

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
кафедра інформаційних систем та мереж

Григорович В.Г.

Алгоритмізація та програмування

Розгалуження

Навчальний посібник

2020

Григорович Віктор Геннадійович

Алгоритмізація та програмування. Розгалуження. Навчальний посібник.

Дисципліна «Алгоритмізація та програмування» вивчається на першому курсі, – з цієї дисципліни розпочинається цикл предметів, що стосуються програмування та розробки програмного забезпечення.

В посібнику містяться теоретичні відомості, приклади, методичні вказівки з їх розв’язування, варіанти лабораторних завдань та питання і завдання з контролю знань з теми «Розгалуження».

Розглядаються наступні роботи лабораторного практикуму:

Лабораторна робота № 3.1.

Розгалуження, задане формулою: функція однієї змінної

Лабораторна робота № 3.2.

Розгалуження, задане формулою: функція з параметрами

Лабораторна робота № 3.3.

Розгалуження, задане графіком функції

Лабораторна робота № 3.4.

Розгалуження, задане плоскою фігурою

В посібнику наведено рекомендований набір лабораторних робіт та приклади їх виконання.

Відповідальний за випуск – Григорович В.Г.

Стислий зміст

Вступ.....	12
Тема 3. Розгалуження	13
Стисло та головне про розгалуження.....	13
Булеві операції та вирази.....	13
Команда розгалуження <code>if</code>	13
Умовна операція <code>?:</code>	15
Команда вибору варіанта <code>switch</code>	15
Теоретичні відомості.....	17
Булеві операції та вирази.....	17
Елементи блок-схем та UML-діаграм дій	19
Команди та конструкції керування порядком обчислень	20
Математичні відомості для виконання лабораторних завдань.....	31
Лабораторний практикум	36
Оформлення звіту про виконання лабораторних робіт	36
Лабораторна робота № 3.1. Розгалуження, задане формулою: функція однієї змінної....	38
Лабораторна робота № 3.2. Розгалуження, задане формулою: функція з параметрами ...	50
Лабораторна робота № 3.3. Розгалуження, задане графіком функції.....	62
Лабораторна робота № 3.4. Розгалуження, задане плоскою фігурою	80
Питання та завдання для контролю знань	97
Предметний покажчик	100
Література	101

Зміст

Вступ.....	12
Тема 3. Розгалуження	13
Стисло та головне про розгалуження.....	13
Булеві операції та вирази.....	13
Команда розгалуження if	13
Скорочена форма.....	13
Повна форма	14
Умовна операція ?:	15
Команда вибору варіанта switch	15
Теоретичні відомості.....	17
Булеві операції та вирази.....	17
Булевий тип (bool)	17
Вирази та операції відношення.....	17
Операції відношення.....	17
Вирази відношення	18
Перевірка на рівність / нерівність (збіг / відмінність)	18
Цілі числа в якості булевих значень.....	18
Логічні операції.....	18
Кон'юнкція та диз'юнкція.....	18
Заперечення	19
Елементи блок-схем та UML-діаграм дій	19
Команди та конструкції керування порядком обчислень	20
Порожня команда	20
Команда переходу	20
Блок команд. Операторні дужки.....	21
Команда розгалуження if	22
Скорочена форма.....	22
Повна форма	22
Типові помилки при організації розгалужень	23
Присвоєння замість порівняння на збіг	23
Ланцюжок порівнянь	23
Вкладений if замість складної умови	25
Використання команди if для організації розгалужень на багато гілок	26
Група if в скороченій формі	26

Вкладені if у повній формі	27
Типові помилки при організації розгалужень на багато гілок	27
Умовна операція ? :	28
Команда вибору варіанта switch	28
Команда break	31
Математичні відомості для виконання лабораторних завдань	31
Рівняння відрізка прямої, яка проходить через задані точки	31
Рівняння дуги кола заданого радіусу з центром в початку координат	31
Рівняння дуги кола заданого радіусу з центром в певній точці	32
Нерівності, які описують ліву, праву, верхню, нижню, внутрішню, зовнішню частини координатної площини, поділеної прямою лінією чи колом	32
Об'єднання та перетини плоских фігур та їх подання за допомогою логічних операцій	33
Лабораторний практикум	36
Оформлення звіту про виконання лабораторних робіт	36
Вимоги до оформлення звіту	36
Зразок оформлення звіту про виконання лабораторних робіт №№ 3.1–3.4	36
Лабораторна робота № 3.1. Розгалуження, задане формулою: функція однієї змінної	38
Мета роботи	38
Питання, які необхідно вивчити та пояснити на захисті	38
Оформлення звіту	38
Приклади розв'язання лабораторних завдань	38
Варіант 0	38
Умова завдання	38
Алгоритм	39
Текст програми	39
Варіанти лабораторних завдань	40
Варіант 1	41
Варіант 2	41
Варіант 3	41
Варіант 4	41
Варіант 5	41
Варіант 6	41
Варіант 7	42
Варіант 8	42

Варіант 9.....	42
Варіант 10.....	42
Варіант 11.....	42
Варіант 12.....	42
Варіант 13.....	43
Варіант 14.....	43
Варіант 15.....	43
Варіант 16.....	43
Варіант 17.....	43
Варіант 18.....	43
Варіант 19.....	44
Варіант 20.....	44
Варіант 21.....	44
Варіант 22.....	44
Варіант 23.....	44
Варіант 24.....	44
Варіант 25.....	45
Варіант 26.....	45
Варіант 27.....	45
Варіант 28.....	45
Варіант 29.....	45
Варіант 30.....	46
Варіант 31.....	46
Варіант 32.....	46
Варіант 33.....	46
Варіант 34.....	46
Варіант 35.....	47
Варіант 36.....	47
Варіант 37.....	47
Варіант 38.....	47
Варіант 39.....	47
Варіант 40.....	47
Варіант 41.....	48
Варіант 42.....	48
Варіант 43.....	48

Варіант 44.....	48
Варіант 45.....	48
Варіант 46.....	48
Варіант 47.....	49
Варіант 48.....	49
Варіант 49.....	49
Варіант 50.....	49
Лабораторна робота № 3.2. Розгалуження, задане формулою: функція з параметрами ...	50
Мета роботи	50
Питання, які необхідно вивчити та пояснити на захисті.....	50
Оформлення звіту.....	50
Приклади розв'язання лабораторних завдань	50
Варіант 0.....	50
Умова завдання.....	50
Алгоритм.....	51
Текст програми	51
Варіанти лабораторних завдань	52
Варіант 1.....	52
Варіант 2.....	53
Варіант 3.....	53
Варіант 4.....	53
Варіант 5.....	53
Варіант 6.....	53
Варіант 7.....	54
Варіант 8.....	54
Варіант 9.....	54
Варіант 10.....	54
Варіант 11.....	54
Варіант 12.....	55
Варіант 13.....	55
Варіант 14.....	55
Варіант 15.....	55
Варіант 16.....	55
Варіант 17.....	56
Варіант 18.....	56

Варіант 19.....	56
Варіант 20.....	56
Варіант 21.....	56
Варіант 22.....	57
Варіант 23.....	57
Варіант 24.....	57
Варіант 25.....	57
Варіант 26.....	57
Варіант 27.....	58
Варіант 28.....	58
Варіант 29.....	58
Варіант 30.....	58
Варіант 31.....	58
Варіант 32.....	59
Варіант 33.....	59
Варіант 34.....	59
Варіант 35.....	59
Варіант 36.....	59
Варіант 37.....	60
Варіант 38.....	60
Варіант 39.....	60
Варіант 40.....	60
Лабораторна робота № 3.3. Розгалуження, задане графіком функції.....	62
Мета роботи	62
Питання, які необхідно вивчити та пояснити на захисті.....	62
Оформлення звіту.....	62
Приклади розв'язання лабораторних завдань	62
Варіант 0.....	62
Умова завдання.....	62
Розв'язок (побудова математичної моделі)	63
Алгоритм.....	65
Текст програми	65
Варіанти лабораторних завдань	66
Варіант 1.....	66
Варіант 2.....	67

Варіант 3.....	67
Варіант 4.....	67
Варіант 5.....	68
Варіант 6.....	68
Варіант 7.....	68
Варіант 8.....	69
Варіант 9.....	69
Варіант 10.....	69
Варіант 11.....	70
Варіант 12.....	70
Варіант 13.....	70
Варіант 14.....	71
Варіант 15.....	71
Варіант 16.....	71
Варіант 17.....	72
Варіант 18.....	72
Варіант 19.....	72
Варіант 20.....	73
Варіант 21.....	73
Варіант 22.....	73
Варіант 23.....	74
Варіант 24.....	74
Варіант 25.....	74
Варіант 26.....	75
Варіант 27.....	75
Варіант 28.....	75
Варіант 29.....	76
Варіант 30.....	76
Варіант 31.....	76
Варіант 32.....	77
Варіант 33.....	77
Варіант 34.....	77
Варіант 35.....	78
Варіант 36.....	78
Варіант 37.....	78

Варіант 38.....	79
Варіант 39.....	79
Варіант 40.....	79
Лабораторна робота № 3.4. Розгалуження, задане плоскою фігурою	80
Мета роботи	80
Питання, які необхідно вивчити та пояснити на захисті.....	80
Оформлення звіту.....	80
Приклади розв'язання лабораторних завдань	80
Варіант 0.....	80
Умова завдання.....	80
Розв'язок (побудова математичної моделі)	81
Алгоритм.....	82
Текст програми	82
Варіанти лабораторних завдань	83
Варіант 1.....	83
Варіант 2.....	83
Варіант 3.....	84
Варіант 4.....	84
Варіант 5.....	84
Варіант 6.....	85
Варіант 7.....	85
Варіант 8.....	85
Варіант 9.....	86
Варіант 10.....	86
Варіант 11.....	86
Варіант 12.....	87
Варіант 13.....	87
Варіант 14.....	87
Варіант 15.....	88
Варіант 16.....	88
Варіант 17.....	88
Варіант 18.....	89
Варіант 19.....	89
Варіант 20.....	89
Варіант 21.....	90

Варіант 22.....	90
Варіант 23.....	90
Варіант 24.....	91
Варіант 25.....	91
Варіант 26.....	91
Варіант 27.....	92
Варіант 28.....	92
Варіант 29.....	92
Варіант 30.....	93
Варіант 31.....	93
Варіант 32.....	93
Варіант 33.....	94
Варіант 34.....	94
Варіант 35.....	94
Варіант 36.....	95
Варіант 37.....	95
Варіант 38.....	95
Варіант 39.....	96
Варіант 40.....	96
Питання та завдання для контролю знань	97
Предметний покажчик	100
Література	101

Вступ

Дисципліна «Алгоритмізація та програмування» вивчається на першому курсі, – з цієї дисципліни розпочинається цикл предметів, що стосуються програмування та розробки програмного забезпечення.

В посібнику містяться теоретичні відомості, приклади, методичні вказівки з їх розв’язування, варіанти лабораторних завдань та питання і завдання з контролю знань з теми «Розгалуження».

Тема 3. Розгалуження

Стисло та головне про розгалуження

Булеві операції та вирази

Булеві операції – це операції відношення та логічні операції.

Операції відношення: == != < > <= >=

Логічні операції: ! && ||

! Заперечення

A	!A
0	1
1	0

&& Кон'юнкція (логічне «і»)

A	B	A && B
0	0	0
0	1	0
1	0	0
1	1	1

|| Диз'юнкція (логічне «або»)

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

Команда розгалуження if

Скорочена форма

З однією командою:

```
if (/* умова */)
    /* команда */;
```

З блоком команд:

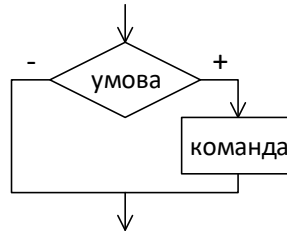
```
if (/* умова */)
{
```

```

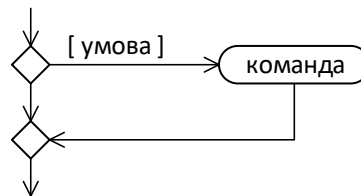
/*
команда;
...
команда;
*/
}

```

Позначення на блок-схемах:



Позначення на UML-activity diagram (діаграмах діяльності):



Повна форма

З однією командою в кожній гілці:

```

if (/* умова */)
    /* команда */;
else
    /* команда */;

```

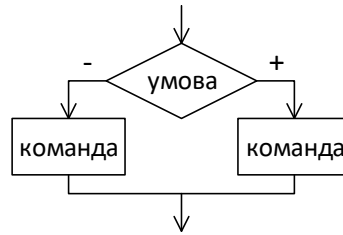
З блоком команд в кожній гілці:

```

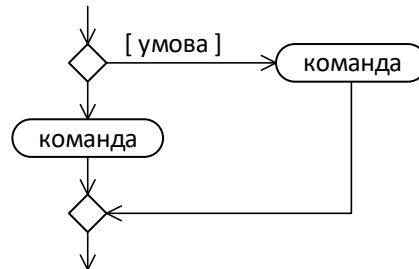
if (/* умова */)
{
    /*
    команда;
    ...
    команда;
    */
}
else
{
    /*
    команда;
    ...
    команда;
    */
}

```

Позначення на блок-схемах:



Позначення на UML-activity diagram (діаграмах діяльності):



Умовна операція ? :

Це – єдина тернарна операція (єдина операція, яка має три операнди).

Загальний вигляд умовного виразу (умовної команди):

вираз_1 ? вираз_2 : вираз_3

Дії комп'ютера при виконанні:

вираз_1 приводиться до типу **bool** та обчислюється його значення.

Якщо вираз_1 має значення **true** (1), то обчислюється значення виразу_2, яке буде результатом усього умовного виразу.

Якщо значення виразу_1 – **false** (0), то обчислюється значення виразу_3, яке стане результатом умовного виразу.

Команда вибору варіанта switch

Загальний вигляд:

```
switch ( вираз-перемикач )
{
    case константа:
    [ ... case константа: ]
        команда;
    [ ... команда; ]
    [ break; ]
    [ ... ]
    [ default:
        команда;
    [ ... команда; ] ]
}
```

Секція альтернатив `case` може містити кілька міток виду `case константа`: якщо значення *виразу-перемикача* збігається з однією з цих *констант*, то управління передається в цю секцію і виконуються її *команди*. *Константи* в мітках `case` мають бути унікальними

Якщо секція альтернатив `case` завершується командою `break`; то після виконання всіх *команд* цієї секції управління виходить за межі блоку `switch`.

Якщо секція альтернатив `case` не завершується командою `break`; то після виконання *команд* цієї секції управління «провалюється» в наступну секцію `case`.

Якщо не спрацювала жодна перевірка жодної секції альтернатив `case`, то управління передається в секцію `default` (якщо вона є).

Якщо не спрацювала жодна перевірка жодної секції альтернатив `case`, і немає секції `default`, то управління виходить за межі блоку `switch`.

Теоретичні відомості

Булеві операції та вирази

Булевий тип (**bool**)

- Зображення констант: **true false**
- Множина значень: {**false, true**}
- Розподіл пам'яті: 1 байт
- Допустимі операції:

Булеві операції – це операції відношення та логічні операції.

Операції відношення: **== != < > <= >=**

Логічні операції: **! && ||**

Розглянемо ці операції:

! Заперечення

A	!A
0	1
1	0

&& Кон'юнкція (логічне «і»)

A	B	A && B
0	0	0
0	1	0
1	0	0
1	1	1

|| Диз'юнкція (логічне «або»)

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

Вирази та операції відношення

Операції відношення

- <** менше
- >** більше
- <=** менше або дорівнює

`>=` більше або дорівнює

Вирази відношення

операнд_1 операція_відношення операнд_2

Операції відношення асоціативні (виконуються) зліва направо: \rightarrow .

Обидва операнди операції відношення мають бути цілих або вказівникових типів.

Зауваження: цілі типи включають до себе і булевий тип. Вказівникові типи будемо розглядати пізніше.

Результат виразу відношення буде булевого типу (**bool**).

Отримане значення буде **false** (0), якщо вираз хибний, та **true** (1), якщо вираз істинний.

Перевірка на рівність / нерівність (збіг / відмінність)

вираз_1 == вираз_2

вираз_1 != вираз_2

Операції перевірок на рівність чи нерівність мають менший пріоритет порівняно з операціями відношення.

Тип результату – булевий (**bool**).

Операція `==` повертає **true** (1), якщо операнди мають однакові значення, в іншому випадку – **false** (0).

Цілі числа в якості булевих значень

0 трактується як **false**.

Не нульове значення трактується як **true**.

Логічні операції

Кон'юнкція та диз'юнкція

`&&` кон'юнкція логічне «і»

`||` диз'юнкція логічне «або»

– мають менший пріоритет, ніж операції відношення:

```
if (x < y && y < z)
    cout << "x < z" << endl;
```

Логічні операції застосовуються до цілих значень, але вони не діють як звичайні арифметичні операції – вони трактують кожний операнд в термінах рівності нулеві.

В мові C результат логічних операцій 0 або 1 – типу `int`.

В мові C++ для них є тип `bool`.

вираз_1 `&&` вираз_2

– результат буде `true`, якщо обидва операнди мають значення `true`, та `false` у інших випадках.

вираз_1 `||` вираз_2

– результат буде `true`, якщо хоча б один операнд має значення `true`, та `false` у іншому випадку.

Заперечення

! логічне заперечення

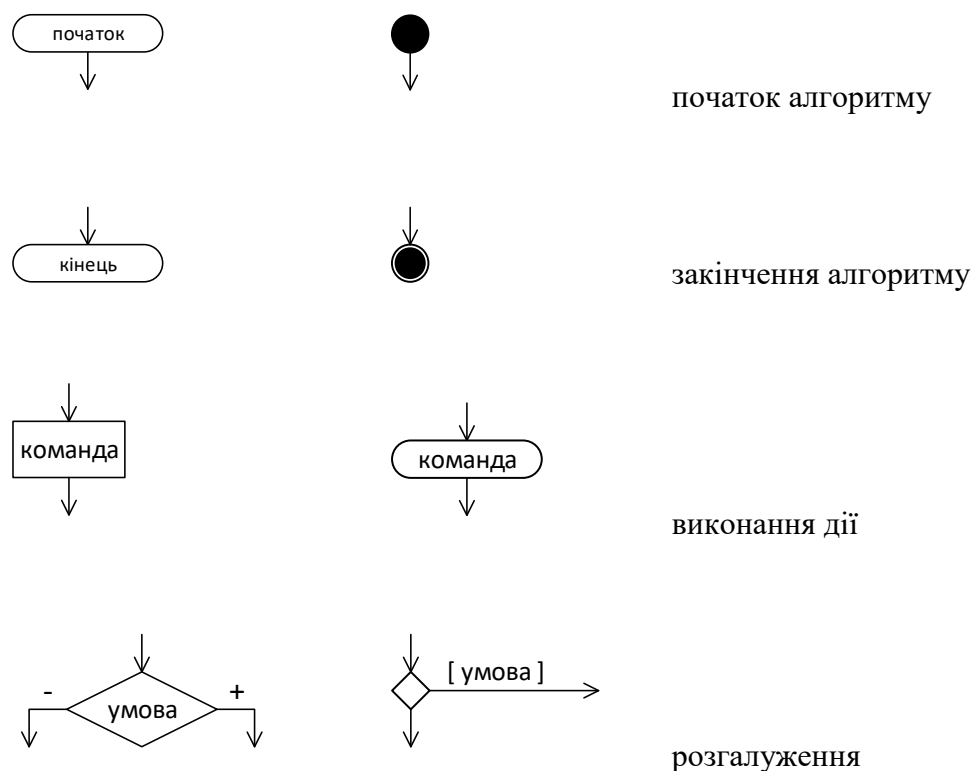
! логічний_вираз

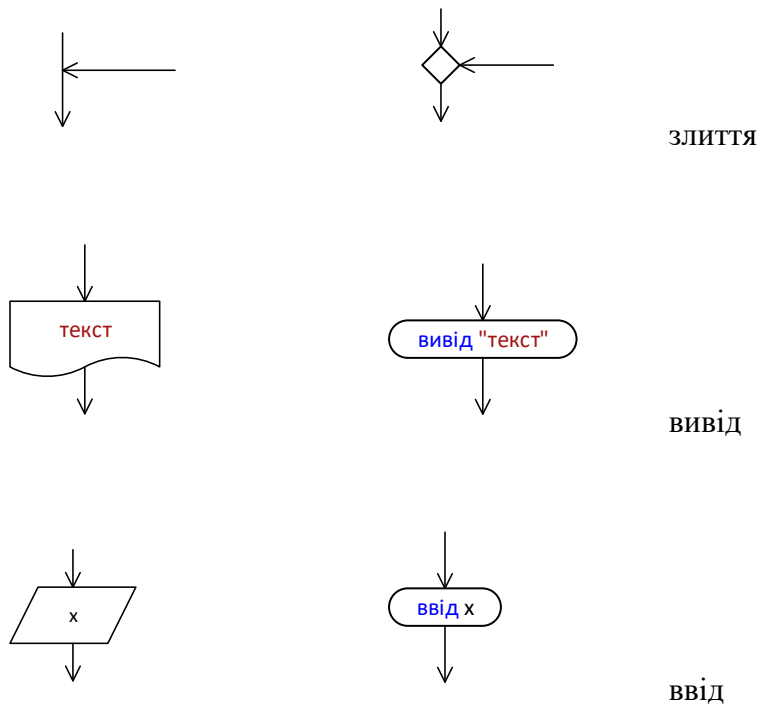
логічний_вираз – має бути цілого або вказівникового типів.

