# Лаб: Условни конструкции

Задачи за упражнение в клас и за домашно към курса: ["Основи на програмирането със С++" @СофтУни](https://softuni.bg/trainings/3654/programming-basics-with-cpp-january-2022)

Тествайте решенията си в **Judge** системата: <https://judge.softuni.org/Contests/Compete/Index/530>

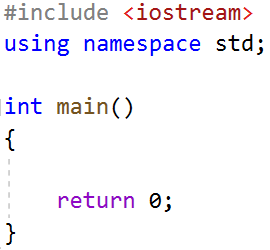
## Отлична оценка

Напише **конзолна програма**, която **чете оценка** (реално число), въведена от потребителя и отпечатва "Excellent!**"** ако оценката е **5.50** или по-висока.

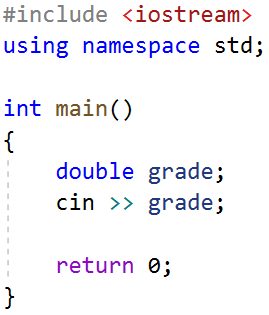
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 6 | Excellent! | 5 | *(няма изход)* | 5.50 | Excellent! | 5.49 | *(няма изход)* |

### Насоки:

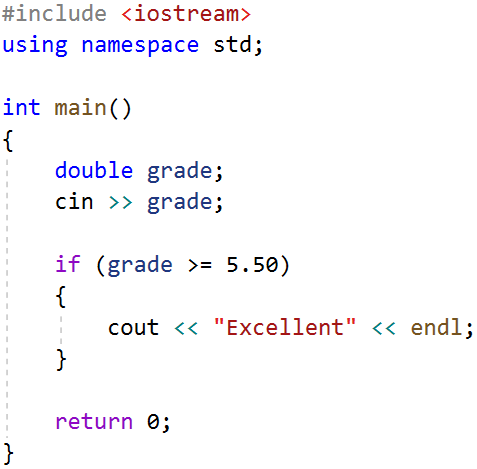
1. Създайте **нов файл** с име "ExcellentResult":
2. Създайте нов main() **метод**



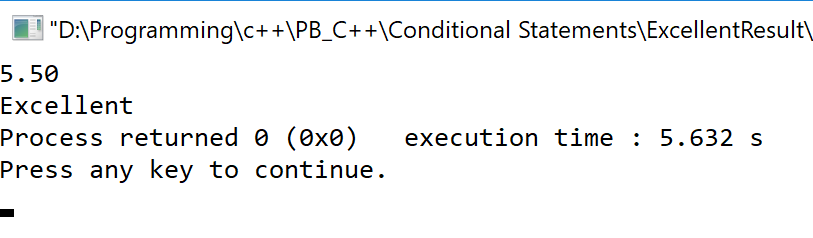
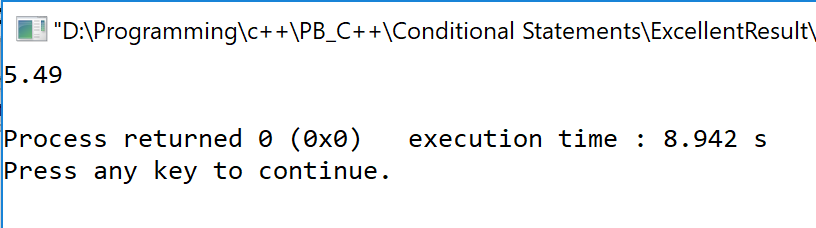
1. Отидете в тялото на метода main() (между къдравите скоби). Създайте една променлива, в която да запазите **реално** **число** – оценката, което ще прочетете от конзолата:



1. Направете проверка за стойността на оценката. Ако тя е по-голяма или равна на 5.50 отпечатайте изхода по условие:



1. Стартирайте програмата с **Ctrl + F5** и я **тествайте** с различни входни стойности:

## По-голямото число

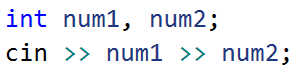
Да се напише програма, която чете **две цели числа**, въведени от потребителя и отпечатва **по-голямото от двете**.

### Примерен вход и изход

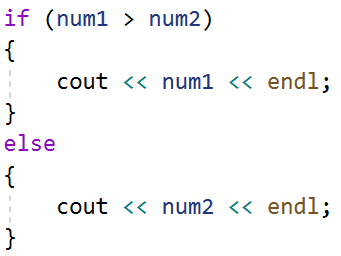
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 5  3 | 5 | 3  5 | 5 | 10  10 | 10 | -5  5 | 5 |

### Насоки:

1. Прочетете две цели числа от конзолата:



1. Сравнете, дали първото число num1 e по-голямо от второто num2. Отпечатайте по-голямото число.



## 3. Четно или нечетно

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

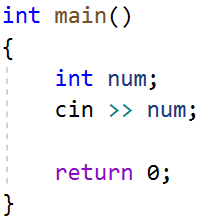
Ако е **четно** отпечатайте "**even"**, ако е нечетно "**odd"**.

