

Лекція 3. Оператори розгалуження

Розгалуженням називається алгоритм, в якому вибирається один з декількох можливих варіантів обчислювального процесу. Кожен подібний шлях називається гілкою алгоритму.

Типи розгалужень

Повне розгалуження

Передбачає виконання дій як за істинності, так і за хибності умови.

Неповне розгалуження

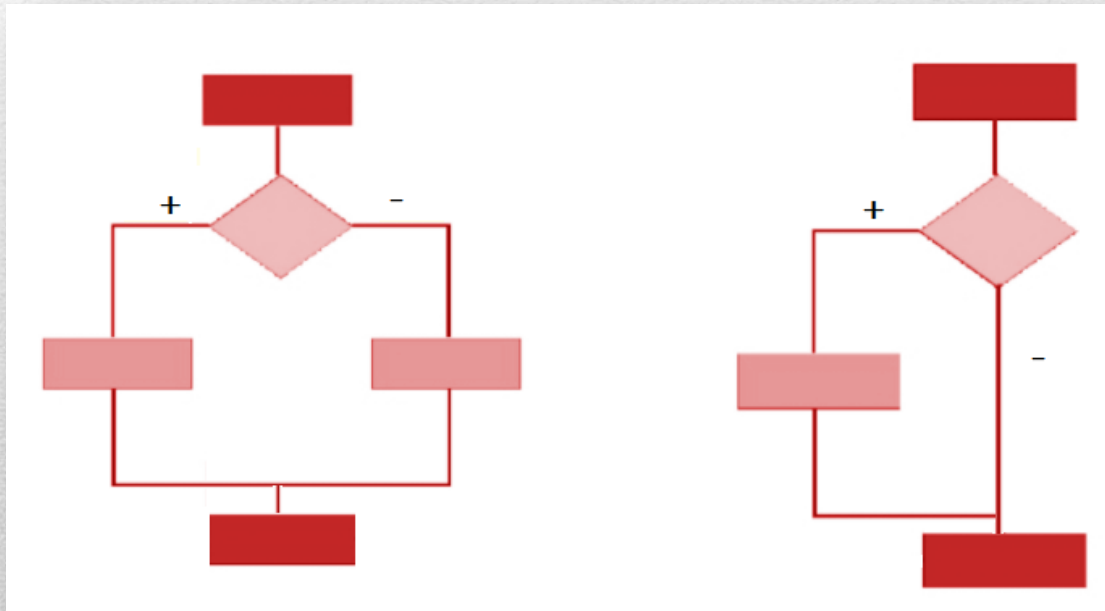
Передбачає виконання дій лише за істинності умови.

Вкладені розгалуження

- Це розгалуження всередині іншого розгалуження.
- Дозволяють створювати складні алгоритми для вирішення різноманітних завдань.

Графічне зображення

Розгалуження часто зображають за допомогою блок-схем. Це допомагає візуалізувати логіку алгоритму.



Повна форма

Неповна форма

Приклад

Перевірити, якщо число більше за нуль, то збільшити лічильник k на 1

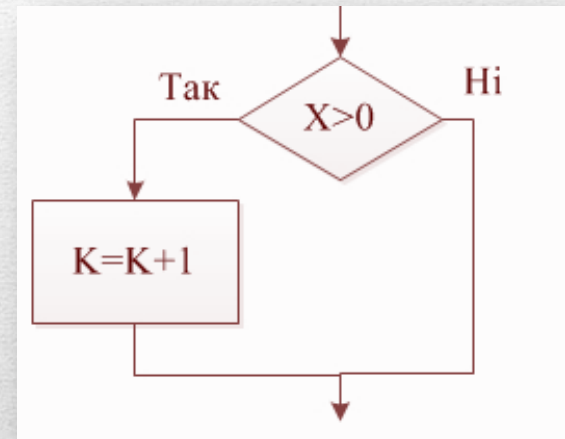
Алгоритм:

Вводимо число (x).

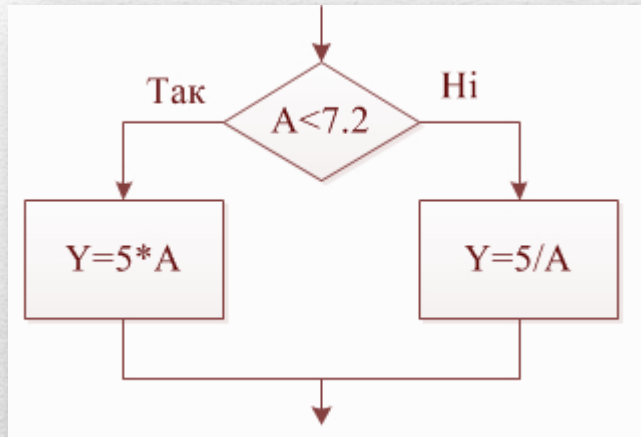
Перевіряємо, число більше 0.

Якщо так, k збільшуємо на 1.

Інакше, продовжуємо роботу алгоритму.



Приклад.



$$Y = \begin{cases} \frac{5}{A}, & A > 7,2 \\ 5 * A, & A < 7,2 \end{cases}$$

Якщо A більше 7.2 , то $Y = 5 / A$
Якщо A менше 7.2 , то $Y = 5 * A$

Приклади

Перевірити значення числа (додатне чи ні) і в залежності від результату вивести відповідне повідомлення.

Алгоритм:

Якщо число більше за нуль,
то вивести "Число додатне", інакше
вивести "Число від'ємне або нуль".

Алгоритм:

Якщо число більше нуля,
то вивести "Число додатне"
Інакше, Якщо число менше нуля,
то вивести "Число від'ємне"
Інакше, вивести "Число дорівнює
нулю"

Оператор if

Оператор **if** дозволяє перевірити умову і виконати оператор коду лише у випадку, якщо умова буде істинною.

Якщо умова не виконається, можна виконати інший оператор.

Синтаксис оператора **if**:

```
if ( умова ) оператор;
```

Наприклад:

```
if ( a > b ) printf("А більше ніж В");
```

умова *оператор*

Приклад:

```
if (a > b) printf("А більше ніж В");
```

Можна записувати і так:

```
if (a > b)  
    printf("А більше ніж В");
```

А так не правильно:

```
if (a > b) ;  
printf("А більше ніж В");
```

оператор, який виконається, якщо умова істинна

оператор, який до if ніякого відношення не має

Оператор **if** може містити частину **else**:

*виконається, якщо
умова істинна*



if (умова) оператор1;

else оператор2;



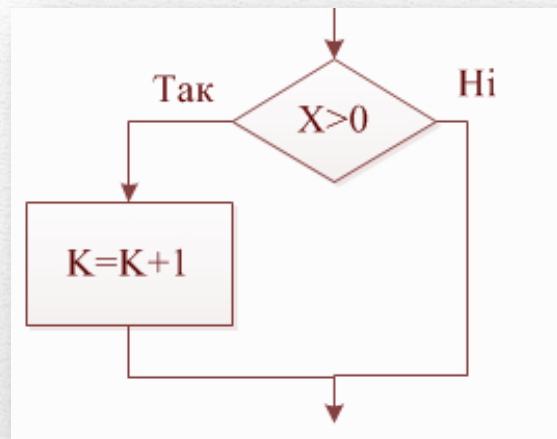
*виконається, якщо
умова хибна*

Приклад:

```
if (a > b) printf("А більше ніж В");  
else printf("А не більше за В");
```

Можна записувати і так:

```
if (a > b)  
    printf("А більше ніж В");  
else  
    printf("А не більше за В");
```



```
1  #include <stdio.h>
2
3  int main() {
4      int number, k = 0; // Ініціалізуємо змінну
5
6      printf("number: ");
7      scanf("%d", &number);
8
9      if (number > 0) k++;
10
11     printf("k = %d\n", k);
12
13     return 0;
14 }
```

```
number: -4
k = 0
```

```
number: 4
k = 1
```

```
number: 0
k = 0
```

При записі умов можна використовувати операції порівняння:

Операція	Пояснення	Приклад
$>$	більше	$a > b$
$<$	менше	$a < b$
$>=$	більше або дорівнює	$a >= b$
$<=$	менше або дорівнює	$a <= b$
$==$	дорівнює	$a == b$
$!=$	не дорівнює	$a != b$

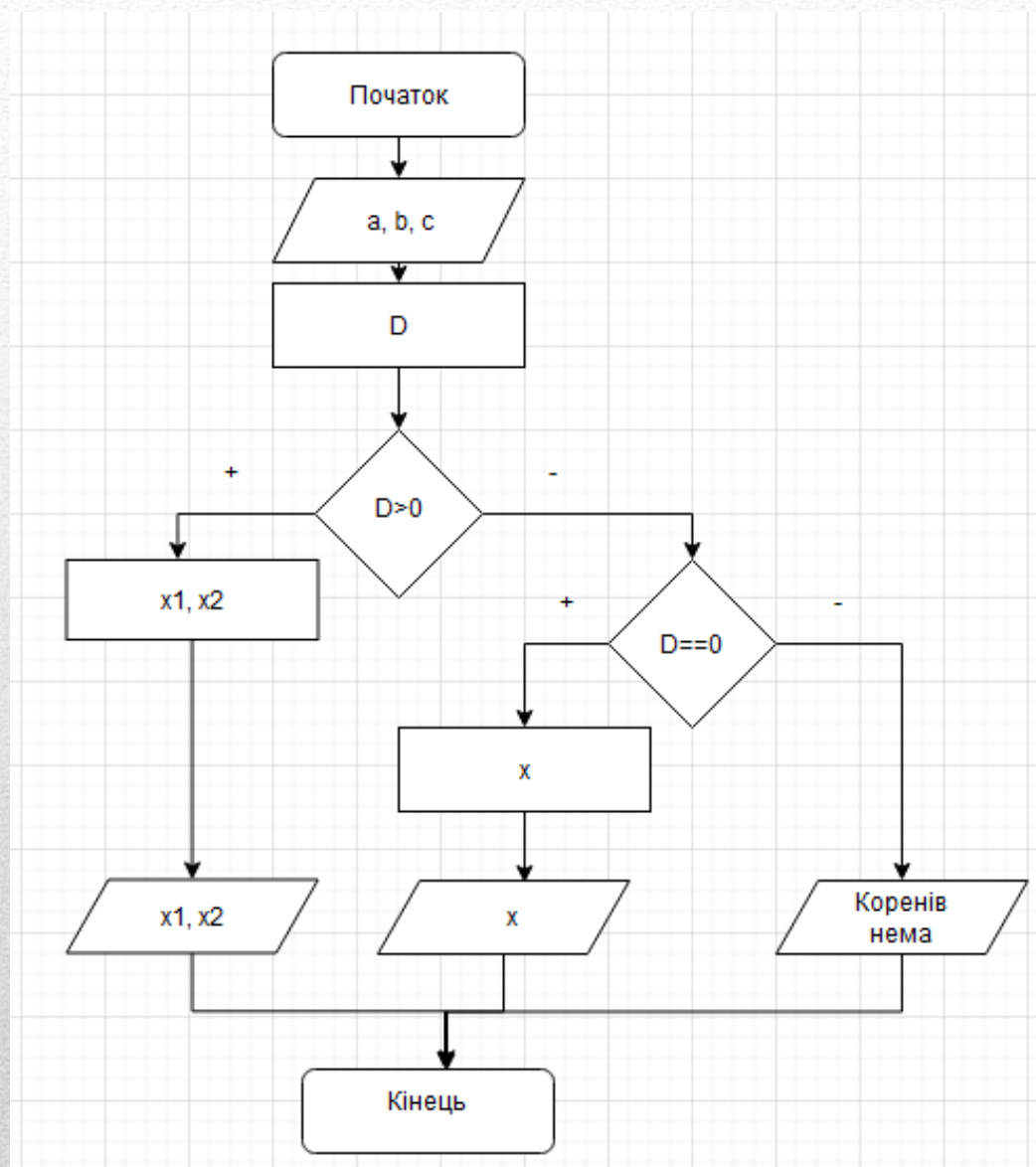
Приклад.

Завдання: знайти корені квадратного рівняння, якщо дано його коефіцієнти.