Перед обчисленням значення `!вираз` сам вираз конвертується до типу `bool`.

!вираз \Leftrightarrow (вираз == 0)

Елементи блок-схем та UML-діаграм дій





Команди та конструкції керування порядком обчислень

Тут будемо розглядати наступні команди:

- Порожня команда
- Команда переходу
- Блок команд. Операторні дужки
- Команда **if**:
 - Скорочена форма
 - Повна форма
- Типові помилки при організації розгалужень
- Умовна операція
- Команда **switch**

Порожня команда

;

Не виконує ніяких дій.

Використовується, коли синтаксис складеної команди вимагає, щоб до її складу входила певна команда, але для реалізації алгоритму не потрібно виконувати ніяких дій.

Команда переходу

goto ім'я_мітки;

ім'я_мітки має бути визначене в цій ж функції (тобто, помічена цією міткою команда має бути записана в цій ж функції, в якій є команда `goto`).

Вказана мітка – це «пункт призначення» для передачі управління поміченій команді. Лише одна команда в певній функції може бути помічена певною міткою (не можна однією міткою помітити кілька команд, «пункт призначення» для команди переходу має бути лише один).

Хороший стиль – уникати команди `goto`.

Дейкстра: «команда `goto` має бути заборонена».

Структурне програмування: замість команди `goto` слід використовувати команди `break continue return if while do for switch`.

Можна передавати управління із блоку команд назовні:

```
// можна
int i;
for (i = 0; i < 10; i++)
{
    if (i == 3)
        goto STOP;
}
STOP:
cout << "i = " << i << endl;
```

Не рекомендується передавати управління всередину блоку (хоча синтаксис мови це дозволяє):

```
// можна, але не рекомендується
int i = 4;
goto IN_FOR;
for (i = 0; i < 101; i++)
{
    IN_FOR:
        cout << i << endl;
}
```

Блок команд. Операторні дужки

Частина команд, записаних всередині фігурних дужок { } – це блок команд.

```
{
    команда;
    ...
    команда;
} // після блоку команд ; не ставимо
```

Блок може містити будь-які команди, у т.ч. і команди визначення / оголошення даних та інші блоки (тоді ці блоки називаються вкладеними).

Всі імена, оголошені в блоці (якщо вони не оголошені як `static`), існують і видимі лише в цьому блоці.

Після блоку команд крапку з комою ; не ставимо!

Команда розгалуження if

Скорочена форма

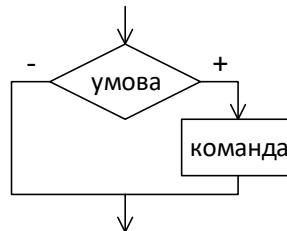
З однією командою:

```
if (/* умова */)
    /* команда */;
```

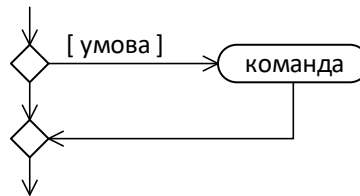
З блоком команд:

```
if (/* умова */)
{
    /*
    команда;
    ...
    команда;
    */
}
```

Позначення на блок-схемах:



Позначення на UML-activity diagram (діаграмах діяльності):



Повна форма

З однією командою в кожній гілці:

```
if (/* умова */)
    /* команда */;
else
    /* команда */;
```

З блоком команд в кожній гілці:

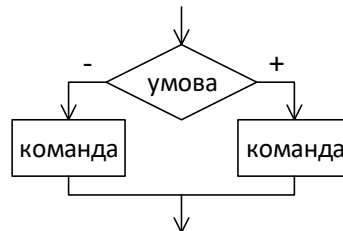
```
if (/* умова */)
{
    /*
    команда;
    ...
    команда;
    */
}
else
```

```

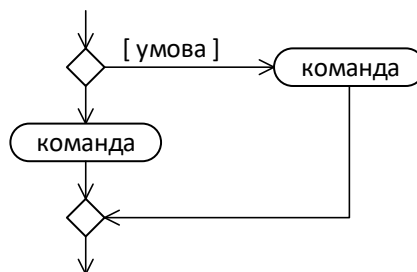
{
    /*
    команда;
    ...
    команда;
    */
}

```

Позначення на блок-схемах:



Позначення на UML-activity diagram (діаграмах діяльності):



Типові помилки при організації розгалужень

Присвоєння замість порівняння на збіг

= замість ==:

```

int x;
/* ... */
if (x = 1)
    cout << "x == 1" << endl;
else
    cout << "x != 1" << endl;

```

Присвоєння – це операція, результат якої = значенню, що було присвоєне.

Наведений приклад завжди буде виводити `x == 1`, бо `1` трактується як `true`.

Ланцюжок порівнянь

Припустимо, потрібно в якості умови розгалуження обчислити значення виразу $2 < x < 5$.

Якщо напишемо цю команду буквально:

```

if (2 < x < 5)
    cout << "x в діапазоні від 2 до 5" << endl;
else

```

```
cout << "x поза діапазоном від 2 до 5" << endl;
```

– то це приведе до помилки!

Припустимо, що $x = 1$:

```
int x = 1;
```

Тоді умова буде перевірятися так:

```
if ((2 < 1) < 5)
```

Перевірка $2 < 1$ дасть `false`, після приведення до цілого типу `false` буде тлумачитися як 0:

```
if (0 < 5)
```

Перевірка $0 < 5$ дає `true`, і програма виконає команду:

```
cout << "x в діапазоні від 2 до 5" << endl;
```

Тепер припустимо, що $x = 3$:

```
int x = 3;
```

Тоді умова буде перевірятися так:

```
if ((2 < 3) < 5)
```

Перевірка $2 < 3$ дасть `true`, після приведення до цілого типу `true` буде тлумачитися як 1:

```
if (1 < 5)
```

Перевірка $1 < 5$ дає `true`, і програма виконає команду:

```
cout << "x в діапазоні від 2 до 5" << endl;
```

Нарешті, припустимо, що $x = 7$:

```
int x = 7;
```

Тоді умова буде перевірятися так:

```
if ((2 < 7) < 5)
```

Перевірка $2 < 7$ дасть `true`, після приведення до цілого типу `true` буде тлумачитися як 1:

```
if (1 < 5)
```

Перевірка $1 < 5$ дає `true`, і програма виконає команду:

```
cout << "x в діапазоні від 2 до 5" << endl;
```

– бачимо, що у всіх випадках наша програма виведе "x в діапазоні від 2 до 5" !

Щоб виправити цю помилку, умову $2 < x < 5$ слід записати як складний булевий вираз:

```
if (2 < x && x < 5)
```

```
cout << "x в діапазоні від 2 до 5" << endl;
```

```
else
```

```
cout << "x поза діапазоном від 2 до 5" << endl;
```


Інший приклад:

```
int a = 2, b = 4, c = 3;
if (a < b < c)
    cout << "менше" << endl;
else
    cout << "не менше" << endl;
```

– порівняння дає результат **false** (0) або **true** (1), і виконується зліва направо:

```
if ((a < b) < c)
    ((2 < 4) < 3)
    =true
    = 1
    (1 < 3)
    =true
```

– тому цей приклад завжди буде виводити "менше".

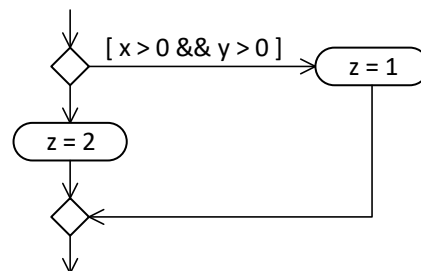
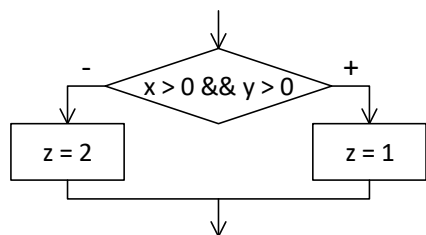
Щоб виправити помилку, слід написати так:

```
int a = 2, b = 4, c = 3;
if (a < b && b < c)
    cout << "менше" << endl;
else
    cout << "не менше" << endl;
```

Вкладений if замість складної умови

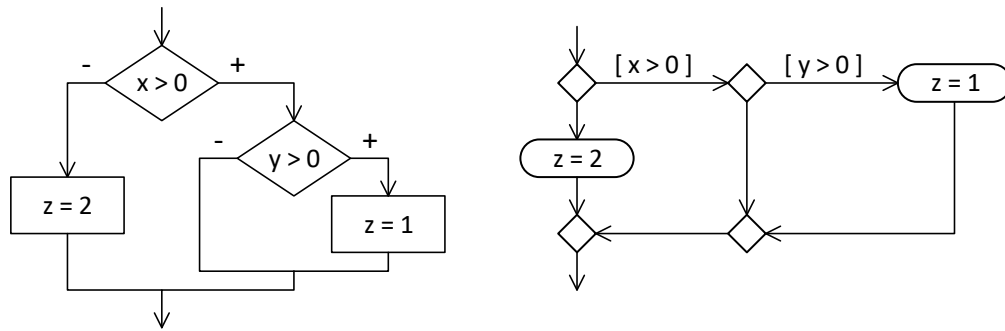
Не можна замість такої конструкції:

```
int x, y, z;
// ...
if (x > 0 && y > 0)
    z = 1;
else
    z = 2;
```



використовувати таку:

```
int x, y, z;
// ...
if (x > 0)
{
    if (y > 0)
        z = 1;
}
else
    z = 2;
```



Як видно з блок-схем та UML-activity діаграм, логіка цих розгалужень – різна!

Використання команди `if` для організації розгалужень на багато гілок

Група `if` в скороченій формі

Послідовні команди `if` у скороченій формі:

```

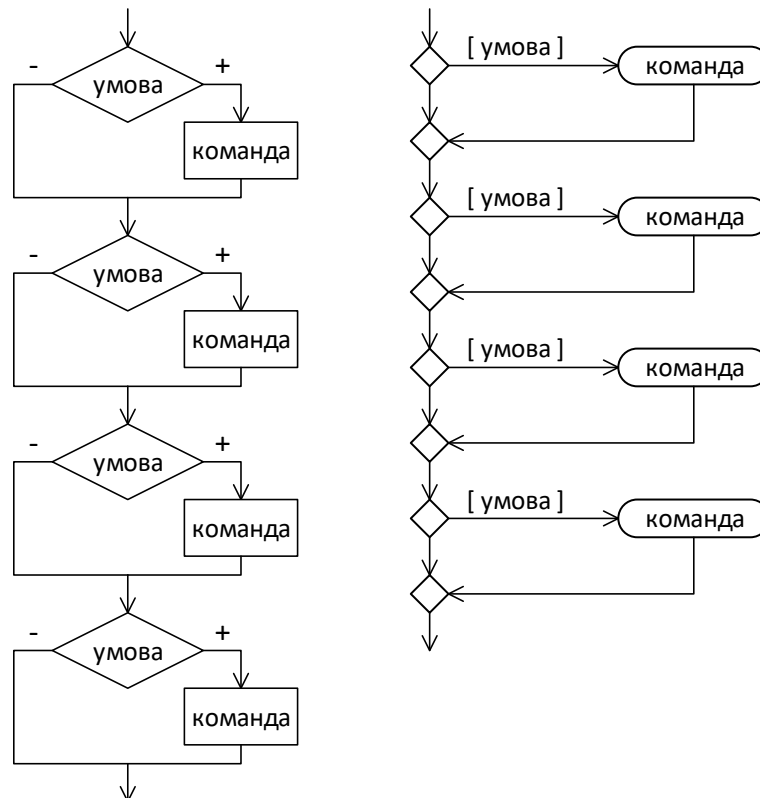
if (/* умова */)
    /* команда */;

if (/* умова */)
    /* команда */;

/* ... */

if (/* умова */)
    /* команда */;

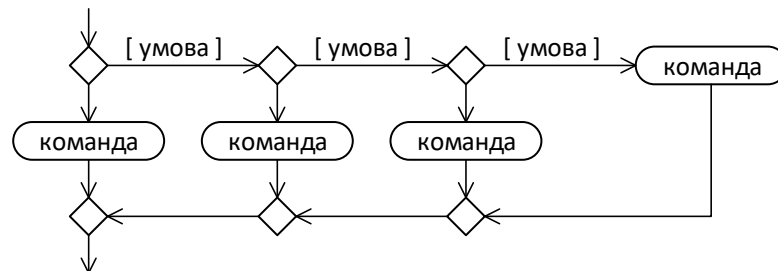
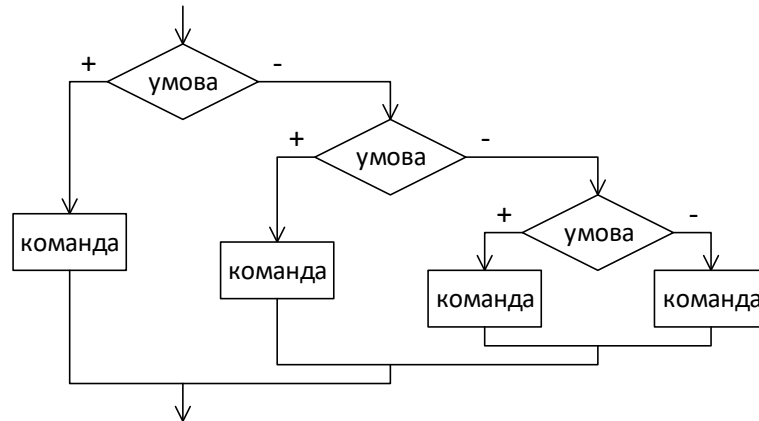
```



Для організації розгалуження на N гілок потрібно N послідовних команди розгалуження у скороченій формі.

Вкладені if у повній формі

```
if (/* умова */)
    /* команда */;
else
    if (/* умова */)
        /* команда */;
    else
        /* ... */
        if (/* умова */)
            /* команда */;
        else
            /* команда */;
```



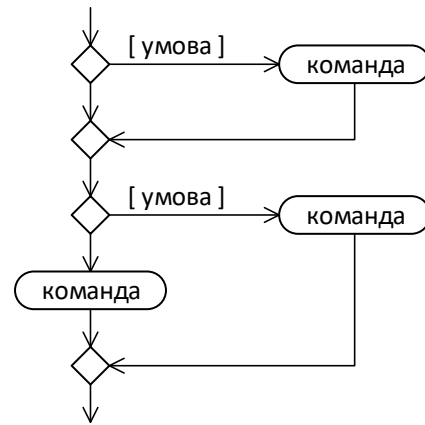
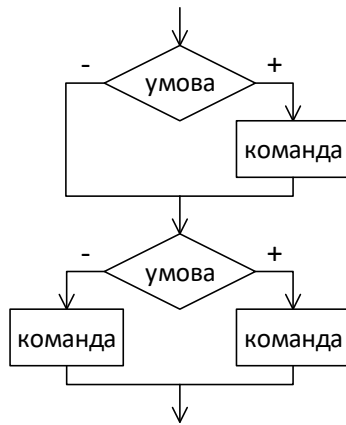
Для організації розгалуження на N гілок достатньо лише $(N-1)$ вкладених команди розгалуження у повній формі.

Типові помилки при організації розгалужень на багато гілок

Найчастіше – неправильне поєднання команд розгалуження:

```
if (/* умова */)
    /* команда */;

if (/* умова */)
    /* команда */;
else
    /* команда */;
```



Це – не розгалуження на три гілки!

Умовна операція ? :

Це – єдина тернарна операція (єдина операція, яка має три операнди).

Загальний вигляд умовного виразу (умовної команди):

вираз_1 ? вираз_2 : вираз_3

Дії комп'ютера при виконанні:

вираз_1 приводиться до типу **bool** та обчислюється його значення.

Якщо вираз_1 має значення **true** (1), то обчислюється значення виразу_2, яке буде результатом усього умовного виразу.

Якщо значення виразу_1 – **false** (0), то обчислюється значення виразу_3, яке стане результатом умовного виразу.

Приклад:

```
int i = 1, j = 2;
int k = i > j ? 3 : 4;
cout << k << endl;
```

Це – еквівалентно такому фрагменту:

```
int i = 1, j = 2, k;
if (i > j)
    k = 3;
else
    k = 4;
cout << k << endl;
```

Команда вибору варіанта switch

Дозволяє вибрати один із багатьох варіантів розгалуження:

Приклад:

```
int x;
```

```

switch (x)
{
case 1:
case 2:
    cout << "x = 1 or x = 2" << endl;
    break;
case 3:
    cout << "x = 3" << endl;
case 4:
    cout << "x = 4" << endl;
    break;
default:
    cout << "x has another value" << endl;
}

```

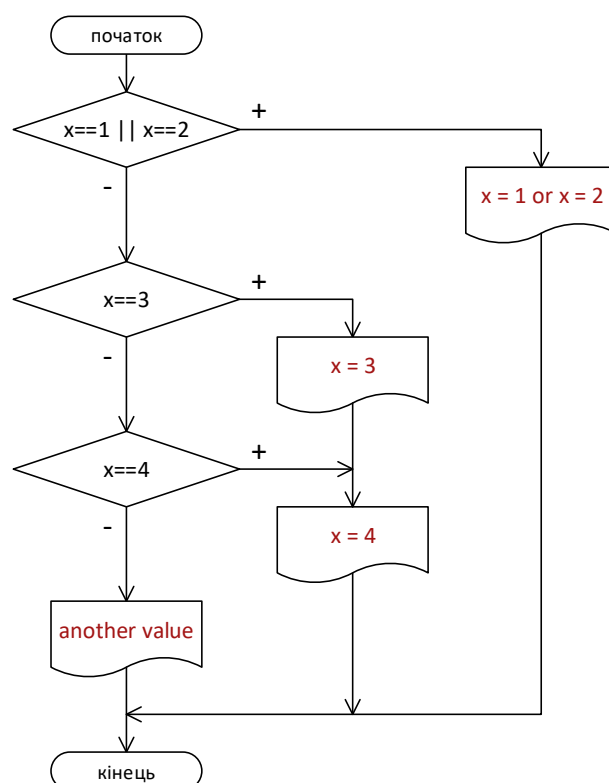
Цей фрагмент – еквівалентний наступному:

```

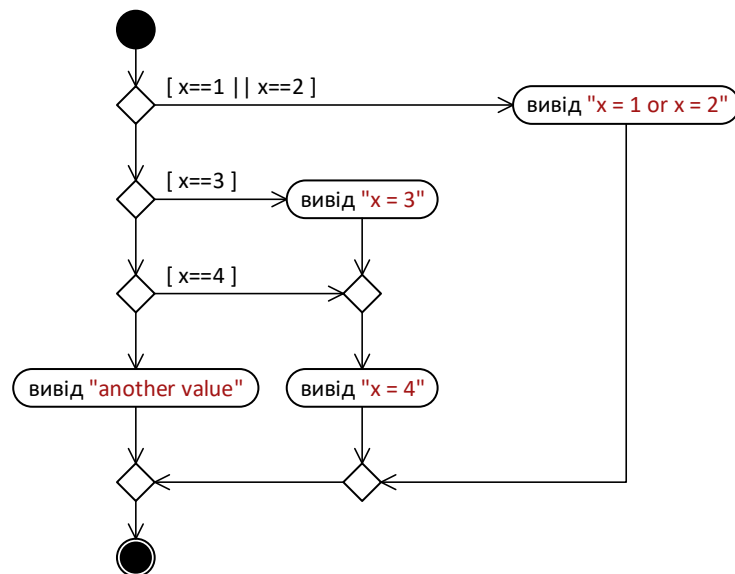
int x=3;
if (x == 1 || x == 2)
    cout << "x = 1 or x = 2" << endl;
else
    if (x == 3)
    {
        cout << "x = 3" << endl;
        goto JUMP_TO_4;
    }
    else
        if (x == 4)
        {
            JUMP_TO_4:
            cout << "x = 4" << endl;
        }
        else
            cout << "x has another value" << endl;

```

Блок-схема:



UML-діаграма дій:



Загальний вигляд:

```
switch ( вираз-перемикач )
{
    case константа:
    [ ... case константа: ]
        команда;
        [ ... команда; ]
        [ break; ]
    [ ... ]
    [ default:
        команда;
        [ ... команда; ] ]
}
```

Секція альтернатив **case** може містити кілька міток виду **case константа**: якщо значення *виразу-перемикача* збігається з однією з цих *констант*, то управління передається в цю секцію і виконуються її *команди*. *Константи* в мітках **case** мають бути унікальними

Якщо секція альтернатив **case** завершується командою **break**; то після виконання всіх *команд* цієї секції управління виходить за межі блоку **switch**.

Якщо секція альтернатив **case** не завершується командою **break**; то після виконання *команд* цієї секції управління «провалюється» в наступну секцію **case**.

Якщо не спрацювала жодна перевірка жодної секції альтернатив **case**, і немає секції **default** (якщо вона є).

Якщо не спрацювала жодна перевірка жодної секції альтернатив **case**, і немає секції **default**, то управління виходить за межі блоку **switch**.

Команда break

Загальний вигляд:

`break;`

Перериває виконання циклу або поточної альтернативи розгалуження чи вибору варіанта. Передає управління наступній команді після поточного структурного оператора.

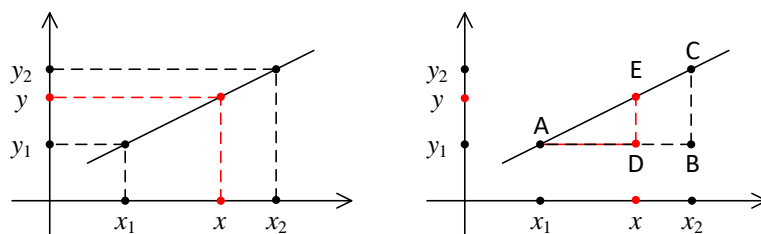
(використовується в командах циклів, `if`, `switch`)

Якщо в команді `switch` не поставити в кінці деякої альтернативи команду `break;` – то після виконання цієї альтернативи управління перейде до наступної альтернативи.

Математичні відомості для виконання лабораторних завдань

Рівняння відрізка прямої, яка проходить через задані точки

Рівняння прямої, яка проходить через точки (x_1, y_1) та (x_2, y_2) отримаємо з пропорції, яка описує подібність трикутників ABC та ADE:

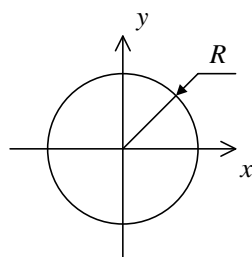


$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

звідси

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

Рівняння дуги кола заданого радіусу з центром в початку координат



Рівняння кола радіусу R з центром в точці $(0, 0)$: $x^2 + y^2 = R^2$.

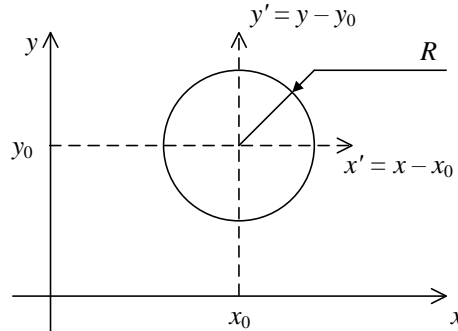
Звідси

$$y = \pm \sqrt{R^2 - x^2}$$

Нижній дузі кола відповідає знак «-».

Верхній дузі кола відповідає знак «+».

Рівняння дуги кола заданого радіусу з центром в певній точці



Рівняння кола радіусу R з центром в точці (x_0, y_0) : $(x-x_0)^2 + (y-y_0)^2 = R^2$.

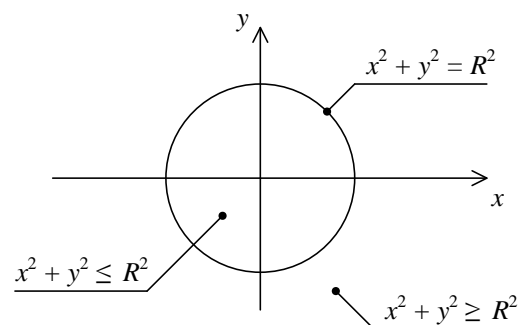
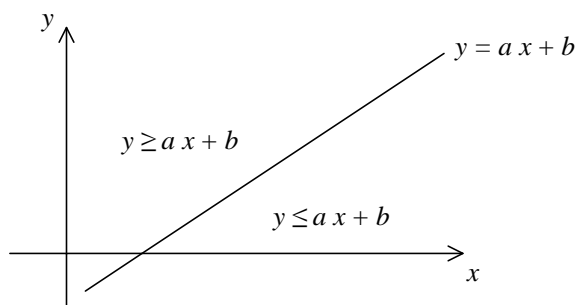
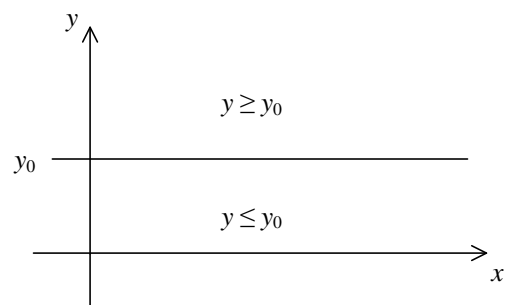
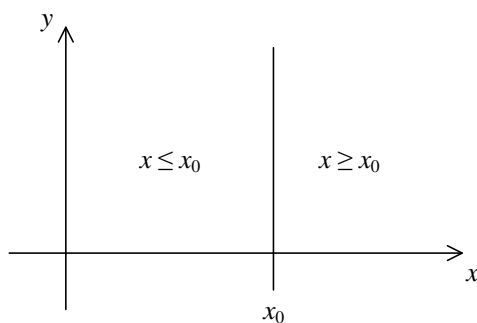
Звідси

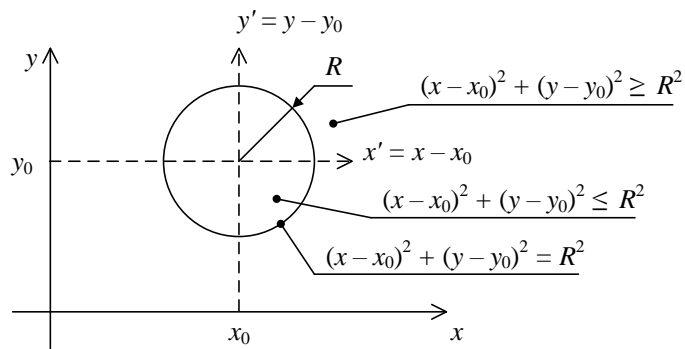
$$y = y_0 \pm \sqrt{R^2 - (x - x_0)^2}$$

Нижній дузі кола відповідає знак «-».

Верхній дузі кола відповідає знак «+».

Нерівності, які описують ліву, праву, верхню, нижню, внутрішню, зовнішню частини координатної площини, поділеної прямою лінією чи колом

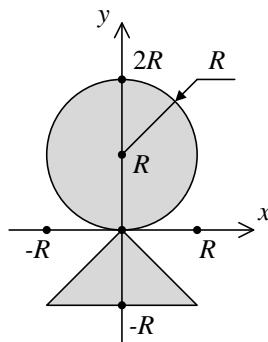




Об'єднання та перетини плоских фігур та їх подання за допомогою логічних операцій

Розглянемо приклад:

Написати програму, яка визначає чи попадає точка з заданими координатами в область, зафарбовану на рисунку сірим кольором. Результати роботи програми вивести у вигляді текстового повідомлення. Якщо у завданні є параметр R (параметри R_1 та R_2 , a , b) – то ці дані вводяться з клавіатури.



Розв'язок:

Загальна схема розв'язку:

```
double x; // вхідний аргумент
double y; // вхідний параметр

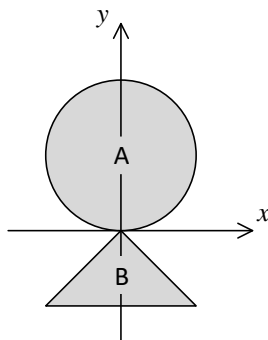
cout << "x = "; cin >> x;
cout << "y = "; cin >> y;

if ( /* умова */ )
    cout << "yes" << endl;
else
    cout << "no" << endl;
```

Ключовою командою програми – розв'язку завдання буде команда

```
if ( /* умова */ )
    cout << "yes" << endl;
else
    cout << "no" << endl;
```

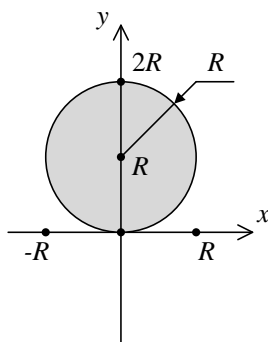
Як видно з рисунку, умову «точка (x, y) належить складній фігурі» можна записати у вигляді «точка (x, y) належить фігурі А або точка (x, y) належить фігурі В»:



```
if ( /* умова А */ || /* умова В */ )
    cout << "yes" << endl;
else
    cout << "no" << endl;
```

Тут

`/* умова А */` – це умова: точка (x, y) належить колу радіуса R з центром в точці $(0, R)$:



Тут $x_0 = 0$, $y_0 = R$

отже

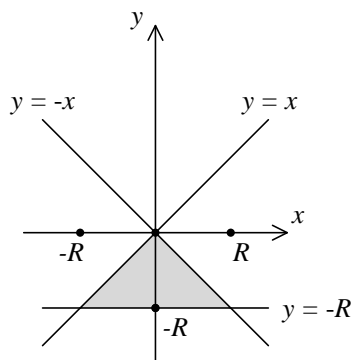
$$(x-x_0)^2 + (y-y_0)^2 \leq R^2$$

тому

`/* умова А */` – це умова: $(x-x_0)*(x-x_0) + (y-y_0)*(y-y_0) \leq R*R$

`/* умова В */` – це умова: точка (x, y) належить трикутнику, обмеженому прямими:

$$y = -R; y = x; y = -x$$



отже

$$y \geq -R; y \leq x; y \leq -x$$

тому

`/* умова В */` – це умова: `y >= -R && y <= x && y <= -x`

Таким чином, отримуємо:

```
if ((x-x0)*(x-x0) + (y-y0)*(y-y0) <= R*R ||  
    y >= -R && y <= x && y <= -x)  
    cout << "yes" << endl;  
else  
    cout << "no" << endl;
```

Лабораторний практикум

Оформлення звіту про виконання лабораторних робіт

Вимоги до оформлення звіту

Звіт про виконання лабораторних робіт №№ 3.1–3.4 повинен містити наступні елементи:

- 1) заголовок;
- 2) опис тих питань, які необхідно вивчити та описати в звіті;
- 3) умову завдання;
- 4) блок-схему алгоритму;
- 5) UML-діаграму дії;

Блок-схему та UML activity діаграму можна малювати «від руки». Якщо звіт подається у паперовому вигляді, то рисунки додаються до роздрукованого звіту. Якщо звіт подається в електронному вигляді (doc/docx файл), то рисунок слід сфотографувати та вставити цифрове фото як зображення у документ.

- 6) текст програми;

Текст програми має бути правильно відформатований: відступами і порожніми рядками слід відображати логічну структуру програми; програма повинна містити необхідні коментарі – про призначення підпрограм, змінних та параметрів – якщо їх імена не значущі, та про призначення окремих змістовних фрагментів програми. Текст програми слід подавати моноширинним шрифтом (Courier New або Consolas) розміром 10 пт. з одинарним міжрядковим інтервалом.

- 7) посилання на git-репозиторій з проектом;
- 8) висновки.

Зразок оформлення звіту про виконання лабораторних робіт

№№ 3.1–3.4

ЗВІТ

про виконання лабораторної роботи № < номер >

« назва теми лабораторної роботи »

з дисципліни

«Алгоритмізація та програмування»

студента(ки) групи КН-16

< Прізвище Ім'я По_батькові >

Питання, які необхідно вивчити та описати в звіті:

Якщо деякі питання вже описані в одному із попередньо захищених звітів, то їх повторно описувати необов'язково.

...

Умова завдання:

...

Блок-схема алгоритму:

...

UML-діаграма дії:

...

Текст програми:

...

Посилання на git-репозиторій з проектом:

...

Висновки:

...

Лабораторна робота № 3.1. Розгалуження, задане формулою: функція однієї змінної

Мета роботи

Навчитися створювати розгалужені програми.

Питання, які необхідно вивчити та пояснити на захисті

Якщо деякі питання вже описані в одному із попередньо захищених звітів, то їх повторно описувати необов'язково.

- 1) Операції відношення.
- 2) Логічні операції над булівськими величинами.
- 3) По-розрядні логічні операції над цілими величинами.
- 4) Команди C/C++: `if()`... ; `if()`...`else`...; `if()`...`else if`...; `(...)?...:...`;
- 5) Блок-схеми алгоритмів. Позначення розгалужень на блок-схемах.
- 6) UML-діаграми дії. Позначення розгалужень на UML-діаграмах дії.
- 7) Блок. Операторні дужки.
- 8) Команда C/C++: `switch(...)`...; команда `break`;
- 9) Стандартні функції, які отримують чи повертають булеві значення.

Оформлення звіту

Вимоги та зразок оформлення звіту наведені у вступі до лабораторного практикуму.

Приклади розв'язання лабораторних завдань

Варіант 0

Умова завдання

Написати програму для обчислення і виводу на екран значення змінної y – функції від аргументу x . x , y – дійсні числа.

Значення x ввести з клавіатури. В одній програмі реалізувати два способи: 1) використання лише команд розгалуження в скороченій формі та 2) використання лише команд розгалуження в повній формі – отримані результати повинні співпадати.

$$y = 2 \sin |x^3| + \begin{cases} \frac{1}{x} + \ln x^2, & 0 < x < 3 \\ \frac{7}{x} + \sqrt[3]{x}, & 3 \leq x < 4 \\ e^{1+\ln|x|}, & x \geq 4 \end{cases}$$

Алгоритм

1. Ввести значення x .
2. Обчислити значення A – функціонально сталої частини виразу.

Спосіб 1: розгалуження в скороченій формі

(обчислення значення B – функціонально змінної частини виразу).

3. Якщо справедлива умова₁, то
 - 5.1. B отримує значення виразу₁.
4. Якщо справедлива умова₂, то
 - 4.1. B отримує значення виразу₂.
5. Якщо справедлива умова₃, то
 - 5.1. B отримує значення виразу₃.
6. Обчислити значення $y = A + B$.
7. Вивести значення y .

Спосіб 2: розгалуження в повній формі

(обчислення значення B – функціонально змінної частини виразу).

8. Якщо справедлива умова₁, то
 - 8.1. B отримує значення виразу₁;
 - 8.2. інакше
 - 8.2.1. Якщо справедлива умова₂, то
 - 8.2.1.1. B отримує значення виразу₂;
 - 8.2.1.2. інакше B отримує значення виразу₃.
9. Обчислити значення $y = A + B$.
10. Вивести значення y .

Текст програми

```
// Lab_03_1.cpp
// < прізвище, ім'я автора >
// Лабораторна робота № 3.1
// Розгалуження, задане формулою: функція однієї змінної.
// Варіант 0.1

#include <iostream>
#include <cmath>
```

```

using namespace std;

int main()
{
    double x; // вхідний параметр
    double y; // результат обчислення виразу
    double A; // проміжний результат - функціонально стала частина виразу
    double B; // проміжний результат - функціонально змінна частина виразу

    cout << "x = "; cin >> x;

    A = 2*sin(fabs(x*x*x));

    // спосіб 1: розгалуження в скороченій формі
    if (0<x && x<3)
        B = 1/x + log(x*x);
    if (3<=x && x<4)
        B = 7/x + pow(x, 1./3.);
    if (x>=4)
        B = exp(1 + log(fabs(x)));

    y = A + B;

    cout << endl;
    cout << "1) y = " << y << endl;

    // спосіб 2: розгалуження в повній формі
    if (0<x && x<3)
        B = 1/x + log(x*x);
    else
        if (3<=x && x<4)
            B = 7/x + pow(x, 1./3.);
        else
            B = exp(1 + log(fabs(x)));

    y = A + B;

    cout << "2) y = " << y << endl;

    cin.get();
    return 0;
}

```

Варіанти лабораторних завдань

Написати програму для обчислення і виводу на екран значення змінної y – функції від аргументу x . x , y – дійсні числа.

Значення x ввести з клавіатури. В одній програмі реалізувати два способи: 1) використання лише команд розгалуження в скороченій формі та 2) використання лише команд розгалуження в повній формі – отримані результати повинні співпадати.

Варіант 1.

$$y = 1 + 9x + \begin{cases} \ln|\sin x| + x^7, & x \leq 0 \\ \operatorname{ctg} \frac{|x+1|}{2}, & 0 < x \leq 3 \\ 3x - x^5, & x > 3 \end{cases}$$

Варіант 2.

$$y = \frac{1}{x} + 4 - \begin{cases} 0,65x + 8, & x < 1 \\ \operatorname{arctg} \frac{x+6,1}{2} + e^x, & 1 \leq x < 5 \\ \sqrt{1+\sqrt{x}}, & x \geq 5 \end{cases}$$

Варіант 3.

$$y = \frac{2}{x} + |x| + \begin{cases} 1 + 4x^2, & x < 0 \\ (e^x + |x|)^2, & 0 \leq x \leq 2 \\ 5 \sin(x^2 + 1), & x > 2 \end{cases}$$

Варіант 4.

$$y = 1 + x + \begin{cases} e^{\ln(2+2x)+2x}, & x \leq 4 \\ \operatorname{ctg} \frac{1+x}{9} + 8x, & 4 < x \leq 7 \\ 1 - 7x + x^2 - 2x^3, & x > 7 \end{cases}$$

Варіант 5.

$$y = \frac{1}{|x+2|} + 1 - \begin{cases} 7x^2 + x - 8, & x < 1 \\ \operatorname{ctg} \frac{x+4}{\sqrt{11}} + 3, & 1 \leq x \leq 4 \\ \sqrt{1+|\cos^3 x|}, & x > 4 \end{cases}$$

Варіант 6.

$$y = 5e^{3x} - \begin{cases} 3 + \sin|x|, & x \leq -1 \\ 2e^{\frac{x}{4}-1}, & -1 < x \leq 3 \\ 7 - \sqrt{2}x^3, & x > 3 \end{cases}$$

Варіант 7.

$$y = x^2 \sin \frac{x}{2} + \begin{cases} \operatorname{arctg} e^x, & x \leq -5 \\ 1 + \frac{x^3}{4}, & -5 < x \leq 0 \\ \ln|x| - \frac{x}{5}, & x > 0 \end{cases}$$

Варіант 8.

$$y = 2 + 6x + \begin{cases} \ln \cos x + x^5, & x \leq 0 \\ \operatorname{ctg} \frac{1 + \ln x}{3}, & 0 < x \leq 3 \\ 12x - x^8, & x > 3 \end{cases}$$

Варіант 9.

$$y = 2|x|^3 - \begin{cases} 5 \cos 18x, & x \leq -0,1 \\ \operatorname{arctg} \frac{x+2}{5}, & -0,1 < x < 1,2 \\ \operatorname{ctg} x + 18, & x \geq 1,2 \end{cases}$$

Варіант 10.

$$y = 4,95x^2 + \begin{cases} 4 + x^{-2}, & x < -3,5 \\ \operatorname{tg} \frac{3,5+x}{5}, & -3,5 \leq x < 1 \\ \sin 3x - \cos x, & x \geq 1 \end{cases}$$

Варіант 11.

$$y = 2|5-x| - \begin{cases} e^{|2+x|}, & x \leq -1 \\ \sin^2 \frac{1}{|2+x|}, & -1 < x < 1 \\ \frac{\cos^2 x}{1+|\sin x|}, & x \geq 1 \end{cases}$$

Варіант 12.

$$y = \frac{2+x}{x^2} + 1 + \begin{cases} x^3 - 2x^4, & x < 0 \\ (|x| + e^x)^3, & 0 \leq x \leq 2 \\ 4 \cos(x^2 - 2), & x > 2 \end{cases}$$

Варіант 13.

$$y = 2|x-5| - \begin{cases} \frac{\sin^2 x}{1+|\cos x|}, & x < -1 \\ \cos^2 \frac{1}{|x+2|}, & -1 \leq x \leq 1 \\ \ln|x+2|, & x > 1 \end{cases}$$

Варіант 14.

$$y = |4x-1| + \begin{cases} x^7 - 2x, & x < 0 \\ \operatorname{arctg} \frac{e^x + 1}{8}, & 0 \leq x < 3 \\ x^4 + e^{x^2+3}, & x \geq 3 \end{cases}$$

Варіант 15.

