



УСЛОВЕН ОПЕРАТОР [SWITCH] ОПЕРАТОРИ ЗА ЦИКЪЛ [FOR]



УСЛОВЕН ОПЕРАТОР SWITCH

• СЕМАНТИКА НА ОПЕРАТОРА SWITCH

Често се налага да се **избере** за изпълнение един от множество варианти.

Варианти на записване

```
IF (УСЛОВИЕ)
{
    ОПЕРАТОР 1
}
ELSE IF (УСЛОВИЕ 2)
{
    ОПЕРАТОР 2
}
ELSE IF (УСЛОВИЕ 3)
{
    ОПЕРАТОР 3
}
ELSE
{
    ОПЕРАТОР 4
}
```

```
SWITCH (ИЗРАЗ)
{
    case <ИЗРАЗ 1> :<ОПЕРРАТОРИ>;
    break;
    case <ИЗРАЗ 2> :<ОПЕРРАТОРИ>;
    break;
    case <ИЗРАЗ 3> :<ОПЕРРАТОРИ>;
    break;
    case <ИЗРАЗ N> :<ОПЕРРАТОРИ>;
    break;
    default :<ОПЕРРАТОРИ>;
}
```



УСЛОВЕН ОПЕРАТОР SWITCH

• СЕМАНТИКА НА ОПЕРАТОРА SWITCH

switch (ключ)

case (случай, вариант)

default (по подразбиране)

Case-израз – израз от допустим тип (**bool, int, char, изборим**). Реалните типове **double** и **float** не са допустими.

Израз 1, 2, 3...N – константни изрази, с различни стойности, съвместими с типа на “израз”

break – Оператор за преход.

default – изпълнява се, в случай че нито един от изразите 1, 2, 3...N не е изпълнен.



УСЛОВЕН ОПЕРАТОР SWITCH

• СЕМАНТИКА НА ОПЕРАТОРА SWITCH

Оператор break - прекратява изпълнението на switch

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

int main()
{
    int i;
    cout<< "Please enter a number" << endl;
    cin>>i;

    switch (i)
    {
        case 1: cout<< "1 is odd number" << endl; break;
        case 3: cout<< "3 is odd number" << endl; break;
        case 5: cout<< "5 is odd number" << endl; break;
        case 7: cout<< "7 is odd number" << endl; break;
        case 9: cout<< "9 is odd number" << endl; break;
        case 0: cout<< "0 is even number" << endl; break;
        case 2: cout<< "2 is even number" << endl; break;
        case 4: cout<< "4 is even number" << endl; break;
        case 6: cout<< "6 is even number" << endl; break;
        case 8: cout<< "8 is even number" << endl; break;
        default:cout << "the number is grater than 9" << endl;
    }
}
```

При i= 1; резултат: 1 is odd number



УСЛОВЕН ОПЕРАТОР SWITCH

• СЕМАНТИКА НА ОПЕРАТОРА SWITCH

Оператор break - прекратява изпълнението на switch

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

int main()
{
    int i;
    cout<< "Please enter a number" << endl;
    cin>>i;

    switch (i)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 9: cout<< "odd number" << endl; break;
        case 0:
        case 2:
        case 4:
        case 6:
        case 8: cout<< "even number" << endl;
    }
}
```

При $i = 1, 3, 5, 7, 9$; резултат: odd number



УСЛОВЕН ОПЕРАТОР SWITCH

• СЕМАНТИКА НА ОПЕРАТОРА SWITCH

Оператор break – прекъсва изпълнението на switch и предава управлението на първия оператор след него