Розв'язання:

- Ввести **A, B, C**
- Обрахувати дискримінант
- Якщо **$D > 0$** , то обчислити **x_1** та **x_2** . Вивести **x_1** та **x_2** .
- Якщо **$D = 0$** , то обчислити **x** . Вивести **x** .
- Якщо **$D < 0$** , то вивести повідомлення “коренів немає”






```
float a, b, c, D, x1, x2, x;
printf("Введіть коефіцієнти a b c");
scanf("%f %f %f", &a, &b, &c);
D = b*b - 4 * a*c;
if (D > 0){
    x1 = (-b + sqrt(D)) / (2 * a);
    x2 = (-b - sqrt(D)) / (2 * a);
    printf("x1 = %f\nx2 = %f", x1, x2);
}
else
    if (D == 0){
        x = -b / (2 * a);
        printf("x = %f\n", x);
    }
    else printf("Коренів немає\n");
```

Операції порівняння дають в результаті значення:

1, якщо умова виконується

0, якщо умова не виконується

Умова	Результат
$3 > 5$	0
$1 \neq 9$	1
$3 \geq 3$	1

Приклад:

```
int x = 10, y = 15;  
printf("x>y -> %d\n", x > y);  
printf("x<y -> %d\n", x < y);
```

Результат:

x > y -> 0

x < y -> 1

Отже, результатом операції є

ціле число 0 або 1

Тому замість умови можна використовувати будь-який вираз:

```
if ( вираз ) оператор1;
```

```
else оператор2;
```

А значенням виразу є число (ціле або дробове)

У мові Сі під поняттям «**істина**» вважається будь-яке **не нульове** значення.

А поняттю «**хиба**» відповідає значення **нуль**.

Значення	Хиба чи істина
-8	істина
112	істина
0	хиба

Якщо в оператор **if** замість умови передавати вираз, то

виконається, якщо вираз дає не нульове значення


if (вираз) оператор1;

else оператор2;

виконається, якщо вираз дає нульове значення

Приклад:

*змінна X не нуль,
тому умова
вважається
істинною*

```
float x = 1;  
if (x)   
    printf("Умова істинна");  
else  
    printf("Умова хибна");
```

Логічні операції

Завдання: Дано числовий проміжок від **a** до **b**. Визначити чи входить число **c** у цей проміжок.



Математичний запис: $a < c < b$

Це має такий зміст: $a < c$ **i** $c < b$

В мові Сі для запису **i** використовується операція **&&**

Математичний запис: $a < c < b$

еквівалентний: $b > c > a$

У мові Сі таку умову можна записати так:

1) $a < c \ \&\& \ c < b$

2) $c > a \ \&\& \ c < b$

3) $a < c \ \&\& \ b > c$

4) $c > a \ \&\& \ b > c$

*всі ці вирази
еквівалентні*

```
if (a < c && b > c) printf("a < c < b");
```

Математичний запис: $a < b < c < d$

У мові Сі таку умову можна записати

так:

```
if (a < b && b < c && c < d)
    printf("a < c < b < d");
```

Ще один приклад:

```
int day;
printf("Введіть номер дня тижня:");
scanf("%d", &day);
if (day > 0 && day < 8)
    ...
else
    printf("Помилка. Такого дня не існує");
```

При записі складних умов можна використовувати **логічні операції**:

Операція	Пояснення	Приклад
&&	і	$a > b \ \&\& \ a < c$
	або	$x == 1 \ \ x == 2$
!	заперечення (унарна операція)	$! (x == 5)$

Кожна логічна операція має свою таблицю істинності.

Наприклад, для операції **&&** таблиця істинності

виглядає так:

&&	істина	хиба
істина	1	0
хиба	0	0

хиба означає **нуль**
істина – **не нуль**

Це означає, що:

Вираз	Результат
істина && істина	1
істина && хиба	0
хиба && істина	0
хиба && хиба	0

Операція **&&** (логічне **i**) дає 1 тільки у випадку, коли **обидва операнди є істиною**

Приклади виразів та їх значень:

Вираз	Результат
28 && 4	1
25 && 0	0
-5 && -3	1
13 && 5 && 3	1
1.5 && -5.5 && -7 && 3.5	1
3.6 && 8 && 0 && 3 && 1	0

Наприклад, для операції **||** таблиця істинності виглядає так:

 	істина	хиба
істина	1	1
хиба	1	0

Операція **||**
(логічне **або**)
дає 1, якщо
хоча б один
операнд є
істиною

Вираз	Результат
істина істина	1
істина хиба	1
хиба істина	1
хиба хиба	0

Приклад.

