

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.bg/Contests/Compete/Index/530>

1. Проверка за отлична оценка

Първата задача от тази тема е да се напише **конзолна програма**, която **чете оценка** (десетично 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 C++ .cpp 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. Намиране на по-голямото число

Да се напише програма, която чете **две цели числа** въведени от потребителя и отпечата **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 отпечатва на конзолата, дали е **четно** или **нечетно**.

Примерен вход и изход

вход	изход
2	even

вход	изход
3	odd

вход	изход
25	odd

вход	изход
1024	even

Насоки:

1. Първо създайте нов **.cpp файл** с подходящо име
2. Прочетете едно цяло число от конзолата:

```

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

    return 0;
}

```

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

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

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

    return 0;
}
```

4. 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. 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**. Да се отпечатаат съответно съобщения, като в примерите по-долу:

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;
}
```

5. Познай паролата

Да се напише програма, която **чете парола** (един ред с произволен текст), въведена от потребителя 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!

Насоки:

1. Прочетете входните данни – един ред текст от конзолата:

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

    return 0;
}
```

2. Сравнете стойността на прочетената парола с текста "s3cr3t!P@ssw0rd". Отпечатайте правилния изход, спрямо въведената парола:

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

    if (password == "s3cr3t!P@ssw0rd") {
        cout << "Welcome" << endl;
    }
    return 0;
}
```

3. Ако паролата не е правилна, отпечатайте съответното съобщение:

```
if (password == "s3cr3t!P@ssw0rd") {
    cout << "Welcome" << endl;
} else {
    cout << "Wrong password!" << endl;
}
```

6. 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 (**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 се 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
square 5	25.000

вход	изход
rectangle 7	17.500

вход	изход
circle 6	113.097

вход	изход
triangle 4.5	45.000

--	--

2.5	
-----	--

--	--

20	
----	--

Насоки:

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

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

    return 0;
}
```

2. Направете **if-else if** 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 форматирайте стойността ѝ до третия знак след десетичната запетая:

```

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

```

Примерна изпитна задача

7. Магазин за детски играчки

Петя има магазин за детски играчки. Тя получава голяма поръчка, която трябва да изпълни. С парите, които ще спечели иска да отиде на екскурзия. Да се напише програма, която пресмята печалбата от поръчката.

Цени на играчките:

- Пъзел - 2.60 лв.
- Говореща кукла - 3 лв.
- Плюшено мече - 4.10 лв.
- Миньон - 8.20 лв.
- Камионче - 2 лв.

Ако поръчаните играчки са **50 или повече** магазинът прави **отстъпка 25% от общата цена**. От спечелените пари Петя трябва да даде **10% за наема** на магазина. Да се пресметне дали парите ще ѝ стигнат да отиде на екскурзия.

Вход

От конзолата се четат **6 реда**:

1. Цена на екскурзията - реално число в интервала [1.00 ... 10000.00]
2. Брой пъзели - цяло число в интервала [0... 1000]
3. Брой говорещи кукли - цяло число в интервала [0 ... 1000]
4. Брой плюшени мечета - цяло число в интервала [0 ... 1000]
5. Брой миньони - цяло число в интервала [0 ... 1000]
6. Брой камиончета - цяло число в интервала [0 ... 1000]

Изход

На конзолата се отпечатва:

- Ако парите са достатъчни се отпечатва:
 - "Yes! {оставащите пари} lv left."
- Ако парите НЕ са достатъчни се отпечатва:
 - "Not enough money! {недостигащите пари} lv needed."

Резултатът трябва да се форматира до втория знак след десетичната запетая.

Примерен вход и изход

Вход	Изход	Обяснения
40.8 20 25 30 50 10	Yes! 418.20 lv left.	<p>Сума: $20 * 2.60 + 25 * 3 + 30 * 4.10 + 50 * 8.20 + 10 * 2 = 680$ лв.</p> <p>Брой на играчките: $20 + 25 + 30 + 50 + 10 = 135$</p> <p>$135 > 50 \Rightarrow 25\%$ отстъпка; 25% от 680 = 170 лв. отстъпка</p> <p>Крайна цена: $680 - 170 = 510$ лв.</p> <p>Наем: 10% от 510 лв. = 51 лв.</p> <p>Печалба: $510 - 51 = 459$ лв.</p> <p>$459 > 40.8 \Rightarrow 459 - 40.8 = 418.20$ лв. остават</p>
Вход	Изход	Обяснения
320 8 2 5 5 1	Not enough money! 238.73 lv needed.	<p>Сума: 90.3 лв.</p> <p>Брой на играчките: 21</p> <p>$21 < 50 \Rightarrow$ няма отстъпка</p> <p>Наем: 10% от 90.3 = 9.03 лв.</p> <p>Печалба: $90.3 - 9.03 = 81.27$ лв.</p> <p>$81.27 < 320 \Rightarrow 320 - 81.27 = 238.73$ лв. не достигат</p>

Насоки

1. Прочетете входните данни – броя на играчките и цената на екскурзията:

```
double excursionPrice;
cin >> excursionPrice;

int puzzlesCnt, talkingDollsCnt, teddyBearsCnt, minionsCnt, trucksCnt;
cin >> puzzlesCnt >> talkingDollsCnt >> teddyBearsCnt >> minionsCnt >> trucksCnt;
```

- Пресметнете общата цена на всеки тип играчка, като умножите броя на всеки тип играчка по единичната им цена:

```
double puzzlePrice = puzzlesCnt * 2.60;
double talkingDollPrice = talkingDollsCnt * 3;
double teddyBearPrice = teddyBearsCnt * 4.10;
double minionPrice = minionsCnt * 8.20;
double truckPrice = trucksCnt * 2;
```

- Пресметнете общата цена на играчките, като съберете цените на всеки тип играчки.

```
double totalPrice = puzzlePrice + talkingDollPrice
    + teddyBearPrice + minionPrice + truckPrice;
```

- Изчислете общия брой на играчките. Създайте променлива **discount**, в която да пазите стойността на отстъпката. Направете проверка, дали броя на играчките надвишава 50 и ако е вярно, изчислете отстъпката, и присвоете стойността към променливата **discount**:

```
int toysCnt = puzzlesCnt + talkingDollsCnt + teddyBearsCnt
    + minionsCnt + trucksCnt;
double discount = 0.0;
if (toysCnt >= 50)
{
    discount = totalPrice * 0.25;
}
```

- Изчислете крайната цена, като извадите от сумата за всички играчки отстъпката, която сте изчислили, както и парите за наема:

```
double finalPrice = totalPrice - discount;
finalPrice = finalPrice * 0.9;
```

- Сравнете крайната печалба с цената на екскурзията. Изпишете правилния изход спрямо резултата: