FEEDBACK

Lecture №
07
Review date:
06/05/2018
Checked by:
Горбушко К
Student:
Маурісіо Морочо
Branch:
lesson7
Mark:
6 / 10

EXERCISE 1

factorial < 1 not always equal to 1, only in case 0 and 1, in case numbers below 0 - factorial not exist. Easy fix - use unsigned type

```
func calculateFactorial(of number: Int) -> Int{
func calculateFactorial(of number: UInt) -> Int{
```

my applause for alternative solution

*Actually, interesting to know - preferred solution for factorial calculation - not recursively. With recursive approach much more resources required to do calculation, and so time

EXERCISE 2

```
    //bucle to control the iterations
```

It can be replaced with while cycle instead.

//somewhere inside your func

```
   var notSorted = true
    while(notSorted) {
        notSorted = false
        for i in 0..<array.count - 1 {
            if sortedArr[i] > sortedArr[i+1] {

                //do stuff here
                notSorted = true
            }
```

function with selection sort approach missed

EXERCISE 3

try to use CharacterSet instead

EXERCISE 4

code style - hard to read =/
don’t use deprecated api ('characters' is deprecated)
actually there is easiest way to do this and so much efficient - iterate only once throw string collection

```
var result = [Character : Int]()
for character in inputString {
    if ! result.keys.contains(character) {
        result[character] = 1
    }
    else {
        result[character]! += 1
    }
}
```

EXERCISE 5

ok

EXERCISE 6

we can discuss it later on

Reccomandations

Improve your codeStyle
 cleanUp all unnecessary comments and commented code. If you think I should see it - uncomment it and add comments only if you think its strictly required (for example variant 2 or something like this)

Note: *If you need additional translation of homework task - just let me know