



## **Лекція 4.**

# **Тернарна операція. Оператор switch**

---

Назва «**тернарний**» походить від латинського *ternarius* - потрійний.

Оператор приймає три аргументи.

Якщо перший аргумент істина, то повертається другий аргумент, якщо хибна, то повертається третій.

Синтаксис оператора:

**<умова> ? <аргумент 2> : <аргумент 3>**

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
```

```
int main()
{
```

```
    int a, b;
    printf("a = ");
    scanf("%d", &a);
    printf("b = ");
    scanf("%d", &b);
```

Наприклад,  
якщо

**a == 2, b == 5**

тоді  
управління  
до третього  
аргументу

```
    a > b ? printf("a>b\n") : printf("a<b\n");
    return 0;
}
```

Наприклад,  
якщо  
**a == 7, b == 4**  
тоді  
управління  
до другого  
аргументу

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
```

```
int main(){
    int a, b, max;
    printf("a = ");    scanf("%d", &a);
    printf("b = ");    scanf("%d", &b);
```

```
    if (a > b)
        max = a;
    else
        max = b;
    printf("max=%d\n", max);
```

```
    a > b ? max=a : max=b;
    printf("max=%d\n", max);
```

```
    max=(a > b) ? a : b;
    printf("max=%d\n", max);
    return 0;
}
```

## З двох чисел a і b вибрати максимальне

За допомогою if

За допомогою  
тернарної операції

За допомогою  
тернарної операції

Тернарний оператор можна використовувати замість аргументу функції printf.

```
#include "pch.h"
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(){
int x = 0;
printf("%d\n", ++x ? x++ : x++);
printf("%d\n", x);
return 0;
}
```



Оператор? може бути вкладеним.

Наприклад, знайдемо середнє з трьох чисел:

```
int a = 1, b = 2, c = 3;  
int mid;
```

```
mid = (a > b) ? (a > c) ? (c > b) ? c : b : a : (b > c) ?  
(c > a) ? c : a : b;
```

```
printf("%d", mid);
```

Поясніть вираз.



Дано $x$ . Обрахувати $y$ , якщо:	$y = \begin{cases} x^2 + 4x + 5, & \text{при } x \leq 2, \\ \frac{1}{x^2 + 4x + 5}, & \text{при } x > 2 \end{cases}$
Дано $x, y, z$ . Знайти: $\min\{(x + y) - 7, y + 2z\} - 4$ .	
Дано $x$ . Обрахувати $y$ , якщо:	$y = \begin{cases} x^2, & \text{при } -2 \leq x \leq 2, \\ 4, & \text{при } x < -2 \text{ и } x > 2 \end{cases}$
Дано $x$ . Обрахувати $y$ , якщо:	$y = \begin{cases} 0, & \text{при } x \leq 0 \\ x, & \text{при } 0 < x \leq 1 \\ x^4, & \text{при } x > 1 \end{cases}$

# Оператор switch

```
switch ( вираз )
```

```
{
```

```
  case знач.1 : оператор11;
```

```
              оператор1N;
```

```
              break;
```

```
  case знач.2 : оператор21;
```

```
              оператор2M;
```

```
              break;
```

```
  default :
```

```
              оператор1;
```

```
              операторK;
```

```
}
```

*оператор  
break  
може  
бути  
відсутнім*



Якщо **break** відсутній, тоді програмний код продовжує виконуватись, навіть, якщо далі знаходить наступна гілка **case**

```
int day;  
printf("Day = ");  
scanf("%d", &day);  
switch (day)
```

далі  
виконуються  
усі блоки case  
та default

```
{  
    case 1: printf("понеділок\n");  
    case 2: printf("вівторок\n");  
    case 3: printf("середа\n");  
    case 4: printf("четвер\n");  
    case 5: printf("п'ятниця\n");  
    case 6: printf("субота\n");  
    case 7: printf("неділя\n");  
    default: printf("помилка\n");  
}
```

Наприклад,  
якщо  
`day == 2`,  
тоді  
управління  
передається  
на case 2.

```
C:\Windows\system32\cmd.exe
```

```
Day = 3  
середа  
четвер  
п'ятниця  
субота  
неділя  
помилка
```

```
C:\Windows\system32\cmd.exe
```

```
Day = 5  
п'ятниця  
субота  
неділя  
помилка
```

```
C:\Windows\system32\cmd.exe
```

```
Day = 12  
помилка
```

Блок **case** можуть бути записані у будь-якому порядку.

