Part 1
Tail recursion

- The goal is to write a function that counts the number of characters passed as program arguments.

- The program should be named **counter** and when called like this:
  > scala counter hello there
  it would print “12”

- The Array of arguments needs to be converted to the List and then the tail-recursively the number of characters in each of the string should be accumulated and returned.

  - If String **size** function is used to get the length 4 points are given for the exercise

  - If the length of each string is obtained by another tail-recursive function the complete 6 points are granted

- The solutions have to be clean, i.e. technique with hiding “buffer” variables need to be used. The approach with external “var" filled while recursing is not accepted.

Part 2
New expression

In this exercise a new “expression” needs to be added. It should be called “**randomly**” and when used in following example:

```
for ( i <- 1 to 10 ) {
  val z = randomly(0.25){
    println(“calling this"); 1
  }{
    println(“calling that"); 2
  }
  println(z);
}
```

it would execute one of the two pieces of code with the probability given in the first argument (here “calling this” would be executes in 1 out of four cases - approximately).

It can be assumed that the return value is always of integer type.

Reminder: 4 points, can use any material