### Примерен вход и изход

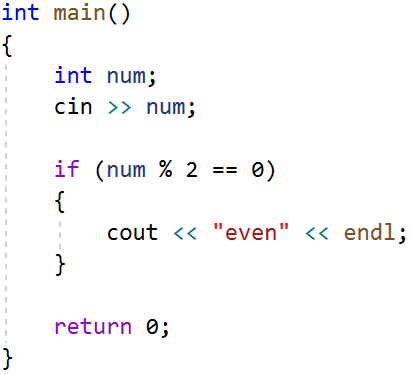
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 2 | even | 3 | odd | 25 | odd | 1024 | even |

### Насоки:

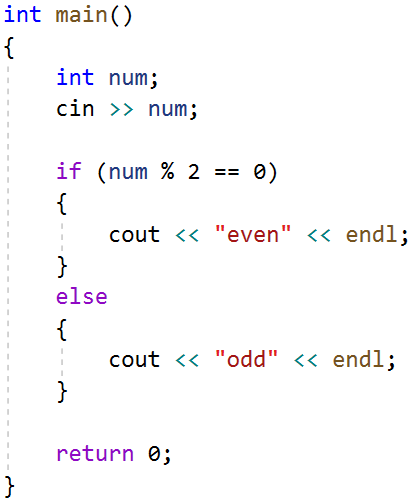
1. Прочетете едно цяло число от конзолата:



1. Проверете, дали числото е четно, като го разделите на 2 и проверите, дали има остатък при делението. Отпечатайте изхода по условие – текстът "even":



1. В противен случай отпечатайте "odd":



## 4. 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! |

## 5. 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:

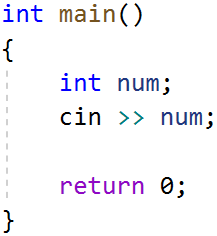
* под 100 отпечатайте: **"Less than 100"**
* между 100 и 200 (включително) отпечатайте: **"Between 100 and 200"**
* над 200 отпечатайте: **"Greater than 200"**

### Примерен вход и изход

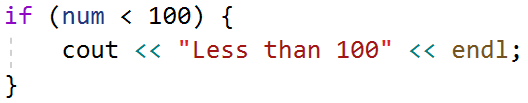
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 95 | Less than 100 | 120 | Between 100 and 200 | 210 | Greater than 200 |

### Насоки:

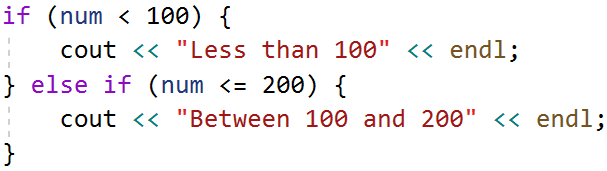
1. Прочетете едно цяло число от конзолата:

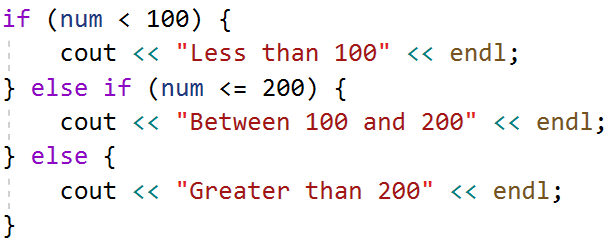


1. Проверете дали числото е **по-малко от 100**:



1. Ако числото не е по-малко от 100 проверете, дали е **по-малко или равно на 200**:



1. Ако числото не е по-малко от 200, то то е по-голямо. Изпишете текста по условие: 

## 6. 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 |

## 7. 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

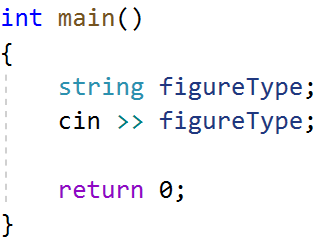
Резултатът да се закръгли до **3 цифри след десетичната запетая**.

### Примерен вход и изход

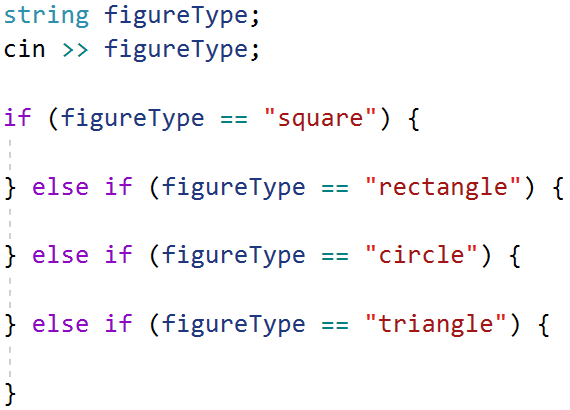
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| square  5 | 25.000 | rectangle  7  2.5 | 17.500 | circle  6 | 113.097 | triangle  4.5  20 | 45.000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

### Насоки:

1. Прочетете тип на фигурата – текст въведен от конзолата:



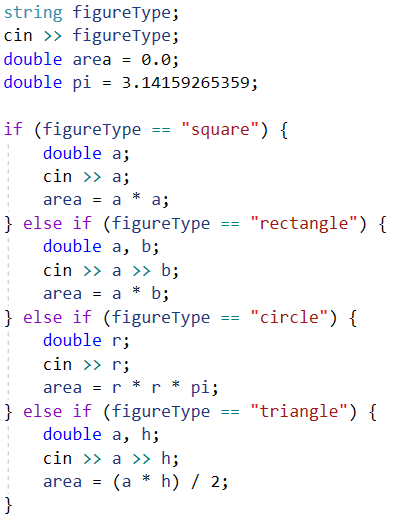
1. Направете **if-else** конструкция, която съдържа случаи за типа фигури:



1. Създайте променлива **double pi** със стойност **3.14159265359.**



1. За всеки случай, четете нужния брой страни и извършвайте математически операции за намиране на лицето според типа на фигурата. Направете променлива area, в която да пазите пресметнатото лице на фигурата:



1. Принтирайте стойността на променливата area, като форматирате стойността й до третия знак след десетичната запетая:  
   