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

int main()
{
    int i;
    cout << "Please enter an odd number between 1 and 9" << endl;
    cin >> i;

    switch (i)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 9: cout << "odd number and "; break;
    }
    switch (i)
    {
        case 1: cout << "it is 1" << endl; break;
        case 3: cout << "it is 3" << endl; break;
        case 5: cout << "it is 5" << endl; break;
        case 7: cout << "it is 7" << endl; break;
        case 9: cout << "it is 7" << endl; break;
    }
}
```



УСЛОВЕН ОПЕРАТОР SWITCH

- ПРЕДИМСТВА НА ОПЕРАТОРА SWITCH ПРЕД IF И IF/ELSE

Прави реализацията по-ясна

- НЕДОСТАТЪК НА ОПЕРАТОРА SWITCH

Наложени ограничения на типа на switch-израза, който трябва да е:

Цял

Булев

Символен

Изброен



ОПЕРАТОР ЗА ЦИКЪЛ FOR

• ЦИКЛИЧЕН ИЗЧИСЛИТЕЛЕН ПРОЦЕС

Изчислителен процес, при който оператор или група оператори се изпълняват многократно за различни стойности на техните параметри, се нарича **цикличен**.

Индуктивен цикличен процес – цикличен процес, при който **броят на повторенията е предварително известен**

Итеративен цикличен процес – цикличен процес, при който **броят на повторенията не е известен предварително**.

Оператор for – чрез него могат да се реализират произволни циклични процеси, но се използва главно за реализиране на индуктивни циклични процеси.



ОПЕРАТОР ЗА ЦИКЪЛ FOR

• СЕМАНТИКА НА ОПЕРАТОРА FOR

```
for ( <инициализация>; <условие>; <корекция> )  
{  
    ОПЕРАТОР – Тяло на цикъла  
}
```

for - запазена дума

<инициализация> - задава начало на изпълнението, като тя се изпълнява **само веднъж в началото на цикъла. В нея не са допустими дефиниции на променливи.**

<условие> - булеви израз. Ако стойността му е **false** изпълнението на **for** завършва **без тялото на цикъла да се изпълни.**

Ако стойността му е **true** се повтарят следните **действия:**

- изпълнение на тялото на цикъла
- корекция
- Пресмятане на стойността на условие, **докато е true**



ОПЕРАТОР ЗА ЦИКЪЛ FOR

- СЕМАНТИКА НА ОПЕРАТОРА FOR

Пример:

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

int main()
{
    int n;
    n = 5;

    for(int i = 1; i<=5; i++)
    {
        cout << n*i << endl;
    }
}
```

Резултат: 5, 10, 15, 20, 25



ЗАДАЧИ

- **ЗАДАЧА 17**

Да се напише програма, в която потребителят въвежда оценка от 2 до 6 и извежда на екрана оценката с думи [Пример: Вход '2', Изход 'Слаб'].

- **ЗАДАЧА 18**

Да се напише програма, в която потребителят въвежда номер на месец и година, програмата извежда броят на дните в него.



ЗАДАЧИ

• ЗАДАЧА 19

В играта спортен бридж се използва колода от 52 карти. Те се разпределят по групи както следва: 2, 3, 4, 5, 6, 7, 8, 9, 10, вале, дама, поп и асо.

Всяка от тях има 4 разновидности: спатия, каро, купа и пика.

Да се състави програма, която при въвеждане на естествено число от интервала $[1 - 52]$ извежда точната карта от колодата.



ЗАДАЧИ

• ЗАДАЧА 20

Да се състави програма, в която потребителят въвежда естествено число и извежда на екрана дали то е просто.

Използвайте операторът за цикъл `for`.

• ЗАДАЧА 21

Да се състави програма, в която потребителят въвежда естествено число и извежда неговите делители.

Използвайте операторът за цикъл `for`.



ЗАДАЧИ

• ЗАДАЧА 22

Да се състави програма, която изчислява сумата от цифрите на всички естествени 2-цифрени числа до въведено от клавиатурата 2-цифрено число.

Пример: при въведено число 13, програмата извежда 51 ($1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 1 + 0 + 1 + 1 + 1 + 2$).

• ЗАДАЧА 23

Да се състави програма, която извежда всички естествени трицифрени числа, които нямат еднакви цифри т.е. 100, 101 и подобните на тях не се извеждат.



ЗАДАЧИ

- Допълнителни задачи

✓ Напишете програма, която приема числа от клавиатурата, докато не срещне 0 и отпечатва сумата и произведението им.

✓ Напишете програма, която приема цяло положително число и намира броя на цифрите му.

Примери:

1 -> 1

1740784 -> 7

875 -> 3

0 -> 0



ВЪПРОСИ