$$y = x^3 + 2 + \begin{cases} 5x^8 + x^6 - x^2 + 3, & x < 4 \\ \operatorname{arctg} \left| \frac{x+3}{2} \right| + 7x, & 4 \leq x < 7 \\ \lg(2x + e^{5x+5}), & x \geq 7 \end{cases}$$

Варіант 16.

$$y = 5e^{3x} - \begin{cases} \sqrt{2}x^3 - 7, & x < -1 \\ 2\lg\left(1 - \frac{x}{4}\right), & -1 \leq x < 3 \\ \cos|x| + 3, & x \geq 3 \end{cases}$$

Варіант 17.

$$y = x^2 + 1 + \begin{cases} 4x^7 - x^5 + x^3 - 2, & x < 4 \\ \operatorname{arctg} \frac{|x|+1}{2} + 8,3x, & 4 \leq x < 7 \\ \ln|2x + e^{4x+1}|, & x \geq 7 \end{cases}$$

Варіант 18.

$$y = 13,5 - 2x - \begin{cases} e^{0,4+x}, & x \leq -1 \\ 1 - \sin^2 x, & -1 < x < 1 \\ \frac{\cos x}{1 + \sin^2 x}, & x \geq 1 \end{cases}$$

Варіант 19.

$$y = 2x - 13,5 - \begin{cases} \frac{\sin x}{1 + \cos^2 x}, & x < -1 \\ \cos^2 \sin^2 x - 1, & -1 \leq x \leq 1 \\ \lg(x + 0,4), & x > 1 \end{cases}$$

Варіант 20.

$$y = \operatorname{tg} x + 1 - \begin{cases} e^{\frac{x}{[1-x]}}, & x \leq -5 \\ 3,2 + \lg(1,8x^2), & -5 < x \leq 3 \\ |x|\sqrt{x}, & x > 3 \end{cases}$$

Варіант 21.

$$y = 2 + \frac{1}{|1+x|} - \begin{cases} \sqrt{|\cos x| + 13}, & x < 1 \\ 3 + \operatorname{ctg} \frac{4+x}{\sqrt{2}}, & 1 \leq x \leq 5 \\ 8 + 0,7x, & x > 5 \end{cases}$$

Варіант 22.

$$y = |x^3| + \begin{cases} |2+x| + \sin^2 x, & x < -1 \\ \operatorname{arctg}(x^3 + 1) + 1, & -1 \leq x \leq 1 \\ e^{\cos x} + \lg\left(\frac{1}{x} + 1\right), & x > 1 \end{cases}$$

Варіант 23.

$$y = x^2 + \sin \frac{4x}{3} + \begin{cases} \operatorname{ctg} x, & x < -5 \\ 4 - \frac{x^2}{2}, & -5 \leq x \leq 0 \\ \lg x^2 - \frac{4x}{3}, & x > 0 \end{cases}$$

Варіант 24.

$$y = x + \begin{cases} \lg|\cos 5x| + e^{\frac{1}{x}+x}, & x \leq -1 \\ \sqrt{(2-x)^3} - \operatorname{tg} x, & -1 < x < -0,4 \\ \sin 5x - \sqrt{1-x}, & x \geq 0,4 \end{cases}$$

Варіант 25.

$$y = 8,1 + x^3 + \begin{cases} 1 - x^{-5}, & x < -3,5 \\ \operatorname{ctg}|x+1|, & -3,5 \leq x < 1 \\ \operatorname{arctg} 2x - \lg \frac{x}{2}, & x \geq 1 \end{cases}$$

Варіант 26.

$$y = 4,95x^2 + \begin{cases} \cos 3x - \sin x, & x \leq -3,5 \\ \operatorname{arctg} \frac{x+3,5}{5}, & -3,5 < x \leq 1 \\ x^{-2} + 4, & x > 1 \end{cases}$$

Варіант 27.

$$y = |9x^3 + 2| + \begin{cases} 3x^5 - x^3 + 2x - 1, & x < 4 \\ \operatorname{arctg} \frac{x-2}{3} +, & 4 \leq x < 7 \\ \lg(2x^{-1} + e^{3x+1}), & x \geq 7 \end{cases}$$

Варіант 28.

$$y = \frac{x^2}{2,1 + \sin|x|} + \begin{cases} \operatorname{ctg} e^x, & x \leq -5 \\ 2 - \frac{x^3}{|x|+1}, & -5 < x < 0 \\ \ln \sqrt{|x| - \frac{x^2}{2}}, & x \geq 0 \end{cases}$$

Варіант 29.

$$y = \frac{1}{x} + 4 - \begin{cases} 8 + 0,65x, & x \leq 1 \\ \ln x + \operatorname{ctg} \frac{6,1+x}{2}, & 1 < x \leq 5 \\ \sqrt{x + \sqrt{2}}, & x > 5 \end{cases}$$

Варіант 30.

$$y = x^2 \cos \frac{x}{2} + \begin{cases} \frac{x}{5} - e^{|x|} & , \quad x < -5 \\ \frac{x^3}{4} + 1 & , \quad -5 \leq x \leq 0 \\ \operatorname{ctg} \ln x & , \quad x > 0 \end{cases}$$

Варіант 31.

$$y = 2 |13 - x| - \begin{cases} e^{|x-5|} & , \quad x \leq -1 \\ \sin^2 x^3 - 1 & , \quad -1 < x < 1 \\ \frac{\cos |x|}{1 + \sin^2 x} & , \quad x \geq 1 \end{cases}$$

Варіант 32.

$$y = \frac{1}{x^3} + 1 - \begin{cases} 0,3x + 7 & , \quad x < 1 \\ \operatorname{arctg} \frac{x+2}{8,1} + e^x & , \quad 1 \leq x \leq 5,5 \\ \sqrt{1 + \sqrt{1 + \sqrt{x}}} & , \quad x > 5,5 \end{cases}$$

Варіант 33.

$$y = x + 1 + \begin{cases} 2x^4 + x^6 - x^8 & , \quad x < 4 \\ \operatorname{arctg} \frac{x+2}{3} + 8x & , \quad 4 \leq x < 7 \\ \lg (x^2 + e^{x^2+2}) & , \quad x \geq 7 \end{cases}$$

Варіант 34.

$$y = 2x^2 - \begin{cases} 5 \sin 3x & , \quad x \leq -0,1 \\ \operatorname{arctg} \frac{x+1}{4} & , \quad -0,1 < x < 1,2 \\ \operatorname{tg} x + 2 & , \quad x \geq 1,2 \end{cases}$$

Варіант 35.

$$y = 4x - 7 + \begin{cases} x^2 - 5x & , \quad x < 0 \\ \operatorname{arctg} \frac{x+4}{8} & , \quad 0 \leq x < 3 \\ x^3 + e^{2x+3} & , \quad x \geq 3 \end{cases}$$

Варіант 36.

$$y = \frac{1}{x^2} + 4 - \begin{cases} 0,3x^2 + 8x + 1 & , \quad x < 1 \\ \operatorname{arctg} \frac{x+6}{2} + e^x & , \quad 1 \leq x < 5 \\ \sqrt{1+\sqrt{x}} & , \quad x \geq 5 \end{cases}$$

Варіант 37.

$$y = 8,1x^3 + \begin{cases} \operatorname{tg} 2x - \operatorname{tg} \frac{x}{2} & , \quad x \leq -3,5 \\ \operatorname{arctg} |x+1| & , \quad -3,5 < x \leq 1 \\ x^{-5} - 1 & , \quad x > 1 \end{cases}$$

Варіант 38.

$$y = \sin x + \begin{cases} \sqrt{|1-x|} - \cos 8x & , \quad x < -1 \\ \operatorname{arctg} x - \sqrt{(4-x)^5} & , \quad -1 \leq x \leq 0,4 \\ e^{\frac{\sin x}{2}} + \ln(\sin x + 2) & , \quad x > 0,4 \end{cases}$$

Варіант 39.

$$y = 2|x^7| - \begin{cases} \operatorname{arctg} \ln(x+1) & , \quad x < -0,1 \\ \operatorname{ctg} \frac{1+x}{8} & , \quad -0,1 \leq x \leq 1,2 \\ 8 \cos 9x & , \quad x > 1,2 \end{cases}$$

Варіант 40.

$$y = 8,7x^3 + \begin{cases} 2 + x^{-4} & , \quad x < -3,5 \\ \operatorname{tg} \left| \frac{1+x}{2} \right| & , \quad -3,5 \leq x < 1 \\ \sin \frac{x^2}{2} - \cos 4x & , \quad x \geq 1 \end{cases}$$

Варіант 41.

$$y = |2 + 9x^3| + \begin{cases} e^{2x^{-1} + \ln(3x+1)} & , \quad x < 4 \\ \operatorname{ctg} \frac{2-x}{3} + 2x & , \quad 4 \leq x < 7 \\ 1 - 2x - x^3 + 3x^5 & , \quad x > 7 \end{cases}$$

Варіант 42.

$$y = 2|x-13| - \begin{cases} \frac{\sin|x|}{1+\cos^2 x} & , \quad x < -1 \\ \cos^2 x^3 - 1 & , \quad -1 \leq x \leq 1 \\ \ln|x-5| & , \quad x > 1 \end{cases}$$

Варіант 43.

$$y = x + \frac{2}{x^2 + 1} + \begin{cases} 4\sin(2-x) & , \quad x \leq 0 \\ (x + \ln x)^2 & , \quad 0 < x < 2 \\ 2x^2 - x & , \quad x \geq 2 \end{cases}$$

Варіант 44.

$$y = 1 + \frac{1}{x^3} - \begin{cases} 7 + 0,3x & , \quad x \leq 1 \\ \ln x + \operatorname{tg} \frac{2+x}{8,1} & , \quad 1 < x < 5,5 \\ \sqrt{1 + \sqrt{x + \sqrt{2}}} & , \quad x \geq 5,5 \end{cases}$$

Варіант 45.

$$y = \frac{x+2}{x^2} + 1 + \begin{cases} 4\sin(2-x^2) & , \quad x \leq 0 \\ (|x| + \ln x)^3 & , \quad 0 < x < 2 \\ 2x^4 - x^3 & , \quad x \geq 2 \end{cases}$$

Варіант 46.

$$y = \frac{x}{1,1 + \cos x} + \begin{cases} \frac{x}{7} + e^{x^2} & , \quad x < -5 \\ \frac{x^3}{8} + 3 & , \quad -5 \leq x < 0 \\ \operatorname{arctg}|x| & , \quad x \geq 0 \end{cases}$$

Варіант 47.

$$y = 4 \ln 2x - \begin{cases} 6 + \sin x & , \quad x \leq -1 \\ 2e^{x-4} & , \quad -1 < x \leq 3 \\ 3 - \sqrt{2}x^2 & , \quad x > 3 \end{cases}$$

Варіант 48.

$$y = \ln 1 + \begin{cases} |\cos 5x| + 1 & , \quad x < 0 \\ x^{\sin x} - \operatorname{arctg} \frac{x}{8} & , \quad 0 \leq x \leq 4 \\ \sqrt{3,5 + x^2} + \lg(x^2 + 1) & , \quad x > 4 \end{cases}$$

Варіант 49.

$$y = \ln 2 + \begin{cases} \cos^3 x + 7 & , \quad x < 0 \\ \sqrt{x+3} + \operatorname{arctg} x & , \quad 0 \leq x < 4 \\ \sqrt[3]{2+x^2} + \lg(x+2x^2) & , \quad x \geq 4 \end{cases}$$

Варіант 50.

$$y = 2 \sin |x^3| + \begin{cases} \frac{1}{x} + \ln x^2 & , \quad x < 3 \\ \frac{7}{x} + \sqrt[3]{x} & , \quad 3 \leq x < 4 \\ e^{1+\ln|x|} & , \quad x \geq 4 \end{cases}$$

Лабораторна робота № 3.2. Розгалуження, задане формулою: функція з параметрами

Мета роботи

Навчитися створювати розгалужені програми.

Питання, які необхідно вивчити та пояснити на захисті

Якщо деякі питання вже описані в одному із попередньо захищених звітів, то їх повторно описувати необов'язково.

- 1) Операції відношення.
- 2) Логічні операції над булівськими величинами.
- 3) По-розрядні логічні операції над цілими величинами.
- 4) Команди C/C++: `if()`... ; `if()`...`else`...; `if()`...`else if`...; `(...)?...:...`;
- 5) Блок-схеми алгоритмів. Позначення розгалужень на блок-схемах.
- 6) UML-діаграми дії. Позначення розгалужень на UML-діаграмах дії.
- 7) Блок. Операторні дужки.
- 8) Команда C/C++: `switch(...)`...; команда `break`;
- 9) Стандартні функції, які отримують чи повертають булеві значення.

Оформлення звіту

Вимоги та зразок оформлення звіту наведені у вступі до лабораторного практикуму.

Приклади розв'язання лабораторних завдань

Варіант 0

Умова завдання

Написати програму для обчислення і виводу на екран значення функції F від аргументу x . a, b, c – дійсні числа.

Значення a, b, c, x ввести з клавіатури. В одній програмі реалізувати два способи: 1) використання лише команд розгалуження в скороченій формі та 2) використання лише команд розгалуження в повній формі – отримані результати повинні співпадати.

$$F = \begin{cases} -\frac{2x-c}{bx-a} & \text{при } x < 0 \text{ і } b \neq 0 \\ \frac{x-a}{x-c} & \text{при } x > 0 \text{ і } b = 0 \\ -\frac{x}{c} + \frac{-c}{2x+1} & \text{в інших випадках} \end{cases}$$

Алгоритм

1. Ввести значення a .
2. Ввести значення b .
3. Ввести значення c .
4. Ввести значення x .

Спосіб 1: розгалуження в скороченій формі

5. Якщо справедлива умова₁, то
 - 5.1. F отримує значення виразу₁.
6. Якщо справедлива умова₂, то
 - 6.1. F отримує значення виразу₂.
7. Якщо не справедлива умова₁ і не справедлива умова₂, то
 - 7.1. F отримує значення виразу₃.
8. Вивести значення F .

Спосіб 2: розгалуження в повній формі

9. Якщо справедлива умова₁, то
 - 2.1. F отримує значення виразу₁;
 - 2.2. інакше
 - 9.2.1. Якщо справедлива умова₂, то
 - 9.2.1.1. F отримує значення виразу₂;
 - 9.2.1.2. інакше F отримує значення виразу₃.
10. Вивести значення F .

Текст програми

```
// Lab_03_2.cpp
// < прізвище, ім'я автора >
// Лабораторна робота № 3.2
// Розгалуження, задане формулою: функція з параметрами.
// Варіант 0.1

#include <iostream>

using namespace std;

int main()
```

```

{
    double x; // вхідний аргумент
    double a; // вхідний параметр
    double b; // вхідний параметр
    double c; // вхідний параметр
    double F; // результат обчислення виразу

    cout << "a = "; cin >> a;
    cout << "b = "; cin >> b;
    cout << "c = "; cin >> c;
    cout << "x = "; cin >> x;

    // спосіб 1: розгалуження в скороченій формі
    if (x<0 && b!=0)
        F = -(2*x-c)/(b*x-a);
    if (x>0 && b==0)
        F = (x-a)/(x-c);
    if (!(x<0 && b!=0) && !(x>0 && b==0))
        F = -x/c + (-c)/(2*x+1);

    cout << endl;
    cout << "1) F = " << F << endl;

    // спосіб 2: розгалуження в повній формі
    if (x<0 && b!=0)
        F = -(2*x-c)/(b*x-a);
    else
        if (x>0 && b==0)
            F = (x-a)/(x-c);
        else
            F = -x/c + (-c)/(2*x+1);

    cout << "2) F = " << F << endl;

    cin.get();
    return 0;
}

```

Варіанти лабораторних завдань

Написати програму для обчислення і виводу на екран значення функції F від аргументу x . a, b, c – дійсні числа.

Значення a, b, c, x ввести з клавіатури. В одній програмі реалізувати два способи: 1) використання лише команд розгалуження в скороченій формі та 2) використання лише команд розгалуження в повній формі – отримані результати повинні співпадати.

Варіант 1.

$$F = \begin{cases} ax^2 + b & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x-a}{x-c} & \text{при } x > 0 \quad i \quad b = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 2.

$$F = \begin{cases} \frac{1}{ax} - b & \text{при } x + 5 < 0 \quad i \quad c = 0 \\ \frac{x - a}{x} & \text{при } x + 5 > 0 \quad i \quad c \neq 0 \\ \frac{10x}{c - 4} & \text{в інших випадках} \end{cases}$$

Варіант 3.

$$F = \begin{cases} ax^2 + bx + c & \text{при } a < 0 \quad i \quad c \neq 0 \\ \frac{-a}{x - c} & \text{при } a > 0 \quad i \quad c = 0 \\ a(x + c) & \text{в інших випадках} \end{cases}$$

Варіант 4.

$$F = \begin{cases} -ax - c & \text{при } c < 0 \quad i \quad x \neq 0 \\ \frac{x - a}{-c} & \text{при } c > 0 \quad i \quad x = 0 \\ \frac{bx}{c - a} & \text{в інших випадках} \end{cases}$$

Варіант 5.

$$F = \begin{cases} a - \frac{x}{10 + b} & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x > 0 \quad i \quad b = 0 \\ 3x + \frac{2}{c} & \text{в інших випадках} \end{cases}$$

Варіант 6.

$$F = \begin{cases} ax^2 + b^2x & \text{при } c < 0 \quad i \quad b \neq 0 \\ \frac{x + a}{x + c} & \text{при } c > 0 \quad i \quad b = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 7.

$$F = \begin{cases} -ax^2 - b & \text{при } x < 5 \quad i \quad c \neq 0 \\ \frac{x-a}{x} & \text{при } x > 5 \quad i \quad c = 0 \\ \frac{-x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 8.

$$F = \begin{cases} -ax^2 & \text{при } c < 0 \quad i \quad a \neq 0 \\ \frac{a-x}{cx} & \text{при } c > 0 \quad i \quad a = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 9.

$$F = \begin{cases} ax^2 + b^2x & \text{при } a < 0 \quad i \quad x \neq 0 \\ x - \frac{a}{x-c} & \text{при } a > 0 \quad i \quad x = 0 \\ 1 + \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 10.

$$F = \begin{cases} ax^2 - bx + c & \text{при } x < 3 \quad i \quad b \neq 0 \\ \frac{x-a}{x-c} & \text{при } x > 3 \quad i \quad b = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 11.

$$F = \begin{cases} ax^2 + \frac{b}{c} & \text{при } x < 1 \quad i \quad c \neq 0 \\ \frac{x-a}{(x-c)^2} & \text{при } x > 1.5 \quad i \quad c = 0 \\ \frac{x^2}{c^2} & \text{в інших випадках} \end{cases}$$

Варіант 12.

$$F = \begin{cases} ax^2 + b^2 + c & \text{при } x < 0.6 \quad i \quad b + c \neq 0 \\ \frac{x-a}{x-c} & \text{при } x > 0.6 \quad i \quad b + c = 0 \\ \frac{x}{c} + \frac{x}{a} & \text{в інших випадках} \end{cases}$$

Варіант 13.

$$F = \begin{cases} ax^2 + b & \text{при } x - 1 < 0 \quad i \quad b - x \neq 0 \\ \frac{x-a}{x} & \text{при } x - 1 > 0 \quad i \quad b + x = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 14.

$$F = \begin{cases} -ax^2 - b & \text{при } x + c < 0 \quad i \quad a \neq 0 \\ \frac{x-a}{x-c} & \text{при } x + c > 0 \quad i \quad a = 0 \\ \frac{x}{c} + \frac{c}{x} & \text{в інших випадках} \end{cases}$$

Варіант 15.

$$F = \begin{cases} -ax^2 + b & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x}{x-c} + 5.5 & \text{при } x > 0 \quad i \quad b = 0 \\ \frac{x}{-c} & \text{в інших випадках} \end{cases}$$

Варіант 16.

$$F = \begin{cases} a(x+c)^2 - b & \text{при } x = 0 \quad i \quad b \neq 0 \\ \frac{x-a}{-c} & \text{при } x = 0 \quad i \quad b = 0 \\ a + \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 17.

$$F = \begin{cases} ax^2 - cx + b & \text{при } x + 10 < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x + 10 > 0 \quad i \quad b = 0 \\ \frac{-x}{a - c} & \text{в інших випадках} \end{cases}$$

Варіант 18.

$$F = \begin{cases} ax^2 - bx^2 & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x > 0 \quad i \quad b = 0 \\ \frac{x + 5}{c(x - 10)} & \text{в інших випадках} \end{cases}$$

Варіант 19.

$$F = \begin{cases} a(x + 7)^2 - b & \text{при } x < 5 \quad i \quad b \neq 0 \\ \frac{x - cd}{ax} & \text{при } x > 5 \quad i \quad b = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 20.

