**Лаб: Повторения с цикли – While цикъл**

## Четене на думи

Напишете програма, която чете текст от конзолата(стринг) и го принтира, докато не получи командата "**Stop**".  
Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| Gadzhalova  PGEE  Sliven  Bulgaria  SomeText  Stop  AfterStop  Europe  HelloWorld | Gadzhalova  PGEE  Sliven  Bulgaria  SomeText |  | Sofia  Berlin  Moscow Athens  Madrid  London  Paris  Stop | Sofia  Berlin  Moscow Athens  Madrid  London  Paris |

## 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** |  | **вход** | **изход** |
| Gadzhalova  1234  pass  1324  1234 | Welcome Gadzhalova! | Gosho  secret  secret | Welcome Gosho! |

### Насоки

1. Инициализирайте две променливи username и password, които ще съдържат потребителското име и паролата:
2. Инициализирайте променлива data, която ще държи въведената от потребителя парола за вход:
3. В while цикъл, до въвеждане на валидна парола, четете нова:
4. Когато се въведе **валидна парола** **принтирайте съобщението за успешен вход**:

## 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  10  20  30  40 | 100 |  | 20  1  2  3  4  5  6 | 21 |

## Редица числа 2k + 1

Напишете програма, която чете число n, въведено от потребителя,и отпечатва **всички числа ≤ 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** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 3 | 1  3 | 8 | 1  3  7 | 17 | 1  3  7  15 | 31 | 1  3  7  15  31 |

### Насоки

1. Прочетете от конзолата цяло число;
2. **Създайте** променлива с **първоначална стойност** 1, която ще използвате за брояч:
3. **Създайте while** цикъл, който се повтаря, докато **броячът е по-малък** от числото, прочетено от конзолата:
4. При всяко повторение на цикъла **принтирайте стойността на брояча** и му **прибавяйте дадената стойност**:

## 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** |  | **вход** | **изход** |
| 5.51  69.42  100  NoMoreMoney | Increase: 5.51  Increase: 69.42  Increase: 100.00  Total: 174.93 | 120  45.55  -150 | Increase: 120.00  Increase: 45.55  Invalid operation!  Total: 165.55 |

## Най-голямо число

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

### Примерен вход и изход

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

## Най-малко число

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

### Примерен вход и изход

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

## Завършване

Напишете програма, която изчислява **средната оценка** на ученик от цялото му обучение. На първия ред ще получите **името на ученика**, а на всеки следващ ред неговите годишни оценки. Ученикът преминава в следващия клас, **ако годишната му оценка е по-голяма или равна на 4.00**. **Ако ученикът бъде скъсан повече от един път, то той бива изключен и програмата приключва,** като се отпечатва **името на ученика и в кой клас бива изключен**.

При успешно завършване на **12-ти** клас да се отпечата :

"{**име на ученика**} graduated. Average grade: {**средната оценка от цялото обучение**}"

**В случай, че ученикът е изключен от училище, да се отпечата:**

"{**име на ученика**} has been excluded at {**класа, в който е бил изключен**} grade"

**Стойността трябва да бъде форматирана до втория знак** след десетичната запетая.

### Примерен вход и изход

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |
| Gosho  5  5.5  6  5.43  5.5  6  5.55  5  6  6  5.43  5 | Gosho graduated. Average grade: 5.53 | Mimi  5  6  5  6  5  6  6  2  3 | Mimi has been excluded at 8 grade |