Блок **default** може знаходитися і між блоками **case**.

```
int day;
printf("Day = ");
scanf("%d", &day);
switch (day)
{
    case 6: printf("суббота\n");
    case 1: printf("понеділок\n");
    default: printf("помилка\n");
    case 4: printf("четвер\n");
    case 7: printf("неділя\n");
    case 2: printf("вівторок\n");
    case 3: printf("середа\n");
    case 5: printf("п'ятниця\n");
}
```

далі виконуються  
усі блоки **case** та  
**default** у порядку  
запису в коді

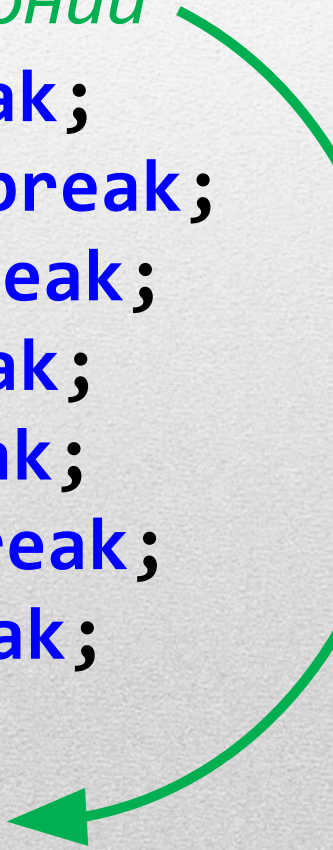
Наприклад,  
якщо  
**day == 1**,  
тоді  
управління  
передається  
на **case 1**.

```
C:\Windows\system32\cmd.exe
Day = 1
понеділок
помилка
четвер
неділя
вівторок
середа
п'ятниця
```

```
C:\Windows\system32\cmd.exe
Day = 12
помилка
четвер
неділя
вівторок
середа
п'ятниця
```

Додавши оператори **break** програмний код працюватиме коректно:

```
switch (day)      після останнього оператора  
{                break не потрібний  
    case 6: printf("субота\n"); break;  
    case 1: printf("понеділок\n"); break;  
    default: printf("помилка\n"); break;  
    case 4: printf("четвер\n"); break;  
    case 7: printf("неділя\n"); break;  
    case 2: printf("вівторок\n"); break;  
    case 3: printf("середа\n"); break;  
    case 5: printf("п'ятниця\n");  
}
```



Оператор **switch** часто використовується для створення меню в програмі.

В блоках **case** та **default** можуть розміщуватися будь-які оператори, у тому числі і **switch** та **if**.





## Приклади

Дано ціле число у діапазоні  $[1,12]$ , що означає місяць. Вивести на екран повідомлення про квартал. Наприклад; 3-перший квартал...

Дано ціле число у діапазоні  $[0,9]$ . Вивести на екран число прописом. Наприклад: 7- сім...

Приклад використання **switch** для  
реалізації розміщено за  
посиланням  
(натисніть для переходу):

[https://1drv.ms/w/s!AvLKc6r1gw0Vt  
Hwp63aYmz6BUJ0n](https://1drv.ms/w/s!AvLKc6r1gw0VtHwp63aYmz6BUJ0n)

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
unsigned day;
```

```
char const * name;
```

```
printf("day: ");
```

```
scanf("%u", &day);
```

```
name = day == 1 ? "Monday" :
```

```
day == 2 ? "Tuesday" :
```

```
day == 3 ? "Wednesday" :
```

```
day == 4 ? "Thursday" :
```

```
day == 5 ? "Friday" :
```

```
day == 6 ? "Saturday" :
```

```
day == 7 ? "Sunday" :
```

```
"Wrong day of week";
```

```
printf("%s", name);
```

```
return 0;
```

```
}
```

Оператор '?' в деяких випадках можна компактно записати як оператор switch: