

# Лаб: Условни конструкции

Задачи за упражнение в клас и за домашно към курса: "[Основи на програмирането със C++](#)" @СофтУни

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

## 1. 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
6	Excellent!

вход	изход
5	(няма изход)

вход	изход
5.50	Excellent!

вход	изход
5.49	(няма изход)

### Насоки:

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

```
#include <iostream>
using namespace std;

int main()
{
    // ...

    return 0;
}
```

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

```
#include <iostream>
using namespace std;

int main()
{
    double grade;
    cin >> grade;

    // ...

    return 0;
}
```

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

```
#include <iostream>
using namespace std;

int main()
{
    double grade;
    cin >> grade;

    if (grade >= 5.50)
    {
        cout << "Excellent" << endl;
    }

    return 0;
}
```

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

```
"D:\Programming\c++\PB_C++\Conditional Statements\ExcellentResult\
5.50
Excellent
Process returned 0 (0x0)   execution time : 5.632 s
Press any key to continue.
■
```

```
"D:\Programming\c++\PB_C++\Conditional Statements\ExcellentResult\
5.49

Process returned 0 (0x0)   execution time : 8.942 s
Press any key to continue.
```

## 2. 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
5 3	5

вход	изход
3 5	5

вход	изход
10 10	10

вход	изход
-5 5	5

### Насоки:

1. Прочетете две цели числа от конзолата:

```
int num1, num2;
cin >> num1 >> num2;
```

2. Сравнете, дали първото число **num1** е по-голямо от второто **num2**. Отпечатайте по-голямото число.

```

if (num1 > num2)
{
    cout << num1 << endl;
}
else
{
    cout << num2 << endl;
}

```

### 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
2	even

ВХОД	ИЗХОД
3	odd

ВХОД	ИЗХОД
25	odd

ВХОД	ИЗХОД
1024	even

#### Насоки:

1. Прочетете едно цяло число от конзолата:

```

int main()
{
    int num;
    cin >> num;

    return 0;
}

```

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

```
int main()
{
    int num;
    cin >> num;

    if (num % 2 == 0)
    {
        cout << "even" << endl;
    }

    return 0;
}
```

3. В противен случай отпечатайте "odd":

```
int main()
{
    int num;
    cin >> num;

    if (num % 2 == 0)
    {
        cout << "even" << endl;
    }
    else
    {
        cout << "odd" << endl;
    }

    return 0;
}
```

## 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
qwerty	Wrong password!

вход	изход
s3cr3t!P@ssw0rd	Welcome

вход	изход
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
95	Less than 100

ВХОД	ИЗХОД
120	Between 100 and 200

ВХОД	ИЗХОД
210	Greater than 200

## Насоки:

1. Прочетете едно цяло число от конзолата:

```
int main()
{
    int num;
    cin >> num;

    return 0;
}
```

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

```
if (num < 100) {
    cout << "Less than 100" << endl;
}
```

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

```
if (num < 100) {
    cout << "Less than 100" << endl;
} else if (num <= 200) {
    cout << "Between 100 and 200" << endl;
}
```

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

```
if (num < 100) {
    cout << "Less than 100" << endl;
} else if (num <= 200) {
    cout << "Between 100 and 200" << endl;
} else {
    cout << "Greater than 200" << endl;
}
```

## 6. Информация за скоростта

Да се напише програма, която **чете скорост (реално число)**, въведена от потребителя и 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
8	slow

вход	изход
49.5	average

вход	изход
126	fast

вход	изход
160	ultra fast

вход	изход
3500	extremely fast

## 7. Лица на фигури

Да се напише програма, в която потребителят **въвежда вида и размерите на геометрична фигура** и пресмята лицето ѝ. 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
square 5	25.000

вход	изход
rectangle 7 2.5	17.500

вход	изход
circle 6	113.097

вход	изход
triangle 4.5 20	45.000

## Насоки:

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

```
int main()
{
    string figureType;
    cin >> figureType;

    return 0;
}
```

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

```
string figureType;
cin >> figureType;

if (figureType == "square") {

} else if (figureType == "rectangle") {

} else if (figureType == "circle") {

} else if (figureType == "triangle") {

}
```

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

```
double pi = 3.14159265359;
```

4. 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:

```
string figureType;
cin >> figureType;
double area = 0.0;
double pi = 3.14159265359;

if (figureType == "square") {
    double a;
    cin >> a;
    area = a * a;
} else if (figureType == "rectangle") {
    double a, b;
    cin >> a >> b;
    area = a * b;
} else if (figureType == "circle") {
    double r;
    cin >> r;
    area = r * r * pi;
} else if (figureType == "triangle") {
    double a, h;
    cin >> a >> h;
    area = (a * h) / 2;
}
```

5. Принтирайте стойността на променливата **area**, като форматирате стойността ѝ до третия знак след десетичната запетая:

```
cout.setf(ios::fixed);  
cout.precision(3);  
cout << area << endl;
```