$$F = \begin{cases} -\frac{2x - c}{cx - a} & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x > 0 \quad i \quad b = 0 \\ -\frac{x}{c} + \frac{-c}{2x} & \text{в інших випадках} \end{cases}$$

Варіант 21.

$$F = \begin{cases} ax^2 + b & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x > 0 \quad i \quad b = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 22.

$$F = \begin{cases} \frac{1}{ax} - b & \text{при } x + 5 < 0 \quad i \quad c = 0 \\ \frac{x - a}{x} & \text{при } x + 5 > 0 \quad i \quad c \neq 0 \\ \frac{10x}{c - 4} & \text{в інших випадках} \end{cases}$$

Варіант 23.

$$F = \begin{cases} ax^2 + bx + c & \text{при } a < 0 \quad i \quad c \neq 0 \\ \frac{-a}{x - c} & \text{при } a > 0 \quad i \quad c = 0 \\ a(x + c) & \text{в інших випадках} \end{cases}$$

Варіант 24.

$$F = \begin{cases} -ax - c & \text{при } c < 0 \quad i \quad x \neq 0 \\ \frac{x - a}{-c} & \text{при } c > 0 \quad i \quad x = 0 \\ \frac{bx}{c - a} & \text{в інших випадках} \end{cases}$$

Варіант 25.

$$F = \begin{cases} a - \frac{x}{10 + b} & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x > 0 \quad i \quad b = 0 \\ 3x + \frac{2}{c} & \text{в інших випадках} \end{cases}$$

Варіант 26.

$$F = \begin{cases} ax^2 + b^2x & \text{при } c < 0 \quad i \quad b \neq 0 \\ \frac{x + a}{x + c} & \text{при } c > 0 \quad i \quad b = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 27.

$$F = \begin{cases} -ax^2 - b & \text{при } x < 5 \quad i \quad c \neq 0 \\ \frac{x-a}{x} & \text{при } x > 5 \quad i \quad c = 0 \\ \frac{-x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 28.

$$F = \begin{cases} -ax^2 & \text{при } c < 0 \quad i \quad a \neq 0 \\ \frac{a-x}{cx} & \text{при } c > 0 \quad i \quad a = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 29.

$$F = \begin{cases} ax^2 + b^2x & \text{при } a < 0 \quad i \quad x \neq 0 \\ x - \frac{a}{x-c} & \text{при } a > 0 \quad i \quad x = 0 \\ 1 + \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 30.

$$F = \begin{cases} ax^2 - bx + c & \text{при } x < 3 \quad i \quad b \neq 0 \\ \frac{x-a}{x-c} & \text{при } x > 3 \quad i \quad b = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 31.

$$F = \begin{cases} ax^2 + \frac{b}{c} & \text{при } x < 1 \quad i \quad c \neq 0 \\ \frac{x-a}{(x-c)^2} & \text{при } x > 1.5 \quad i \quad c = 0 \\ \frac{x^2}{c^2} & \text{в інших випадках} \end{cases}$$

Варіант 32.

$$F = \begin{cases} ax^2 + b^2 + c & \text{при } x < 0.6 \quad i \quad b + c \neq 0 \\ \frac{x-a}{x-c} & \text{при } x > 0.6 \quad i \quad b + c = 0 \\ \frac{x}{c} + \frac{x}{a} & \text{в інших випадках} \end{cases}$$

Варіант 33.

$$F = \begin{cases} ax^2 + b & \text{при } x - 1 < 0 \quad i \quad b - x \neq 0 \\ \frac{x-a}{x} & \text{при } x - 1 > 0 \quad i \quad b + x = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 34.

$$F = \begin{cases} -ax^2 - b & \text{при } x + c < 0 \quad i \quad a \neq 0 \\ \frac{x-a}{x-c} & \text{при } x + c > 0 \quad i \quad a = 0 \\ \frac{x}{c} + \frac{c}{x} & \text{в інших випадках} \end{cases}$$

Варіант 35.

$$F = \begin{cases} -ax^2 + b & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x}{x-c} + 5.5 & \text{при } x > 0 \quad i \quad b = 0 \\ \frac{x}{-c} & \text{в інших випадках} \end{cases}$$

Варіант 36.

$$F = \begin{cases} a(x+c)^2 - b & \text{при } x = 0 \quad i \quad b \neq 0 \\ \frac{x-a}{-c} & \text{при } x = 0 \quad i \quad b = 0 \\ a + \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 37.

$$F = \begin{cases} ax^2 - cx + b & \text{при } x + 10 < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x + 10 > 0 \quad i \quad b = 0 \\ \frac{-x}{a - c} & \text{в інших випадках} \end{cases}$$

Варіант 38.

$$F = \begin{cases} ax^2 - bx^2 & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x > 0 \quad i \quad b = 0 \\ \frac{x + 5}{c(x - 10)} & \text{в інших випадках} \end{cases}$$

Варіант 39.

$$F = \begin{cases} a(x + 7)^2 - b & \text{при } x < 5 \quad i \quad b \neq 0 \\ \frac{x - cd}{ax} & \text{при } x > 5 \quad i \quad b = 0 \\ \frac{x}{c} & \text{в інших випадках} \end{cases}$$

Варіант 40.

$$F = \begin{cases} -\frac{2x - c}{cx - a} & \text{при } x < 0 \quad i \quad b \neq 0 \\ \frac{x - a}{x - c} & \text{при } x > 0 \quad i \quad b = 0 \\ -\frac{x}{c} + \frac{-c}{2x} & \text{в інших випадках} \end{cases}$$

Лабораторна робота № 3.3. Розгалуження, задане графіком функції

Мета роботи

Навчитися описувати формулами функції, задані графіком. Навчитися створювати розгалужені програми.

Питання, які необхідно вивчити та пояснити на захисті

Якщо деякі питання вже описані в одному із попередньо захищених звітів, то їх повторно описувати необов'язково.

- 1) Операції відношення.
- 2) Логічні операції над булівськими величинами.
- 3) По-розрядні логічні операції над цілими величинами.
- 4) Команди C/C++: `if()`... ; `if()`...`else`...; `if()`...`elseif`...; `(...)?`...:; ;
- 5) Блок-схеми алгоритмів. Позначення розгалужень на блок-схемах.
- 6) UML-діаграми дії. Позначення розгалужень на UML-діаграмах дії.
- 7) Блок. Операторні дужки.
- 8) Команда C/C++: `switch()`...; команда `break`;
- 9) Стандартні функції, які отримують чи повертають булеві значення.
- 10) Рівняння відрізка прямої, яка проходить через задані точки.
- 11) Рівняння дуги кола заданого радіусу, яка проходить через задані точки.
- 12) Рівняння дуги параболи, яка проходить через задані точки.
- 13) Побудова математичної моделі розв'язку завдання свого варіанту.

Оформлення звіту

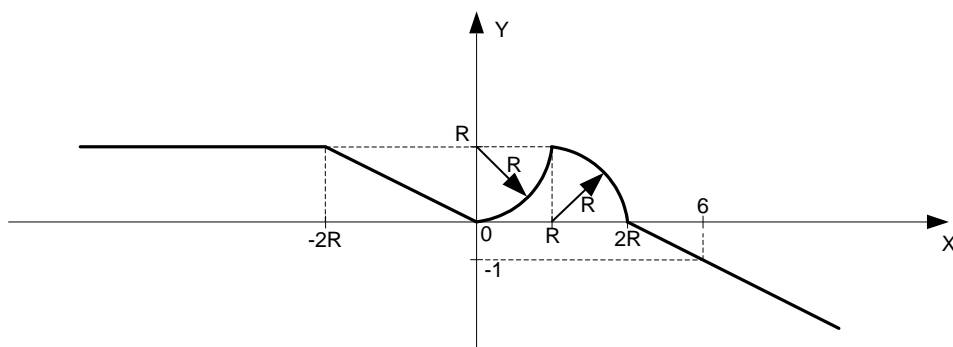
Вимоги та зразок оформлення звіту наведені у вступі до лабораторного практикуму.

Приклади розв'язання лабораторних завдань

Варіант 0

Умова завдання

Написати програму, яка за введеним значенням аргументу обчислює значення функції, заданої у вигляді графіка. Якщо у завданні є параметр R (параметри R₁ та R₂) – то ці дані вводяться з клавіатури.



Розв'язок (побудова математичної моделі)

Етап 1. Визначення кількості функціонально різних елементів графіка.

З графіка функції видно, що він складається із 5-ти функціонально різних елементів:

- 1) на інтервалі $-\infty < x \leq -2R$ графік функції $y = f(x)$ – це пряма лінія, паралельна осі Ox , яка проходить через точку $y = R$;
- 2) на інтервалі $-2R < x \leq 0$ графік функції $y = f(x)$ – це пряма лінія, яка проходить через точки $x_1 = -2R, y_1 = R$ та $x_2 = 0, y_2 = 0$;
- 3) на інтервалі $0 < x \leq R$ графік функції $y = f(x)$ – це фрагмент нижньої дуги кола радіусу R з центром в точці $x_0 = 0, y_0 = R$;
- 4) на інтервалі $R < x \leq 2R$ графік функції $y = f(x)$ – це фрагмент верхньої дуги кола радіусу R з центром в точці $x_0 = R, y_0 = 0$;
- 5) на інтервалі $2R < x < \infty$ графік функції $y = f(x)$ – це пряма лінія, яка проходить через точки $x_1 = 2R, y_1 = 0$ та $x_2 = 6, y_2 = -1$.

Етап 2. Виведення рівнянь для елементів графіка.

- 1) $-\infty < x \leq -2R$: $y = R$;
- 2) $-2R < x \leq 0$: точки $x_1 = -2R, y_1 = R$ та $x_2 = 0, y_2 = 0$.

Рівняння прямої, яка проходить через точки (x_1, y_1) та (x_2, y_2) отримаємо з пропорції:

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

звідси

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

Підставляючи значення $x_1 = -2R, y_1 = R$ та $x_2 = 0, y_2 = 0$, отримаємо:

$$y = R + \frac{0 - R}{0 + 2R}(x + 2R)$$

Скорочуючи, отримаємо:

$$y = R - \frac{1}{2}(x + 2R) = R - \frac{1}{2}x - R$$

Остаточню:

$$y = -\frac{1}{2}x$$

Отже, на інтервалі $-2R < x \leq 0$: $y = -\frac{1}{2}x$

- 3) $0 < x \leq R$: нижня дуга кола радіусу R з центром в точці $x_0 = 0, y_0 = R$.

Рівняння кола радіусу R з центром в точці x_0, y_0 : $(x-x_0)^2 + (y-y_0)^2 = R^2$.

Підставимо значення $x_0 = 0, y_0 = R$ та отримаємо: $x^2 + (y-R)^2 = R^2$.

Звідси: $y = R \pm \sqrt{R^2 - x^2}$.

Нижній дузі кола відповідає знак «-».

Отже, на інтервалі $0 < x \leq R$: $y = R - \sqrt{R^2 - x^2}$

- 4) $R < x \leq 2R$: верхня дуга кола радіусу R з центром в точці $x_0 = R, y_0 = 0$.

Рівняння кола радіусу R з центром в точці x_0, y_0 : $(x-x_0)^2 + (y-y_0)^2 = R^2$.

Підставимо значення $x_0 = R, y_0 = 0$ та отримаємо: $(x-R)^2 + y^2 = R^2$.

Звідси: $y = \pm \sqrt{R^2 - (x-R)^2}$.

Верхній дузі кола відповідає знак «+».

Отже, на інтервалі $R < x \leq 2R$: $y = \sqrt{R^2 - (x-R)^2}$

- 5) $2R < x < \infty$: точки $x_1 = 2R, y_1 = 0$ та $x_2 = 6, y_2 = -1$.

Рівняння прямої, яка проходить через точки (x_1, y_1) та (x_2, y_2) отримаємо з пропорції:

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

звідси

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

Підставляючи значення $x_1 = 2R, y_1 = 0$ та $x_2 = 6, y_2 = -1$, отримаємо:

$$y = 0 + \frac{-1 - 0}{6 - 2R} (x - 2R)$$

Скорочуючи, отримаємо:

$$y = -\frac{x - 2R}{6 - 2R}$$

Отже, на інтервалі $2R < x < \infty$: $y = -\frac{x - 2R}{6 - 2R}$

Етап 3. Запис формули для розгалуження на 5 гілок.

$$y = \begin{cases} R, & -\infty < x \leq -2R \\ -\frac{1}{2}x, & -2R < x \leq 0 \\ R - \sqrt{R^2 - x^2}, & 0 < x \leq R \\ \sqrt{R^2 - (x - R)^2}, & R < x \leq 2R \\ -\frac{x - 2R}{6 - 2R}, & 2R < x < \infty \end{cases}$$

Алгоритм

Використаємо схему розгалуження у повній формі, оскільки вона більш ефективна, ніж схема розгалуження у скороченій формі:

1. Ввести значення x .
2. Якщо справедлива умова₁, то
 - 2.1. у отримує значення виразу₁;
 - 2.2. інакше
3. Якщо справедлива умова₂, то
 - 3.1. у отримує значення виразу₂;
 - 3.2. інакше
4. Якщо справедлива умова₃, то
 - 4.1. у отримує значення виразу₃;
 - 4.2. інакше
5. Якщо справедлива умова₄, то
 - 5.1. у отримує значення виразу₄;
 - 5.2. інакше у отримує значення виразу₅.
6. Вивести значення y .

Текст програми

```
// Lab_03_3.cpp
// < прізвище, ім'я автора >
// Лабораторна робота № 3.3
// Розгалуження, задане графіком функції.
// Варіант 0.1

#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    double x; // вхідний аргумент
```

```

double R; // вхідний параметр
double y; // результат обчислення виразу

cout << "R = "; cin >> R;
cout << "x = "; cin >> x;

// розгалуження в повній формі
if (x<=-2*R)
    y = R;
else
    if (-2*R<x && x<=0)
        y = -1./2.*x;
    else
        if (0<x && x<=R)
            y = R - sqrt(R*R - x*x);
        else
            if (R<x && x<=2*R)
                y = sqrt(R*R - (x-R)*(x-R));
            else
                y = -(x - 2*R)/(6 - 2*R);

cout << endl;
cout << "y = " << y << endl;

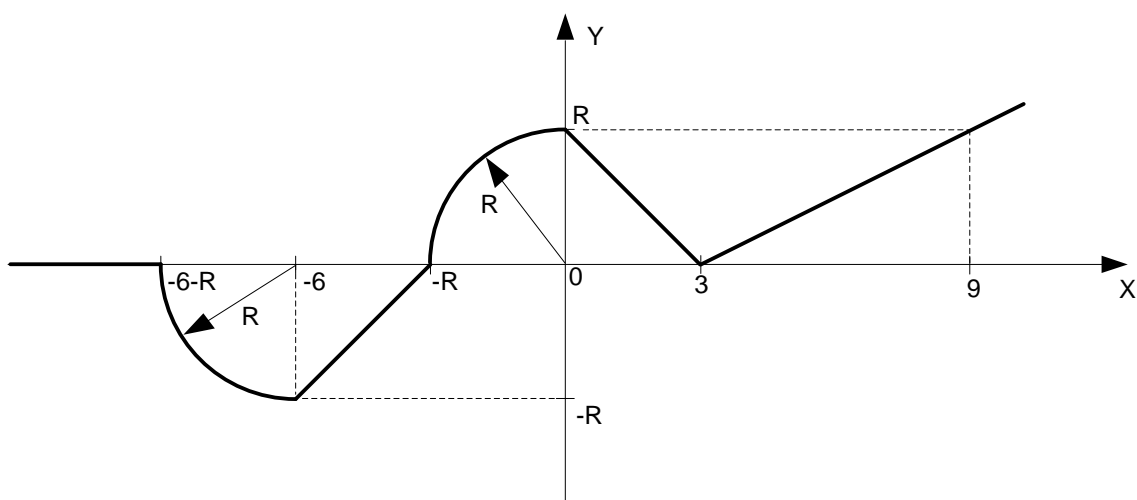
cin.get();
return 0;
}

```

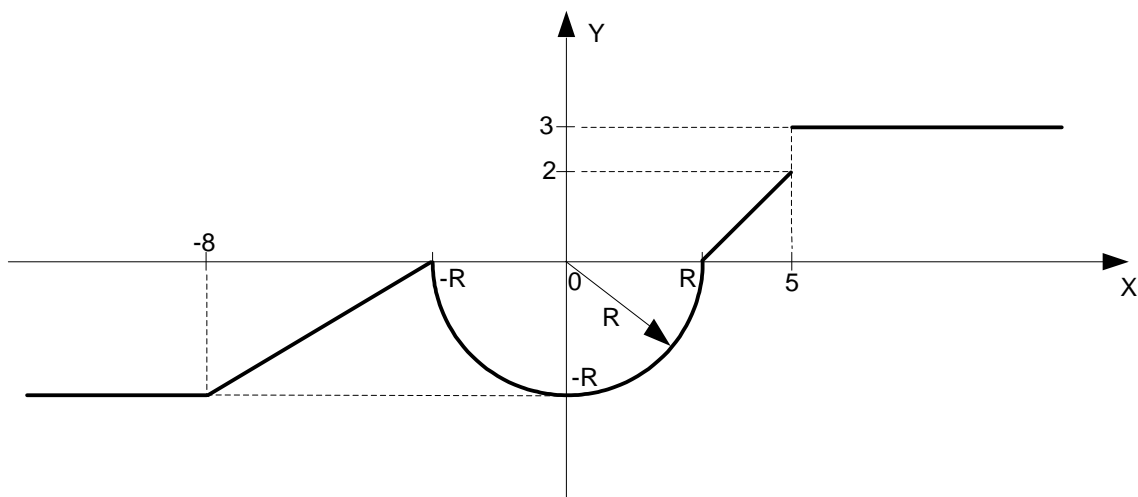
Варіанти лабораторних завдань

Написати програму, яка за введеним значенням аргументу обчислює значення функції, заданої у вигляді графіка. Якщо у завданні є параметр R (параметри R_1 та R_2) – то ці дані вводяться з клавіатури.

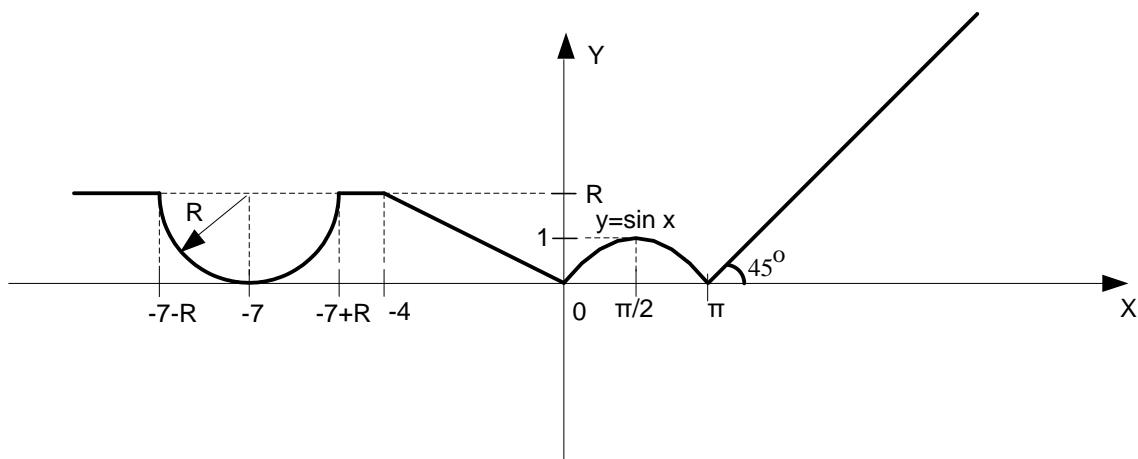
Варіант 1.



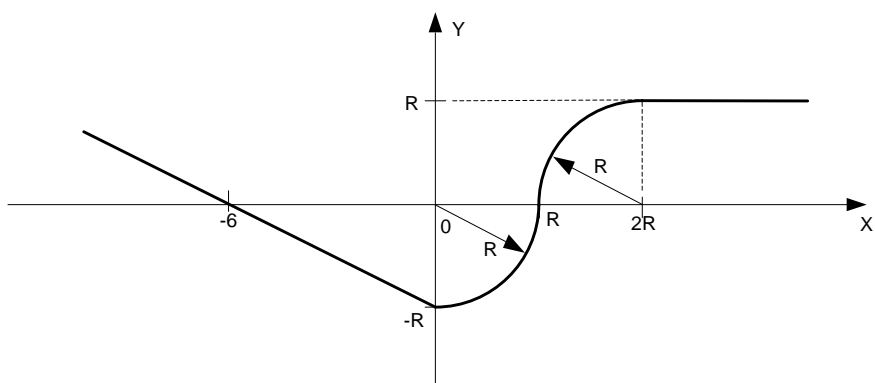
Варіант 2.



