# Лаб: Прости проверки

Задачи за упражнение в клас и за домашно към курса [„Основи на програмирането“ @ СофтУни](https://softuni.bg/courses/programming-basics).

## Създаване на PyCharm проект

Създайте нов проект (**“Simple-Conditions”**), в който да съхраните решенията на задачите.

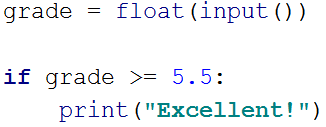
## Проверка за отлична оценка

Първата задача от тази тема е да се напише **конзолна програма**, която **въвежда оценка** (десетично число) и отпечатва “**Excellent!**”, ако оценката е **5.50** или по-висока.

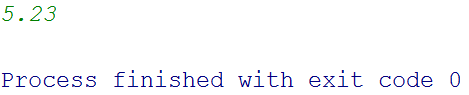
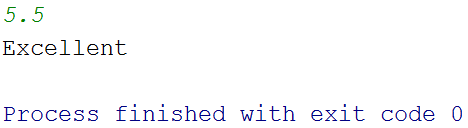
### Примерен вход и изход:

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

1. Създайте **нов файл**
2. Направете нов файл с име “Excellent-Result”:
3. Отидете в **началото на файла** и напишете решението на задачата. Можете да си помогнете с кода от картинката по-долу:



1. **Стартирайте** програмата с [Alt+Shift+F10] и я **тествайте** с различни входни стойности:

1. **Тествайте** решението си в **judge системата** <https://judge.softuni.bg/Contests/Practice/Index/1185#0>. Трябва да получите **100 точки** (напълно коректно решение):



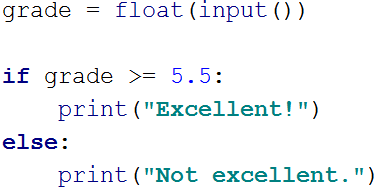
## Отлична оценка или не

Следващата задача от тази тема е да се напише **конзолна програма**, която **въвежда оценка** (десетично число) и отпечатва “**Excellent!**”, ако оценката е **5.50** или по-висока, или “**Not excellent.**” в противен случай.

### Примерен вход и изход:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 6 | Excellent! | 5 | Not excellent. | 5.50 | Excellent! | 5.49 | Not excellent. |

1. Направете нов файл с име “excellent\_or\_not”.
2. **Напишете кода** на програмата. Може да си помогнете с примерния код от картинката:



1. Сега **стартирайте програмата**, както обикновено с [Alt+Shift+F10] и я тествайте

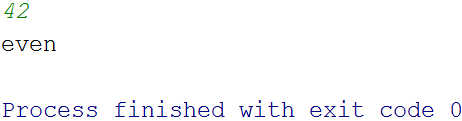
## Четно или нечетно

Да се напише програма, която въвежда **цяло число** и печата дали е **четно** или **нечетно**.

### Примерен вход и изход:

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

1. Първо добавете **нов файл.**
2. **Напишете кода** на програмата. Проверката за четност може да се реализира чрез проверка на **остатъка при деление на 2** по следния начин: even = (num % 2 == 0).
3. **Стартирайте** програмата с **[Alt+Shift+F10]** и я тествайте:



1. Тествайте в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/1185#1>

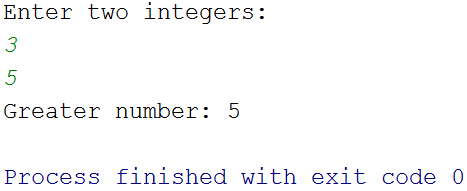
## Намиране на по-голямото число

Да се напише програма, която въвежда **две цели числа** 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 |

1. Първо добавете **нов файл**.
2. **Напишете кода** на програмата. Необходима е единична if-else конструкция.
3. **Стартирайте** програмата с **[Alt+Shift+F10]** и я тествайте:



1. Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0> .

## Изписване на число до 9 с думи

Да се напише програма, която въвежда **цяло число в диапазона [0…10]** и го **изписва с думи** на английски език. Ако числото е извън диапазона, изписва “**number too big**”.

### Примерен вход и изход:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5 | five | 1 | one | 9 | nine | 10 | number too big |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0> .

**Подсказка**: можете да напишете дълга if-elif-elif-elif…else, с която да разгледате възможните **11 случая**.

## Познай паролата

Да се напише програма, която **въвежда парола** (един ред с произволен текст) и проверява дали въведеното **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! |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0> .

**Подсказка**: използвайте if-else конструкцията.

## Число от 100 до 200

Да се напише програма, която **въвежда цяло число** и проверява дали е **под 100**, **между 100 и 200** или **над 200**. Да се отпечатат съответно съобщения като в примерите по-долу:

### Примерен вход и изход:

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

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0> .

**Подсказка**: използвайте if-elif-elif-else конструкция за да проверите всеки от трите случая.

## Еднакви думи

Да се напише програма, която **въвежда две думи** и проверява дали са еднакви. Да не се прави разлика между главни и малки думи. Да се изведе “yes” или “no”.

### Примерен вход и изход:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **Вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| Hello  Hello | yes | SoftUni  Softuni | yes | Soft  Uni | no | beer  vodka | no | HeLlO  hELLo | yes |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0> .

**Подсказка**: използвайте if-else конструкция. Преди сравняване на думите ги обърнете в долен регистър: word = word.lower().

## Информация за скоростта

Да се напише програма, която **въвежда скорост** (десетично число)и отпечатва **информация за скоростта**. При скорост **до 10** (включително) отпечатайте “**slow**”. При скорост **над 10** и **до 50** отпечатайте “**average**”. При скорост **над 50** и **до 150** отпечатайте “**fast**”. При скорост **над 150** и **до 1000** 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 |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0> .

**Подсказка**: използвайте серия от if-elif-else конструкции, за да хванете всичките 5 случая.

## Лица на фигури

Да се напише програма, която **въвежда 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 (square, rectangle, circle или triangle). Ако фигурата е **квадрат**, на следващия ред се чете едно число – дължина на страната му. Ако фигурата е **правоъгълник**, на следващите два реда четат две числа – дължините на страните му. Ако фигурата е **кръг**, на следващия ред чете едно число – радиусът на кръга. Ако фигурата е **триъгълник**, на следващите два реда четат две числа – дължината на страната му и дължината на височината към нея. Резултатът да се закръгли до **3 цифри след десетичната точка**.

### 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 |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/152#0> .

**Подсказка**: използвайте серия от if-elif-else конструкции, за да обработите 4-те вида фигури.