
4     array[0] = 0; array[1] = 1;
5     for (int* p = array+2; p < array + len; ++p) {
6         *p = *(p-2) + *(p-1); }
7     return array; }
8 void print(int* array, int len) {
9     for (int* p = array+2; p < array + len; ++p) {
10        std::cout << *p << " ";
11    }
12 }
13 void test2(unsigned int len) {
14     int* array = fib(len);
15     print(array, len);
16 } // array is leaked; to fix add: delete [] array
```

Find mistakes in the following code and suggest fixes:

```
1 // PRE: len >= 2
2 int* fib(unsigned int len) {
3     // ...
4 }
5 void print(int* m, int len) {
6     for (int* p = m+2; p < m + len; ++p) {
7         std::cout << *p << " ";
8     }
9     delete m;
10 }
11 void test2(unsigned int len) {
12     int* array = fib(len);
13     print(array, len);
14     delete [] array;
15 }
```

```
1 // PRE: len >= 2
2 int* fib(unsigned int len) {
3     // ...
4 }
5 void print(int* m, int len) {
6     for (int* p = m+2; p < m + len; ++p) {
7         std::cout << *p << " ";
8     }
9     delete m; // should be delete []
10 }
11 void test2(unsigned int len) {
12     int* array = fib(len);
13     print(array, len);
14     delete [] array; // array deallocated twice
15 }
```