

FundProg17 – Assignment 4

4.1 Array search

Given an array of numbers, e.g.,

```
int[] numbers = {1, 3, 99, 4, 5};
```

and a single integer variable `int x = 4;`

use a loop to determine the **position** at which you can find the number in the array. In this example, the output should be 3. If the number is not in the array, the output should be -1 (because we know that there is no position -1). Test whether your code works!

4.2 Array shifting

Write code that “shifts” an array by a given number of steps. For example, given the array

```
String[] words = {"a", "book", "about", "aardvarks", "for",  
                 "the", "student", "by", "Prof.", "Smith"}
```

and the number `int n = 4`, you need to produce a new array looking like this

```
String[] words = {"student", "by", "Prof.", "Smith", "a", "book",  
                 "about", "aardvarks", "for", "the"}
```

As you can see, the 4 last words moved to the beginning of the array.

There are several ways to accomplish this (b) is a little more difficult):

- a) Make a second array of the same length. Copy each word over in the correct order.
- b) The problem can actually be solved with three reversals of parts of the array. Figure out how.

Pick one, write a program, and test with different values!

(Hint: you may need to use the modulo operation here)

4.3 Nested loops

- a) Write a nested loop that prints the numbers from 0 to 99 using two for loops that look like this:

```
for (int i = 0; i < 10; i++) {  
    for (int j = 0; j < 10; j++) {  
        // your solution here!  
    }  
}
```

b) You are given the following arrays:

```
String[] prefixes = {"sleep", "walk", "talk", "fall", "go", "kiss"};
String[] suffixes = {"ing", "ed", "s", "er"};
String[] candidates = {"slept", "walker", "talking", "falls", "goes", "went", "kisses"};
```

Write a program that checks for each word in the `candidates` array whether it can be produced by concatenating a prefix from `prefixes` and a suffix from `suffixes`. Print out the list of words that cannot be produced.

Test whether your code does the right thing.

4.4 Functions

a) Write a function

```
static void sayHi()
```

that prints “Hi!”. Then, in your `main` function, write a loop that prints “Hi” 10 times, **using your `sayHi` function**.

b) Write a function

```
static int absoluteValue(int x)
```

that computes the absolute value (“Betrag”) of `x`. In the `main` function, compute the absolute values of the numbers 11, 5, 0, -24, and -3.34. (Note: There may or may not be a problem here.)

c) Adapt your “array search” program from above so that the search is moved into a function.

So, write a function `search`. The function will take two arguments (`int[] anArray`, `int aNumber`) and print the solution for those. Test the function with three examples.

d) Now modify the function so that the result is no longer printed but instead **returned**. Test the function again.

4.5 Strings

```
String text = "a book about aardvarks for the student by Prof. Smith";
```

a) Use `substring` to get the fourth word

b) Use `split` to get the fourth word

c) Write a loop to print each of the words of the sentence on its own line.

d) Write a loop to print the first letter of each word.