

# Лаб: По-сложни цикли

Задачи за упражнение и домашно към курса ["Основи на програмирането с Go"](https://judge.softuni.org/Contests/3363/Advanced-Loops-Lab).

Тествайте решенията си в Judge системата: <https://judge.softuni.org/Contests/3363/Advanced-Loops-Lab>

## 1. Четене на думи

Напишете програма, която чете текст от конзолата (string) и го принтира, докато не получи командата "Stop".

### Примерен вход и изход

| вход       | изход    |
|------------|----------|
| Nakov      | Nakov    |
| SoftUni    | SoftUni  |
| Sofia      | Sofia    |
| Bulgaria   | Bulgaria |
| SomeText   | SomeText |
| Stop       |          |
| AfterStop  |          |
| Europe     |          |
| HelloWorld |          |

| вход      | изход  |
|-----------|--------|
| Sofia     | Sofia  |
| Berlin    | Berlin |
| Moscow    | Moscow |
| Athens    | Athens |
| Madrid    | Madrid |
| London    | London |
| Paris     | Paris  |
| Stop      |        |
| AfterStop |        |

## 2. Парола

Напишете програма, която първоначално прочита име и парола на потребителски профил. След това чете парола за вход.

- при въвеждане на грешна парола: потребителя да се подкани да въведе нова парола.
- при въвеждане на правилна парола: отпечатваме "Welcome {username}!".

### Примерен вход и изход

| вход  | изход          |
|-------|----------------|
| Nakov | Welcome Nakov! |
| 1234  |                |
| pass  |                |
| 1324  |                |
| 1234  |                |

| вход   | изход          |
|--------|----------------|
| Gosho  | Welcome Gosho! |
| secret |                |
| secret |                |

## 3. Сума от числа

Напишете програма, която чете цяло число от конзолата и на всеки следващ ред цели числа, докато тяхната сума стане по-голяма или равна на първоначалното число. След приключване на четенето да се отпечата сумата на въведените числа.

## Примерен вход и изход

| вход | изход | вход | изход |
|------|-------|------|-------|
| 100  | 100   | 20   | 21    |
| 10   |       | 1    |       |
| 20   |       | 2    |       |
| 30   |       | 3    |       |
| 40   |       | 4    |       |
|      |       | 5    |       |
|      |       | 6    |       |

## 4. Редица числа $2k + 1$

Напишете програма, която чете число  $n$ , въведено от потребителя, и отпечатва **всички числа**  $\leq n$  от редицата: **1, 3, 7, 15, 31, ...** Всяко следващо число се изчислява като умножим **предишното** с **2** и добавим **1**.

## Примерен вход и изход

| вход | изход  |
|------|--------|
| 3    | 1<br>3 |

| вход | изход       |
|------|-------------|
| 8    | 1<br>3<br>7 |

| вход | изход             |
|------|-------------------|
| 17   | 1<br>3<br>7<br>15 |

| вход | изход                   |
|------|-------------------------|
| 31   | 1<br>3<br>7<br>15<br>31 |

## 5. Баланс по сметка

Напишете програма, която пресмята **колко общо пари** има в сметката, след като направите **определен брой вноски**. На всеки ред ще получавате сумата, която трябва да внесете в сметката, до получаване на команда "NoMoreMoney". При всяка получена сума на конзолата трябва да се извежда **"Increase: "** + сумата и тя да се **прибавя в сметката**. Ако получите число **по-малко от 0** на конзолата трябва да се изведе **"Invalid operation!"** и **програмата да приключи**. Когато програмата приключи трябва да се принтира **"Total: "** + общата сума в сметката форматирана до втория знак след десетичната запетая.

## Примерен вход и изход

| вход        | изход            |
|-------------|------------------|
| 5.51        | Increase: 5.51   |
| 69.42       | Increase: 69.42  |
| 100         | Increase: 100.00 |
| NoMoreMoney | Total: 174.93    |

| вход  | изход              |
|-------|--------------------|
| 120   | Increase: 120.00   |
| 45.55 | Increase: 45.55    |
| -150  | Invalid operation! |
|       | Total: 165.55      |

## 6. Най-голямо число

Напишете програма, която до получаване на командата "Stop", чете **цели числа**, въведени от потребителя и намира **най-голямото** измежду тях. Въвежда се по едно число на ред.

## Примерен вход и изход

| вход | изход |
|------|-------|
|------|-------|

| вход | изход |
|------|-------|
|------|-------|

| вход | изход |
|------|-------|
|------|-------|

| вход | изход |
|------|-------|
|------|-------|

| вход | изход |
|------|-------|
|------|-------|

|      |     |  |      |    |  |      |    |  |      |     |  |      |    |
|------|-----|--|------|----|--|------|----|--|------|-----|--|------|----|
| 100  | 100 |  | -10  | 20 |  | 45   | 99 |  | 999  | 999 |  | -1   | -1 |
| 99   |     |  | 20   |    |  | -20  |    |  | Stop |     |  | -2   |    |
| 80   |     |  | -30  |    |  | 7    |    |  |      |     |  | Stop |    |
| 70   |     |  | Stop |    |  | 99   |    |  |      |     |  |      |    |
| Stop |     |  |      |    |  | Stop |    |  |      |     |  |      |    |

## 7. Най-малко число

Напишете програма, която до получаване на командата "Stop", чете **цели числа**, въведени от потребителя и намира **най-малкото** измежду тях. Въвежда се по едно число на ред.

### Примерен вход и изход

| вход | изход | вход | изход | вход | изход | вход | изход | вход | изход |
|------|-------|------|-------|------|-------|------|-------|------|-------|
| 100  | 70    | -10  | -30   | 45   | -20   | 999  | 999   | -1   | -2    |
| 99   |       | 20   |       | -20  |       | Stop |       | -2   |       |
| 80   |       | -30  |       | 7    |       |      |       | Stop |       |
| 70   |       | Stop |       | 99   |       |      |       |      |       |
| Stop |       |      |       | Stop |       |      |       |      |       |