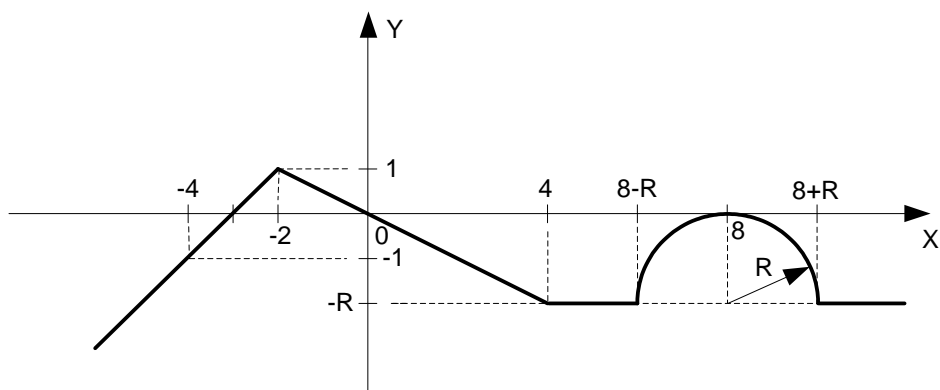
Варіант 3.



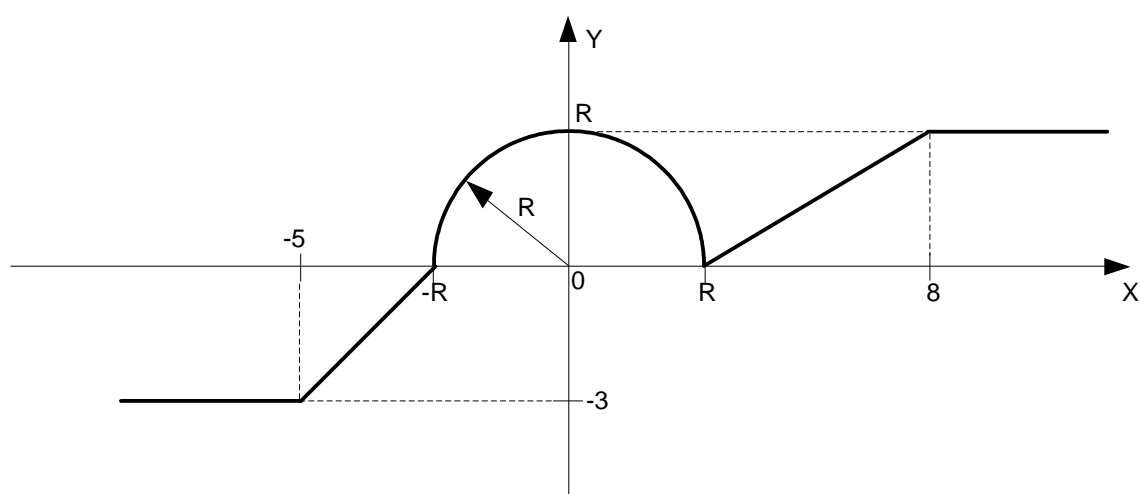
Варіант 4.



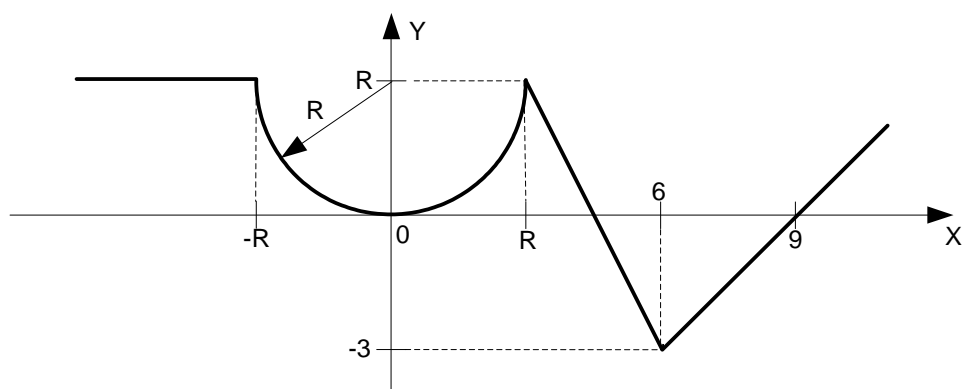
Варіант 5.



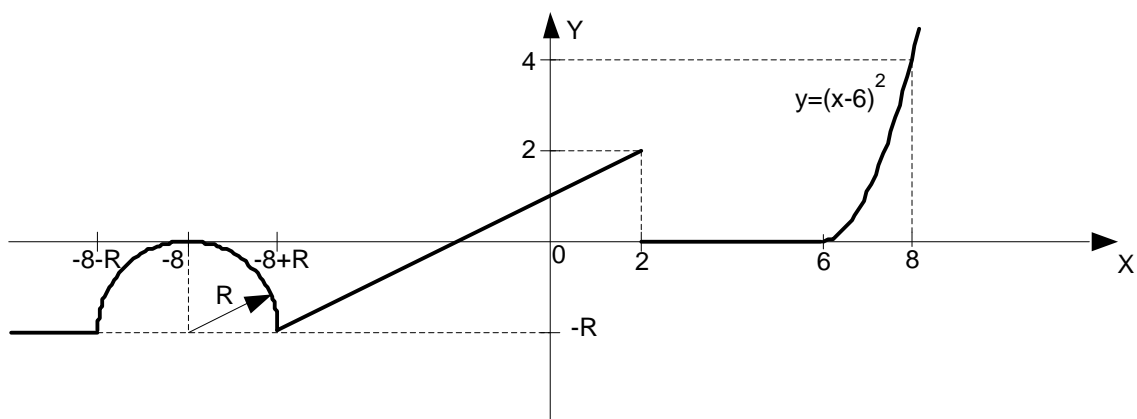
Варіант 6.



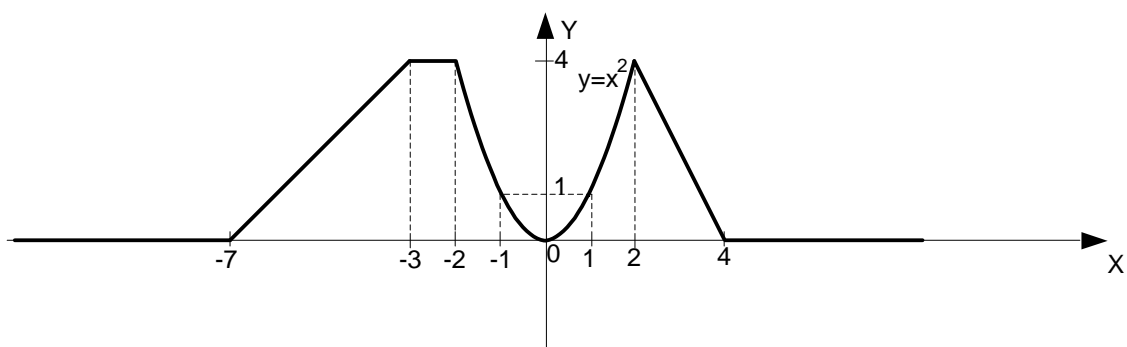
Варіант 7.



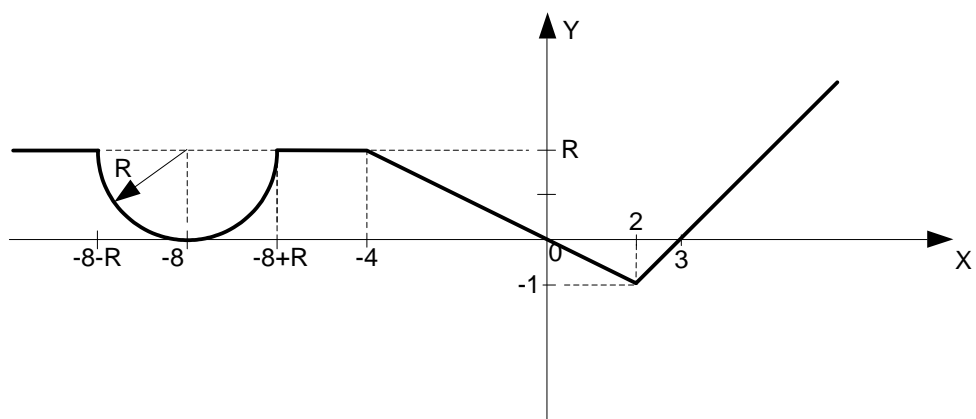
Варіант 8.



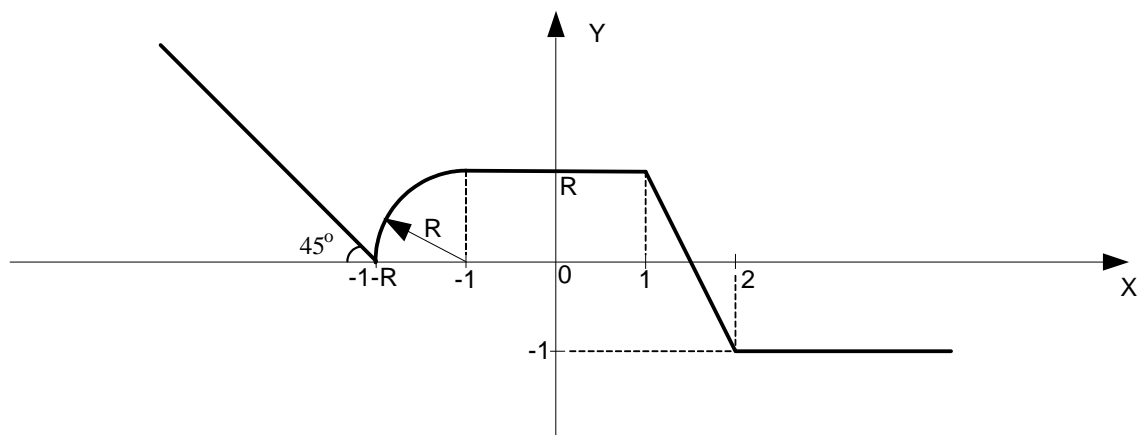
Варіант 9.



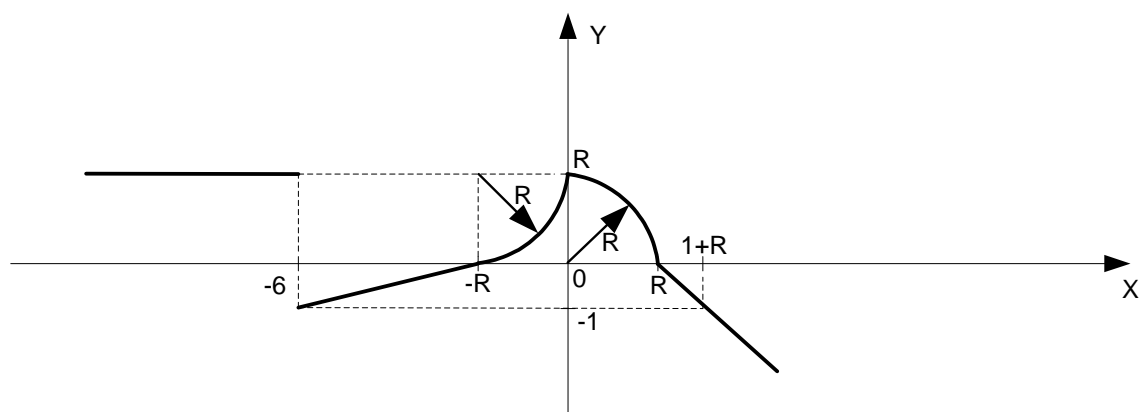
Варіант 10.



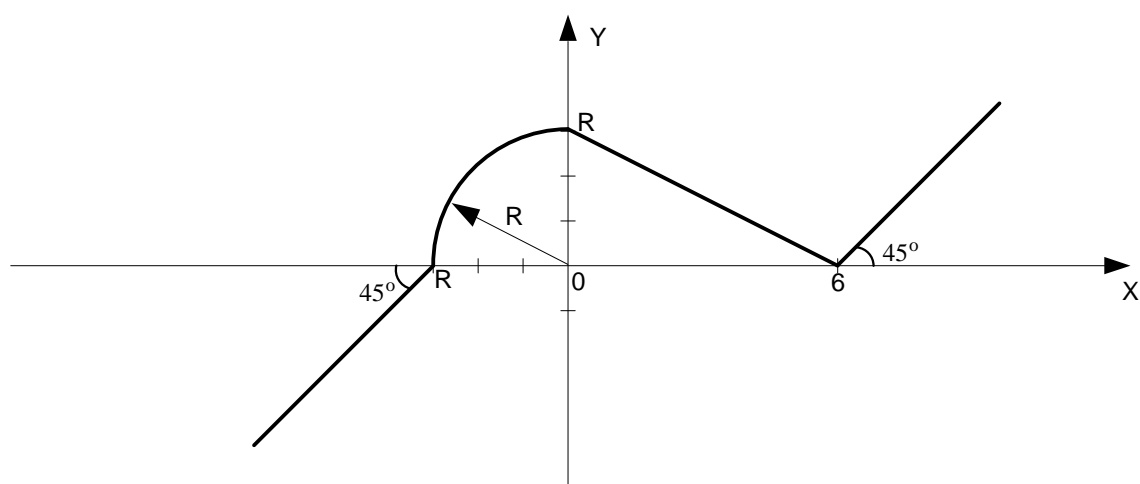
Варіант 11.



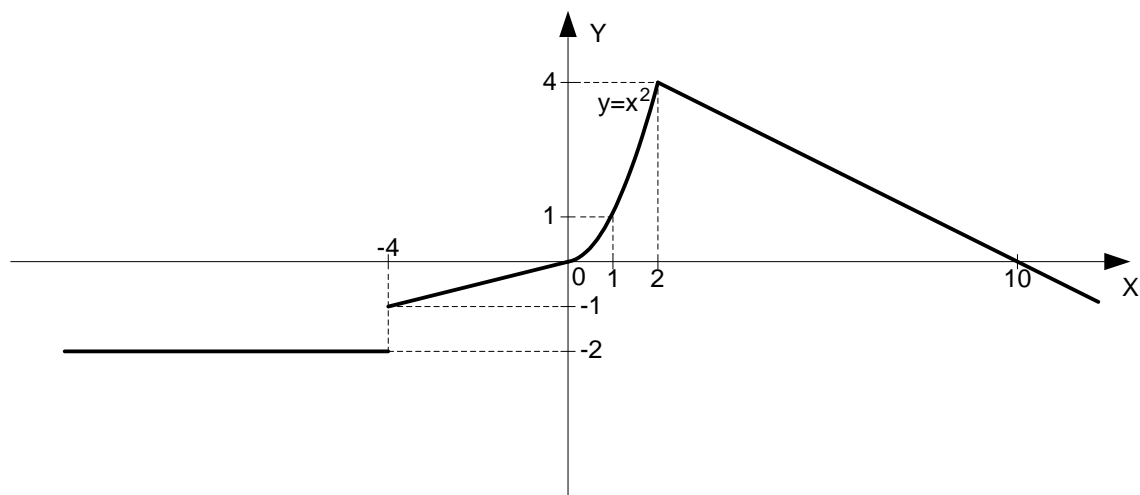
Варіант 12.



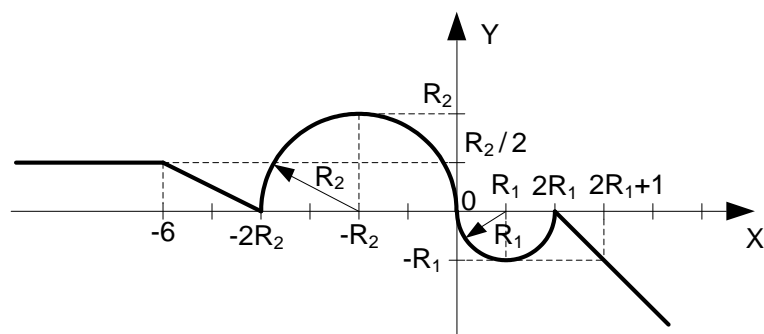
Варіант 13.



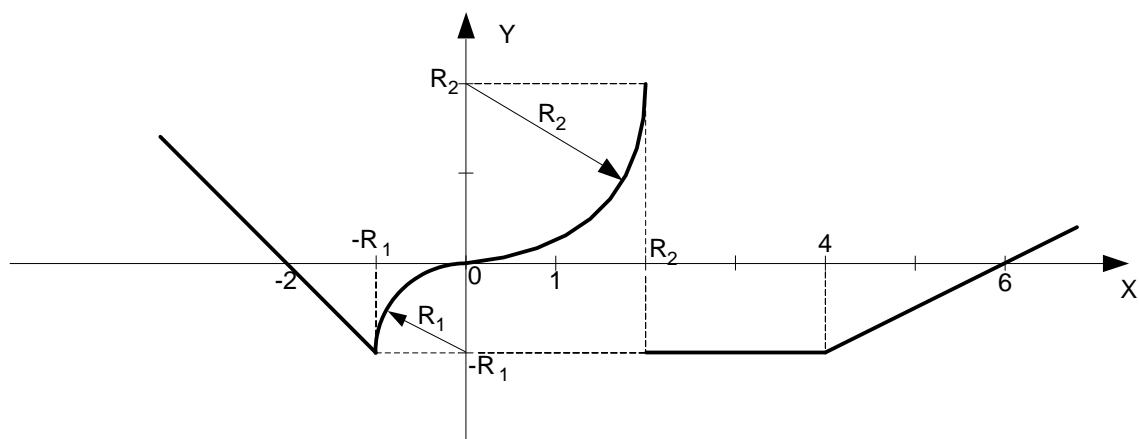
Варіант 14.



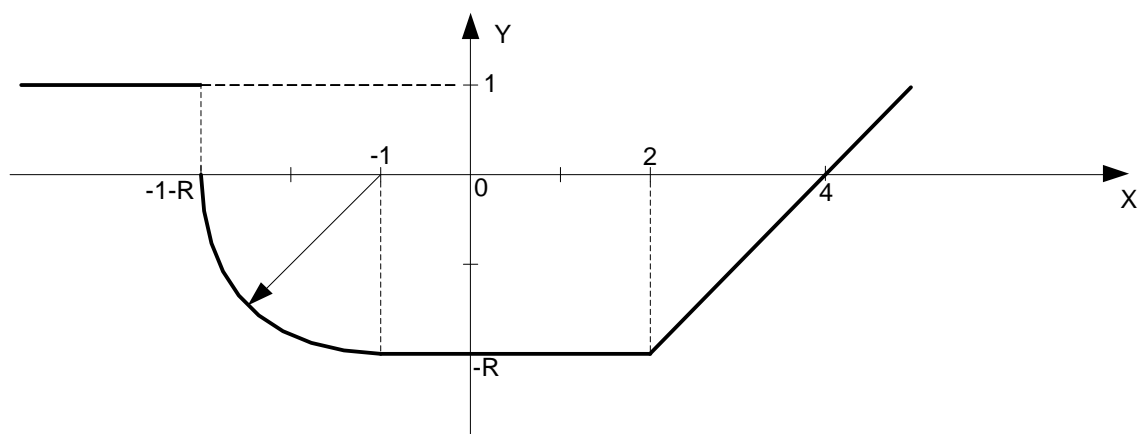
Варіант 15.



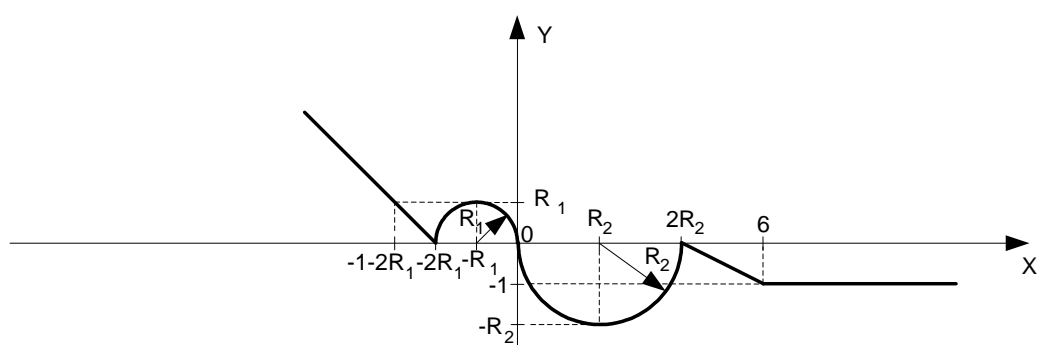
Варіант 16.



