



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 ИУ6-32Б

**О Т Ч Е Т**

по лабораторной работе № 3

**Название:** Основы Golang

**Дисциплина:** Языки интернет-программирования

Студент

ИУ6-32Б

(Группа)

(Подпись, дата)

Кондратов С.Ю.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Шульман В.Д.

(И.О. Фамилия)

# Цель работы: Познакомиться с основами Golang.

## Задание 1

Дана строка, содержащая только арабские цифры. Найти и вывести наибольшую цифру.

**Входные данные**

Вводится строка ненулевой длины. Известно также, что длина строки не превышает 1000 знаков и строка содержит только арабские цифры.

**Выходные данные**

Выведите максимальную цифру, которая встречается во введенной строке.

---

**Sample Input:**  
1112221112

---

**Sample Output:**  
2

Рисунок 1

Очевидно, что нам нужно поочередно пройти по всем символам input и найти самую большую цифру, которая и будет ответом.

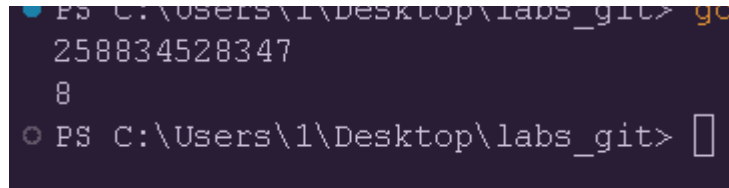
```
Codeium: Refactor | Explain | Generate GoDoc | X
5  func main() {
6      var str string
7      fmt.Scan(&str)
8
9      max := 0
10     for _, digit := range str {
11         d := int(digit) - '0'
12         if d > max {
13             max = d
14         }
15     }
16
17     fmt.Println(max)
18 }
19
```

Рисунок 2

На рисунке 2 показан мой результат.

Интересный факт: для того, чтобы из типа  `rune`, которым является `digit`. Мы можем использовать преобразование, показанное на 11 строке.

Тип `rune` является псевдонимом для `int32`, что означает, что он является 32-битным целым числом, которое может представлять любой код Unicode.



```
PS C:\Users\1\Desktop\labs_git> go
258834528347
8
PS C:\Users\1\Desktop\labs_git> 
```

Рисунок 3

На рисунке 3 показан пример вывода.

## Задание 2

На вход подается целое число. Необходимо возвести в квадрат каждую цифру числа и вывести получившееся число.

Например, у нас есть число 9119. Первая цифра - 9. 9 в квадрате - 81. Дальше 1. Единица в квадрате - 1. В итоге получаем 811181

**Sample Input:**

9119

**Sample Output:**

811181

Рисунок 4

Как и в предыдущем задании мы можем пройти по каждому символу строки, преобразовать его `int` и вывести его квадрат.

```

1  package main
2
3  import (
4      "fmt"
5  )
6
7  Codeium: Refactor | Explain | Generate GoDoc | X
8  func main() {
9      var str string
10     fmt.Scan(&str)
11
12     for _, digit := range str {
13         d := int(digit) - '0'
14         fmt.Printf("%d", d*d)
15     }
16 }

```

Рисунок 5

На рис.5 показан мой результат.

```

PS C:\User
9119
811181

```

Рисунок 6 (Пример вывода)

## Задание 3

## Внимательно прочитайте ВСЕ условия и подсказки чтобы правильно решить задачу!

Требуется вычислить период колебаний ( $t$ ) математического маятника (мы округлили некоторые значения для удобства проверки), для этого нужно найти циклическую частоту колебания пружинного маятника ( $w$ ), в формуле  $w$  встречается масса которую также нужно найти, все нужные формулы приведены ниже:

Напишите три функции, каждая из которых будет выполнять конкретную формулу. Название функций обязательно должны соответствовать букве формулы:  $T()$ ,  $W()$  и  $M()$ . Для того чтобы найти  $t$  - необходимо сначала найти  $w$ , и т.д. Так что используйте результат функции  $W()$  в формуле функции  $T()$  - то-есть вызывайте функцию  $W()$  в  $T()$ . Аналогично и с  $W()$ ,  $M()$ .

$$t = \frac{6}{w}, w = \sqrt{\frac{k}{m}}, m = p * v$$

**ВАЖНО!** Считайте, что пакет `main` уже объявлен, а также функция `main()` с вызовом ВАШЕЙ будущей функции `T()` уже есть. Несмотря на то, что тестирование будет через ввод-вывод, вам НЕ требуется вводить и выводить что-либо. Для подсчета используйте УЖЕ ВВЕДЕННЫЕ ГЛОБАЛЬНЫЕ переменные `k, p, v` ТИПА `float64` !!!