Написати фрагмент коду, який для дійсного x обчислює значення $f(x)$ та присвоює його дійсній змінній y .

$$f(x) = \begin{cases} x, & \text{якщо } 3 \leq x \leq 7, \\ x + 2, & \text{якщо } x > 7, \\ 2x, & \text{якщо } -5 < x < 3, \\ 0, & \text{якщо } x \leq -5. \end{cases}$$

Перепишемо формулу обчислення $f(x)$ у еквівалентному вигляді:

$$f(x) = \begin{cases} 0, & \text{якщо } x \leq -5, \\ 2x, & \text{якщо } -5 < x < 3, \\ x, & \text{якщо } 3 \leq x \leq 7, \\ x + 2, & \text{якщо } 7 < x. \end{cases}$$

Безпосередньо за формулою запишемо такий фрагмент коду:

```
if (x<=-5) y=0;  
if (x>-5 && x<3) y=2*x;  
if (x>=3 && x<=7) y=x;  
if (x>7) y=x+2;
```

Цей код є правильним, але неоптимальним за кількістю виконуваних операцій.

Отже, модифікуємо фрагмент коду наступним чином:

```
if (x<=-5) y=0;  
else if (x>-5 && x<3) y=2*x;  
else if (x>=3 && x<=7) y=x;  
else if (x>7) y=x+2;
```


Приклади:

$$f(x) = \begin{cases} x^3 + 2x - 4, & \text{якщо } -8 \leq x \leq 5; \\ x - 5, & \text{інакше} \end{cases}$$

```
float f, x;
```

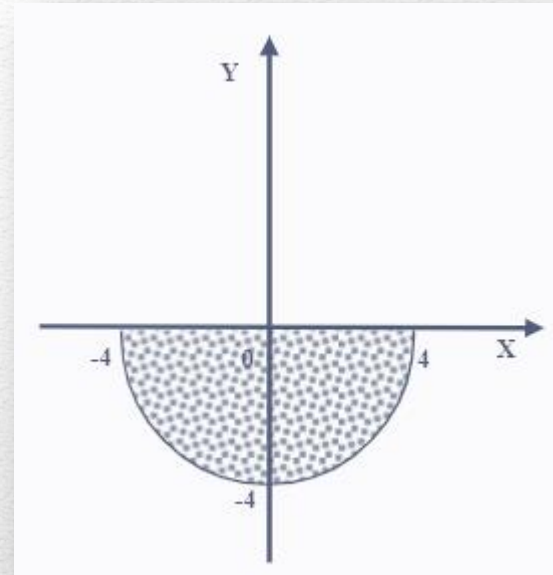
```
// ВВОДИМО x  
// ...
```

```
if ((x >= -8) && (x <= 5))  
    f = pow(x, 3) + 2 * x - 4;  
else  
    f = x - 5;
```

```
int x, y;
```

```
// ВВОДИМО x, y  
// (x1 - x)^2 + (y1 - y)^2 = R^2
```

```
if (y <= 0 && pow(x, 2) + pow(y, 2) <= 16)  
    printf("true\n");  
else  
    printf("no\n");
```



Дано три цілих числа a, b, c.

Розробити програму, яка знаходить мінімальне значення між цими числами.



```
int a, b, c;  
int min;  
a = 8;  
b = -5;  
c = 12;
```

```
min = a;  
if (min > b) min = b;  
if (min > c) min = c;
```

```
printf("min = %d \n", min );
```

```
int a, b, c;  
int min;  
a = 8;  
b = 12;  
c = -8;
```

```
if (a <= b && a <= c) min = a;  
else  
    if (b <= a && b <= c) min = b;  
else  
    min = c;
```

```
printf("min = %d \n", min );
```



$$f(x) = \begin{cases} 3x^2 - x, & \text{якщо } x < -6 \\ \sqrt{7-x}, & \text{якщо } -6 \leq x \leq 5 \\ 8x - 3, & \text{якщо } x > 5 \end{cases}$$

$$y = \begin{cases} a + bx^3, & \text{якщо } 1 \leq x < 2 \\ a \sin^3(x + b), & \text{якщо } 2 \leq x < 3 \\ \sqrt{|a + bx^3|}, & \text{якщо } 3 \leq x < 4 \\ a \lg|b + 2ax|, & \text{якщо } 4 \leq x < 5 \\ e^{a \sin x}, & \text{якщо } 5 \leq x \leq 6 \end{cases}$$