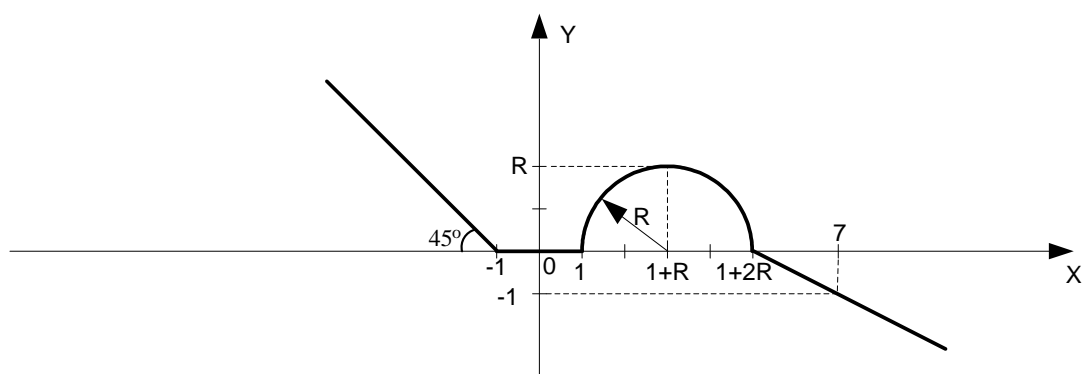
Варіант 17.



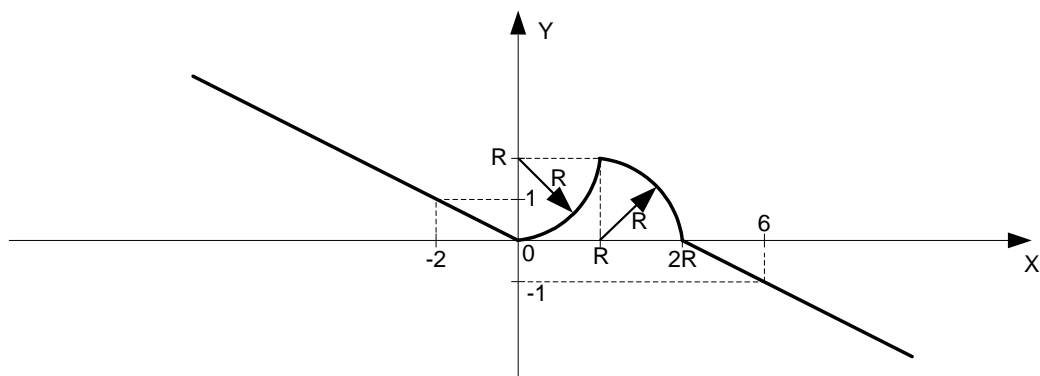
Варіант 18.



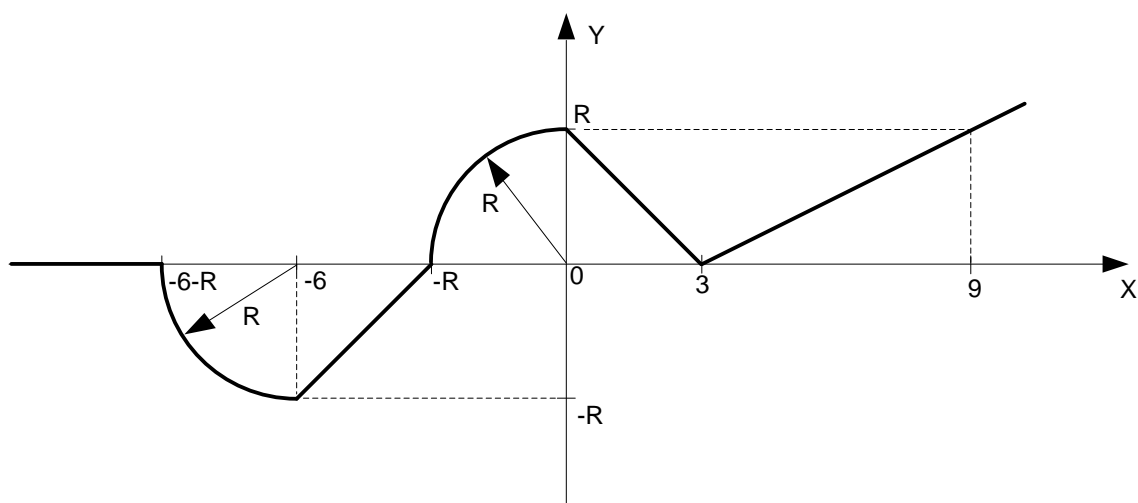
Варіант 19.



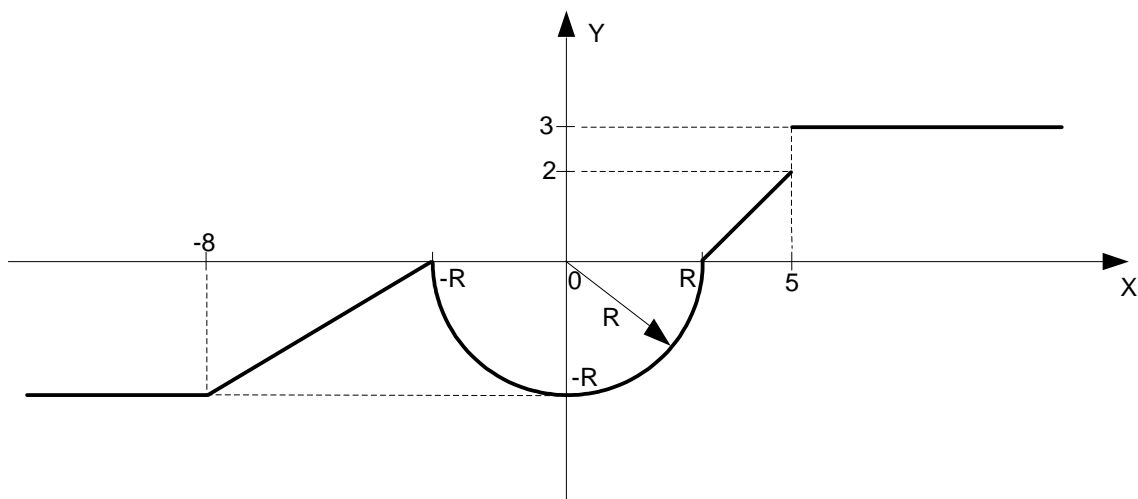
Варіант 20.



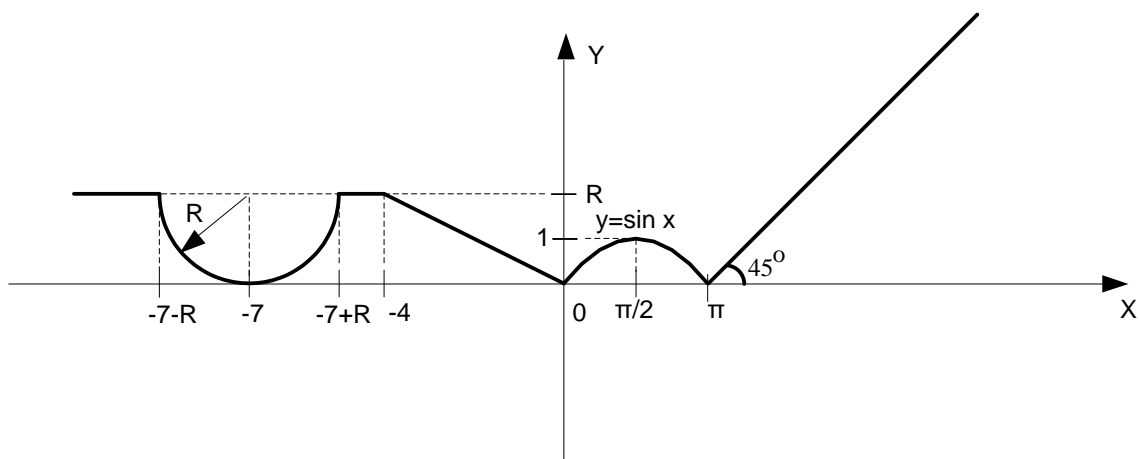
Варіант 21.



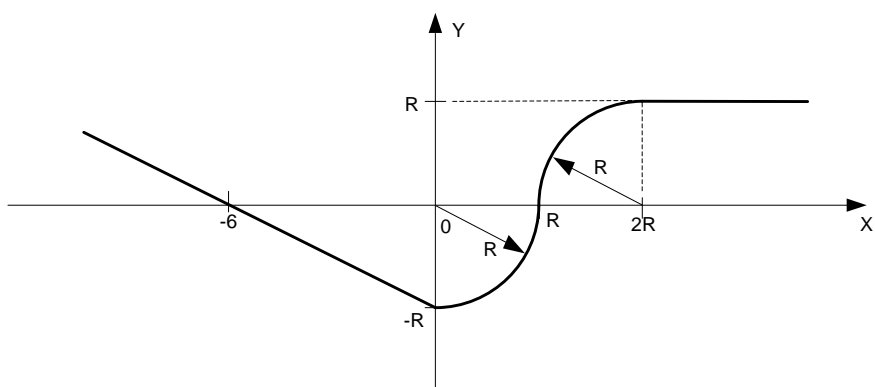
Варіант 22.



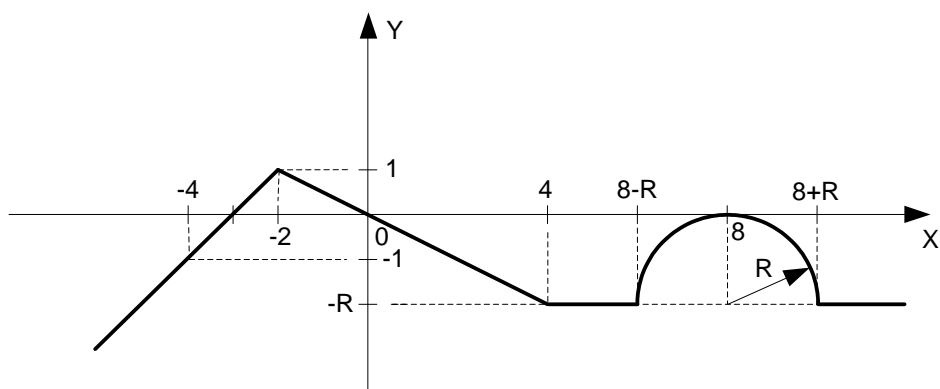
Варіант 23.



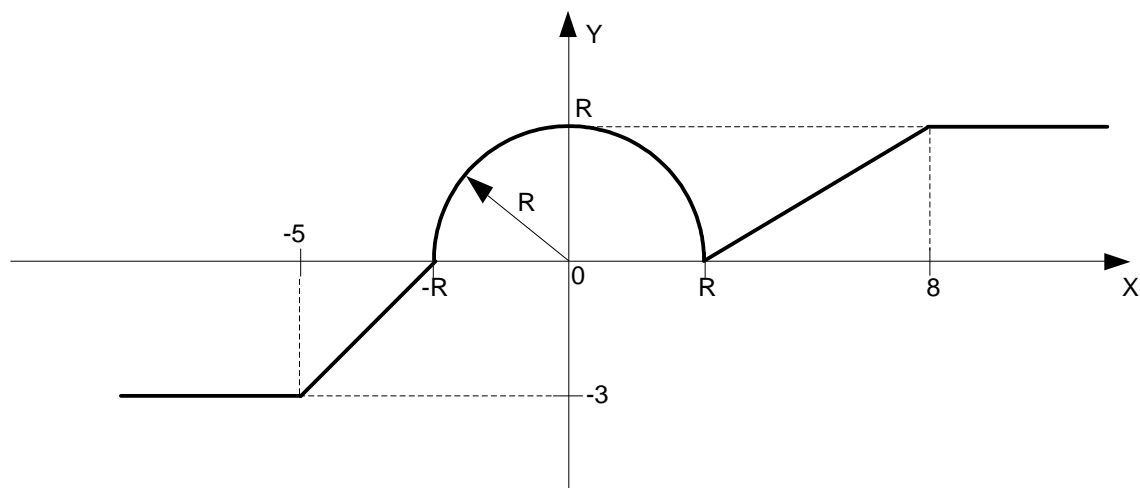
Варіант 24.



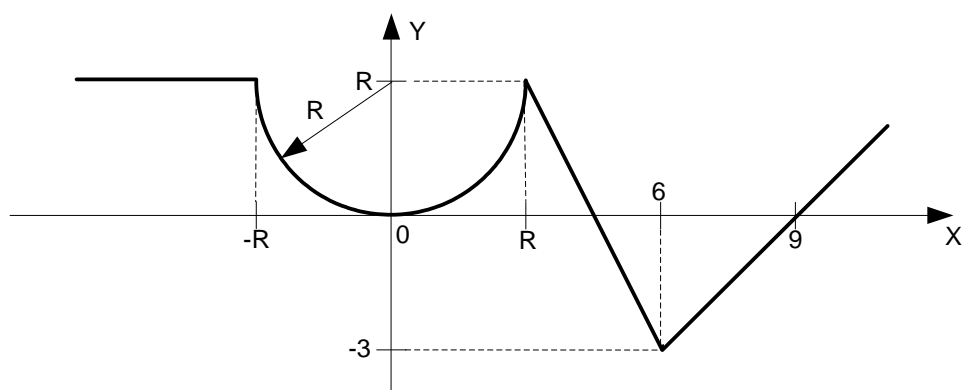
Варіант 25.



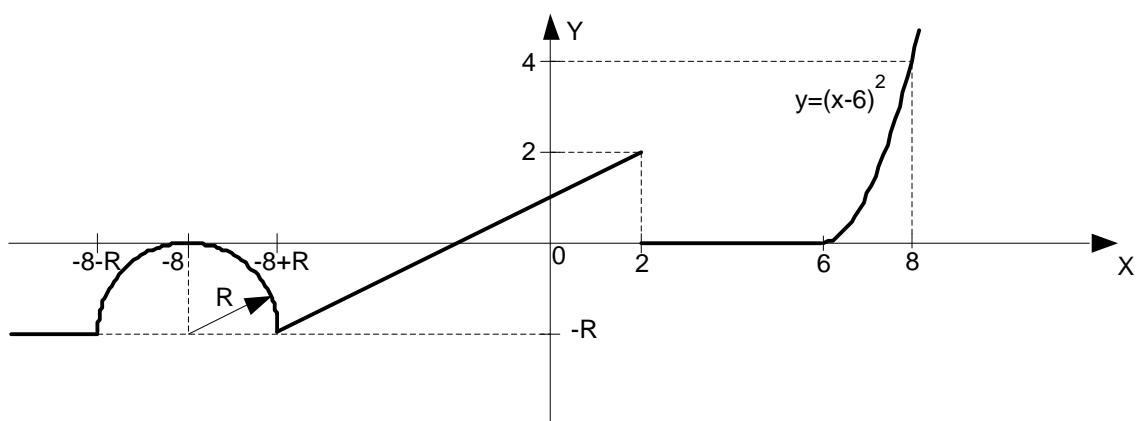
Варіант 26.



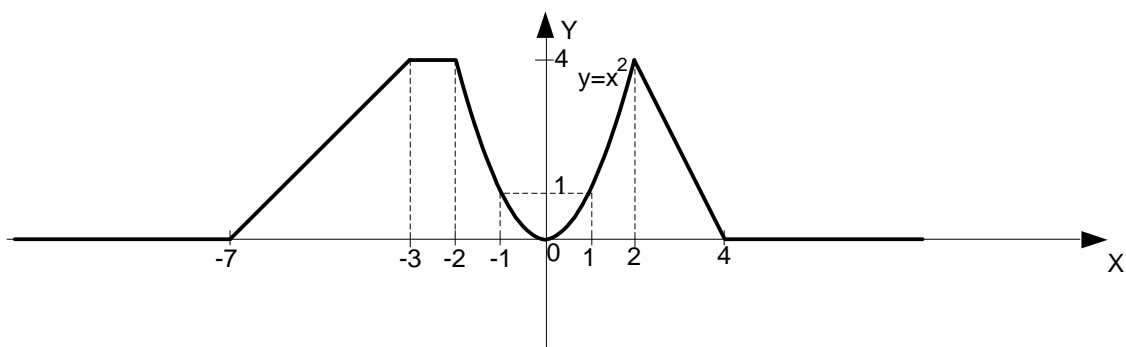
Варіант 27.



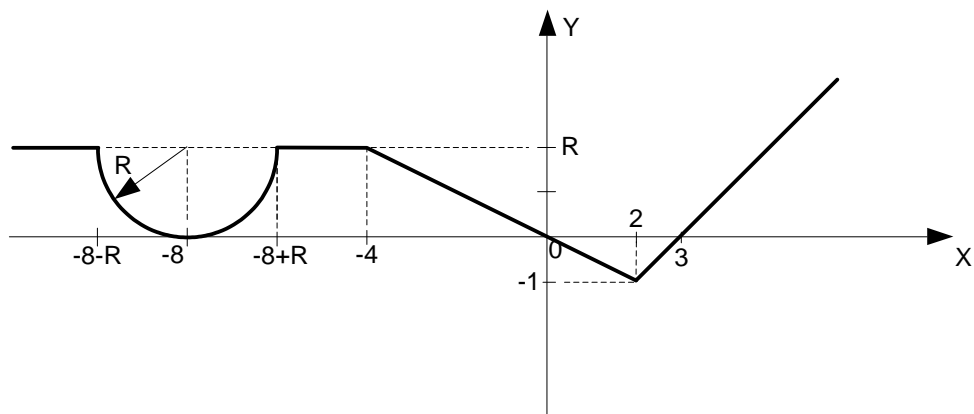
Варіант 28.



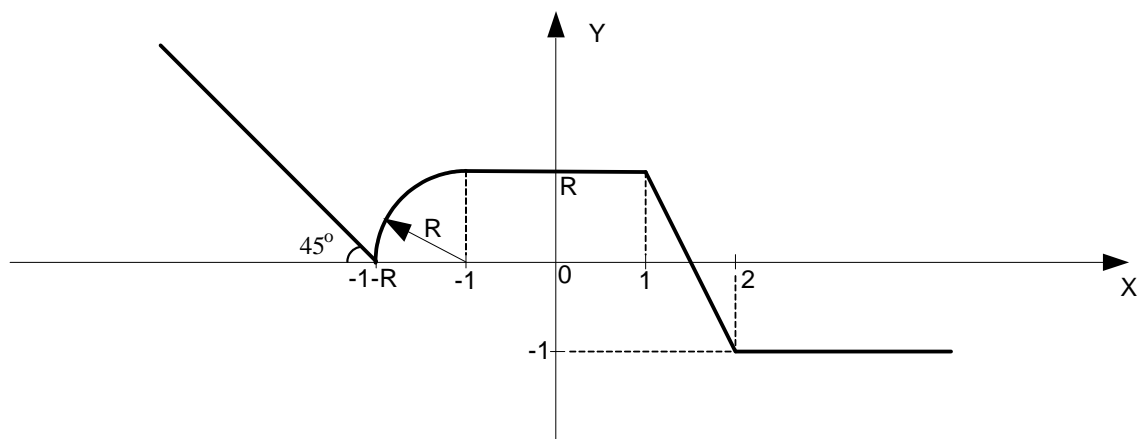
Варіант 29.



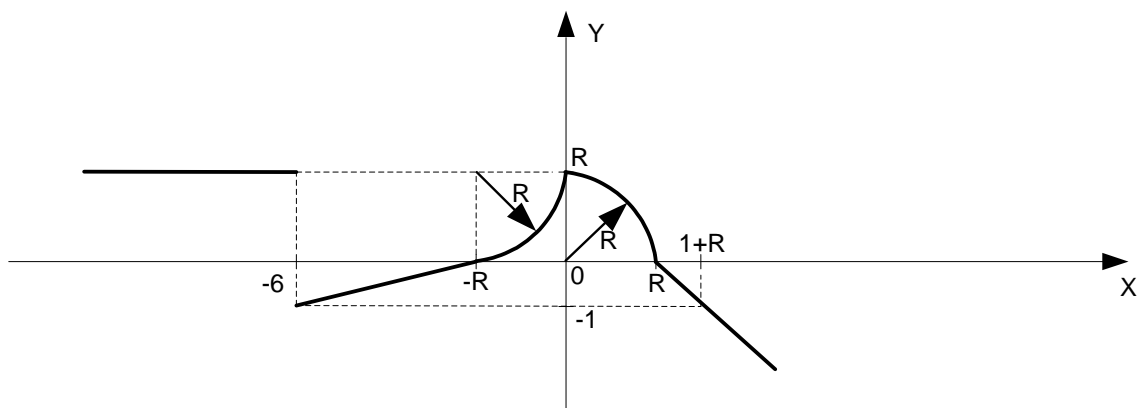
Варіант 30.