Пакет `math` уже импортирован! Напоминаю: корень (`sqrt`) можно найти с помощью пакета `"math"`, например:

Рисунок 7

Для выполнения задания нам придется создать 3 взаимосвязанные функции, одна из которых использует 2 оставшиеся для получения результата.

```
3  import (
4      "fmt"
5      "math"
6  )
7
8  func M(p float64, v float64) float64{return p*v}
9
10 func W(k float64, p float64, v float64) float64{return math.Sqrt(k/M(p, v))}
11
12 func T(k float64, p float64, v float64) float64{return 6/W(k, p, v)}
13
14 Codeium: Refactor | Explain | Generate GoDoc | X
15 func main() {
16     var k float64
17     var p float64
18     var v float64
19     fmt.Scanf("%f %f %f", &k, &p, &v)
20     fmt.Print(T(k, p, v))
21 }
```

Рисунок 8 (Мой результат)

```
PS C:\Users\1\Desktop>
1296 6 6
1
```

Рисунок 9 (Пример вывода)

## Задание 4

Дана строка, содержащая только английские буквы (большие и маленькие). Добавить символ '\*' (звездочка) между буквами (перед первой буквой и после последней символ '\*' добавлять не нужно).

### Входные данные

Вводится строка ненулевой длины. Известно также, что длина строки не превышает 1000 знаков.

### Выходные данные

Вывести строку, которая получится после добавления символов '\*'.

### Sample Input:

LIItBeoFLcSGBDfQxMHoIuDDWcqcVgkcRoAeocX0

### Sample Output:

L\*I\*t\*t\*B\*e\*o\*f\*L\*c\*S\*G\*B\*D\*f\*Q\*x\*M\*H\*o\*I\*u\*D\*D\*W\*c\*q\*c\*V\*g\*k\*c\*R\*o\*A\*e\*o\*c\*X\*0

Рисунок 10

Сразу заметно, что \* не ставится в 2х случаях когда `index==0` или `index==len(str)`. Не забудем обработать это в коде! А так идея обработки такая же, как и раньше.

```
package main
```

```
import "fmt"
```

Codeium: Refactor | Explain | Generate GoDoc | X

```
func main() {  
    var str string  
    fmt.Scan(&str)  
    for index, digit := range str {  
        if index == 0 || index == len(str) {  
            fmt.Print(string(digit))  
        } else {  
            fmt.Print("*")  
            fmt.Print(string(digit))  
        }  
    }  
}
```

Рисунок 11 (Мой результат)

```
PS C:\Users\1\De  
dDesSQ  
d*D*e*s*s*Q
```

Рисунок 12 (Пример вывода)

## Задание 5

На вход подаются  $a$  и  $b$  - катеты прямоугольного треугольника. Нужно найти длину гипотенузы

**Sample Input:**

6 8

**Sample Output:**

10

Рисунок 13

Надо принять 2 значения на входе и вернуть квадратный корень из суммы их квадратов (Теорема Пифагора).

```
package main

import (
    "fmt"
    "math"
)

Codeium: Refactor | Explain | Generate GoDoc | X
func main() {
    var a int
    var b int
    fmt.Scanf("%d %d", &a, &b)
    fmt.Print(math.Sqrt(float64(a*a + b*b)))
}
```

Рисунок 14 (Мой результат)

```
PS C:\Use
3 4
5
```

Рисунок 15 (Пример вывода)

Заключение: Я научился базовым навыкам языка Golang. Обработка значений, логические операторы, циклы...