# Лаб: Условни конструкции

## Отлична оценка

Първата задача от тази тема е да се напише **конзолна програма**, която **чете оценка** (реално число), въведена от потребителя и отпечатва "Excellent!**",** ако оценката е **5.50** или по-висока.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 6 | Excellent! | 5 | *(няма изход)* | 5.50 | Excellent! | 5.49 | *(няма изход)* |

### Насоки:

1. Създайте Python файл с подходящо име, например **excellent\_grade**. Създайте една променлива, в която да запазите **реално** **число** – оценката, което ще прочетете от конзолата:
2. Направете проверка за стойността на оценката. Ако тя е по-голяма или равна на 5.50 отпечатайте изхода по условие:
3. Стартирайте програмата с **Ctrl + Shift + F10** и я **тествайте** с различни входни стойности:

|  |  |
| --- | --- |
|  |  |

## По-голямото число

Да се напише програма, която чете **две цели числа** въведени от потребителя и отпечатва **по-голямото от двете**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5  3 | 5 | 3  5 | 5 | 10  10 | 10 | -5  5 | 5 |

### Насоки:

1. Създайте Python файл с подходящо име в проекта
2. Прочетете две цели числа от конзолата:
3. Сравнете, дали първото число first\_number e по-голямо от второто second\_number. Отпечатайте по-голямото число.

## Четно или нечетно

Да се напише програма, която чете **цяло число**, въведено от потребителя, и печата дали е **четно** или **нечетно**.

Ако е **четно** отпечатайте "**even"**, ако е нечетно "**odd"**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2 | even | 3 | odd | 25 | odd | 1024 | even |

### Насоки:

1. Създайте Python файл с подходящо име в съществуващия проект;
2. Прочетете едно цяло число от конзолата:
3. Проверете дали числото е четно, като го разделите модулно на 2 и проверите дали има остатък от делението. Ако няма остатък, отпечатайте изход "even". В противен случай отпечатайте "odd":

## Познай паролата

Да се напише програма, която **чете парола** (текст), въведена от потребителя и проверява дали въведената парола **съвпада** с фразата "s3cr3t!P@ssw0rd".При съвпадение да се изведе "**Welcome**". При несъвпадение да се изведе "**Wrong password!**".

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| qwerty | Wrong password! | s3cr3t!P@ssw0rd | Welcome | s3cr3t!p@ss | Wrong password! |

## Число от 100 до 200

Да се напише програма, която **чете цяло число**, въведено от потребителя и проверява дали е **под 100**, **между 100 и 200** или **над 200**. Ако числото е:

* под 100 отпечатайте: **"Less than 100"**
* между 100 и 200 отпечатайте: **"Between 100 and 200"**
* над 200 отпечатайте: **"Greater than 200"**

### Примерен вход и изход

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 95 | Less than 100 | 120 | Between 100 and 200 | 210 | Greater than 200 |

## Информация за скоростта

Да се напише програма, която **чете скорост** **(реално число)**, въведена от потребителяи отпечатва **информация за скоростта**.

* При скорост **до 10** (включително) отпечатайте **"slow"**
* При скорост **над 10** и **до 50** (включително) отпечатайте **"average"**
* При скорост **над 50** и **до 150** (включително) отпечатайте **"fast"**
* При скорост **над 150** и **до 1000** (включително) отпечатайте **"ultra fast"**
* При по-висока скорост отпечатайте **"extremely fast"**

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 8 | slow | 49.5 | average | 126 | fast | 160 | ultra fast | 3500 | extremely fast |

## Лица на фигури

Да се напише програма, в която потребителят **въвежда вида и размерите на геометрична** фигура и пресмята лицето й. Фигурите са четири вида: квадрат (**square**), правоъгълник (**rectangle**), кръг (**circle**) и триъгълник (**triangle**). На първия ред на входа се чете вида на фигурата (текст със следните възможности: square, rectangle, circle или triangle).

* Ако фигурата е **квадрат (**square): на следващия ред се чете едно дробно число - дължина на страната му
* Ако фигурата е **правоъгълник (**rectangle): на следващите два реда четат две дробни числа - дължините на страните му
* Ако фигурата е **кръг (**circle): на следващия ред чете едно дробно число - радиусът на кръга
* Ако фигурата е **триъгълник (**triangle): на следващите два реда четат две дробни числа - дължината на страната му и дължината на височината към нея

Резултатът да се закръгли до **3 цифри след десетичната запетая**.

### Примерен вход и изход

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| square  5 | 25.000 | rectangle  7  2.5 | 17.500 | circle  6 | 113.097 | triangle  4.5  20 | 45.000 |