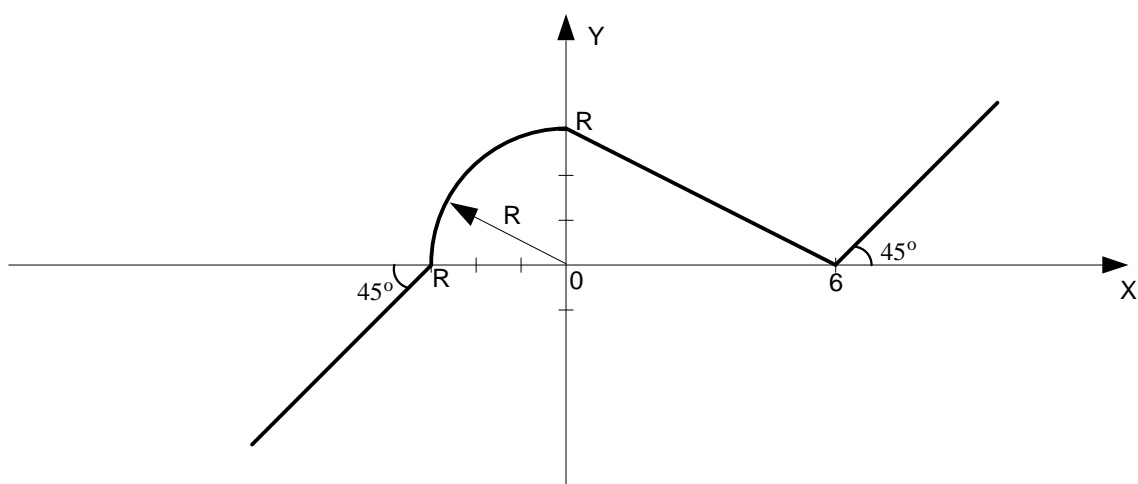
Варіант 31.



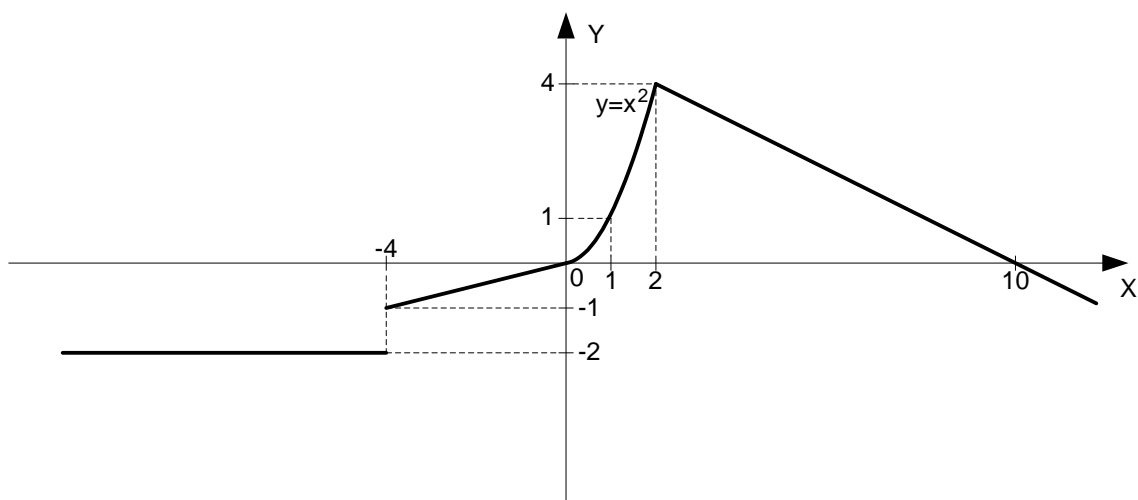
Варіант 32.



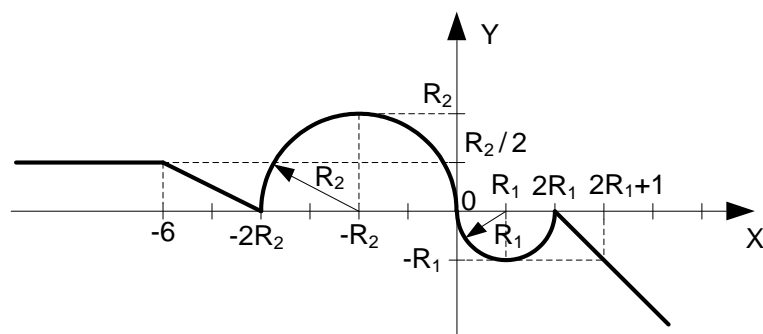
Варіант 33.



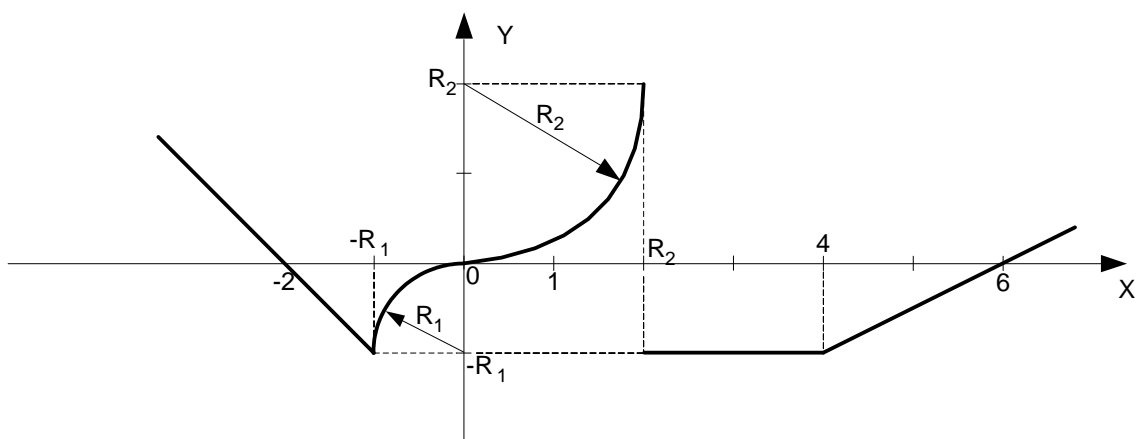
Варіант 34.



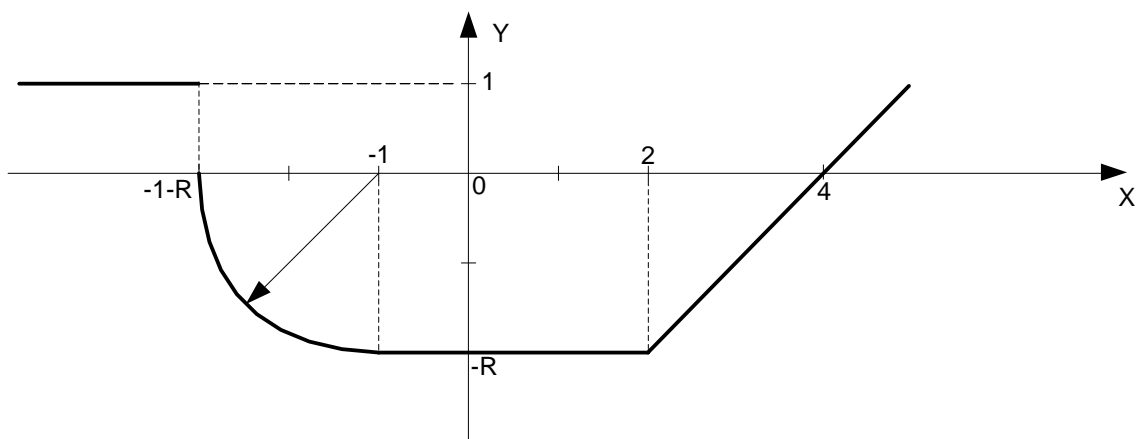
Варіант 35.



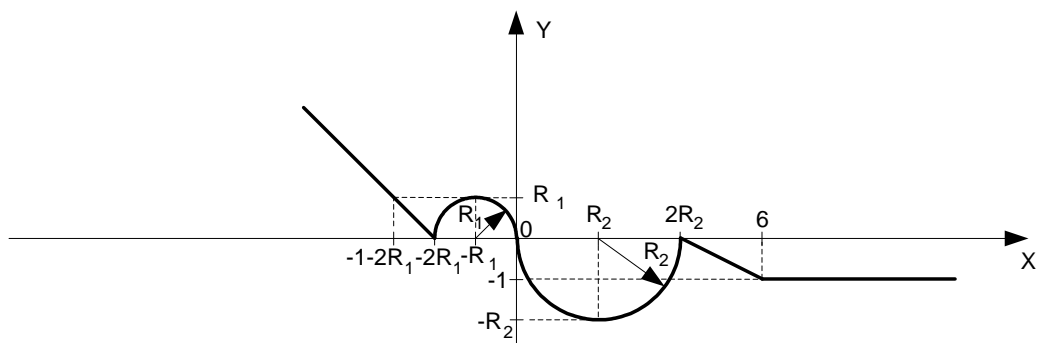
Варіант 36.



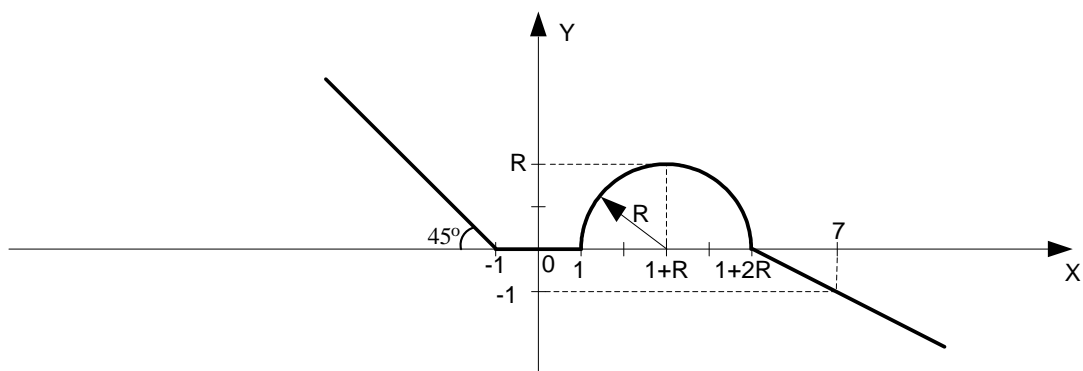
Варіант 37.



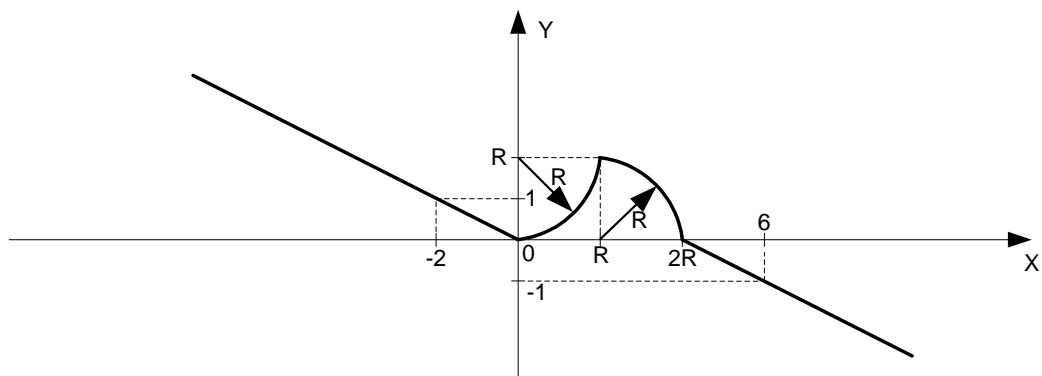
Варіант 38.



Варіант 39.



Варіант 40.



Лабораторна робота № 3.4. Розгалуження, задане плоскою фігурою

Мета роботи

Навчитися описувати формулами нерівності, задані плоскою фігурою.

Питання, які необхідно вивчити та пояснити на захисті

Якщо деякі питання вже описані в одному із попередньо захищених звітів, то їх повторно описувати необов'язково.

- 1) Операції відношення.
- 2) Логічні операції над булівськими величинами.
- 3) По-розрядні логічні операції над цілими величинами.
- 4) Команди C/C++: `if()`... ; `if()`...`else`...; `if()`...`elseif`...; `(...)?`...:....;
- 5) Блок-схеми алгоритмів. Позначення розгалужень на блок-схемах.
- 6) UML-діаграми дії. Позначення розгалужень на UML-діаграмах дії.
- 7) Блок. Операторні дужки.
- 8) Команда C/C++: `switch()`...; команда `break`;
- 9) Стандартні функції, які отримують чи повертають булеві значення.
- 10) Рівняння відрізка прямої, яка проходить через задані точки.
- 11) Рівняння дуги кола заданого радіуса, яка проходить через задані точки.
- 12) Рівняння дуги параболи, яка проходить через задані точки.
- 13) Нерівності, які описують ліву, праву, верхню, нижню, внутрішню, зовнішню частини координатної площини, поділеної прямою лінією, колом, параболою.
- 14) Об'єднання та перетини плоских фігур та їх подання за допомогою логічних операцій.
- 15) Побудова математичної моделі розв'язку завдання свого варіанту.

Оформлення звіту

Вимоги та зразок оформлення звіту наведені у вступі до лабораторного практикуму.

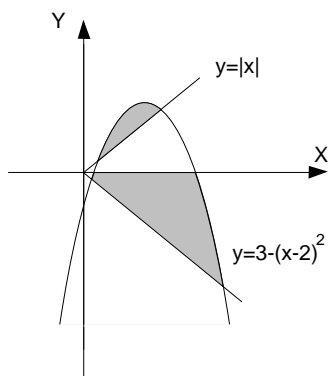
Приклади розв'язання лабораторних завдань

Варіант 0

Умова завдання

Написати програму, яка визначає чи попадає точка з заданими координатами в

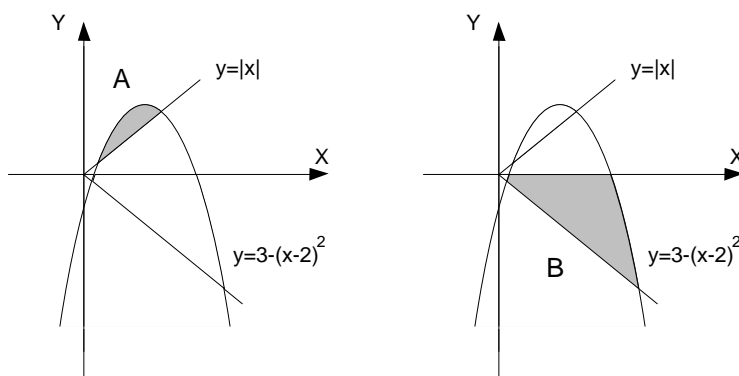
область, зафарбовану на рисунку сірим кольором. Результати роботи програми вивести у вигляді текстового повідомлення. Якщо у завданні є параметр R (параметри R_1 та R_2 , a , b) – то ці дані вводяться з клавіатури.



Розв'язок (побудова математичної моделі)

Етап 1. Визначення кількості незв'язних елементів області, зафарбованої на рисунку з умови завдання сірим кольором.

Область, яка на рисунку зафарбована на рисунку з умови завдання сірим кольором, складається із двох незв'язних елементів: **A** та **B**:



Етап 2. Виведення нерівностей, що описують кожний незв'язний елемент області, зафарбованої на рисунку з умови завдання сірим кольором.

А) Під-область **A** обмежена зверху параболою $y = 3 - (x - 2)^2$

– тому $y \leq 3 - (x - 2)^2$;

знизу – лінією $y = x$

– тому $y \geq x$

та знаходиться над горизонтальною віссю координат $y = 0$

– тому $y > 0$

Оскільки під-область **A** однозначно ідентифікується першими двома нерівностями, то

A: $y \leq 3 - (x - 2)^2$ і $y \geq x$

В) Під-область **B** обмежена зверху параболою $y = 3 - (x - 2)^2$

– тому $y \leq 3 - (x - 2)^2$;

знизу – лінією $y = -x$

– тому $y \geq -x$

та знаходиться під горизонтальною віссю координат $y = 0$

– тому $y \leq 0$

Отже

В: $y \leq 3 - (x - 2)^2$ і $y \geq -x$ і $y \leq 0$

Етап 3. Запис нерівностей, що описують область, зафарбовану на рисунку з умови завдання сірим кольором.

Нерівності, що описують кожний незв'язний елемент області, зафарбованої на рисунку з умови завдання сірим кольором, об'єднуються логічною операцією **або**:

$$(y \leq 3 - (x - 2)^2 \text{ і } y \geq x) \text{ або } (y \leq 3 - (x - 2)^2 \text{ і } y \geq -x \text{ і } y \leq 0)$$

Алгоритм

Використаємо схему розгалуження у повній формі, оскільки вона більш ефективна, ніж схема розгалуження у скороченій формі:

1. Ввести значення x .
2. Ввести значення y .
3. Якщо справедлива умова, то
 - 3.1. вивести значення *так*;
 - 3.2. інакше вивести значення *ні*.

Текст програми

```
// Lab_03_4.cpp
// < прізвище, ім'я автора >
// Лабораторна робота № 3.4
// Розгалуження, задане плоскою фігурою.
// Варіант 0.1

#include <iostream>

using namespace std;

int main()
{
    double x; // вхідний аргумент
    double y; // вхідний параметр

    cout << "x = "; cin >> x;
    cout << "y = "; cin >> y;

    // розгалуження в повній формі
    if (( y <= 3 - (x - 2)*(x - 2) && y >= x ) ||
        ( y <= 3 - (x - 2)*(x - 2) && y >= -x && y <= 0 ))
        cout << "yes" << endl;
    else
```

```

        cout << "no" << endl;

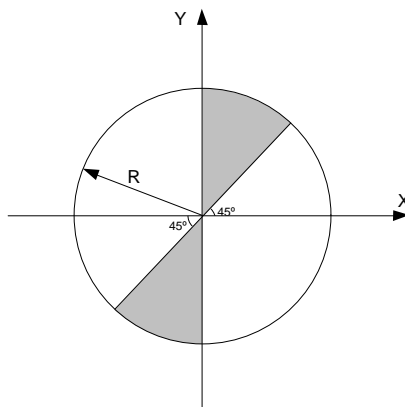
    cin.get();
    return 0;
}

```

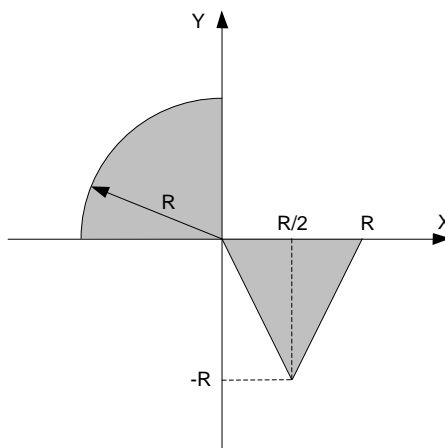
Варіанти лабораторних завдань

Написати програму, яка визначає чи попадає точка з заданими координатами в область, зафарбовану на рисунку сірим кольором. Результати роботи програми вивести у вигляді текстового повідомлення. Якщо у завданні є параметр R (параметри R_1 та R_2 , a , b) – то ці дані вводяться з клавіатури.

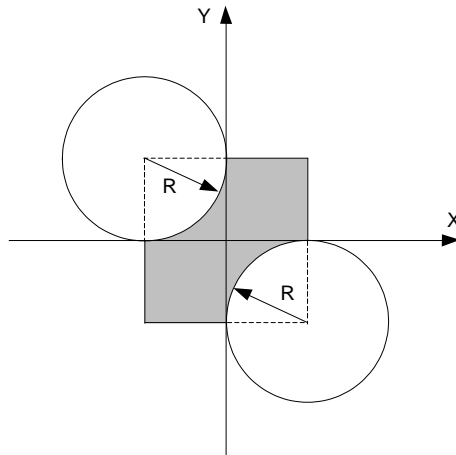
Варіант 1.



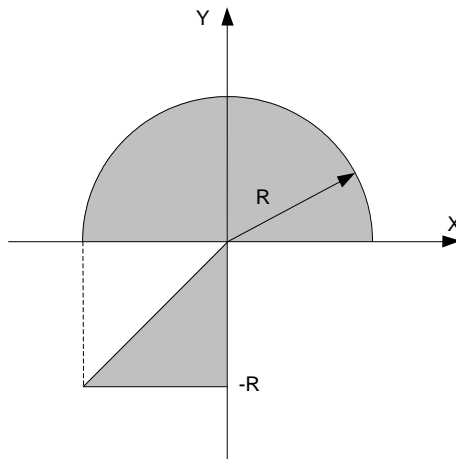
Варіант 2.



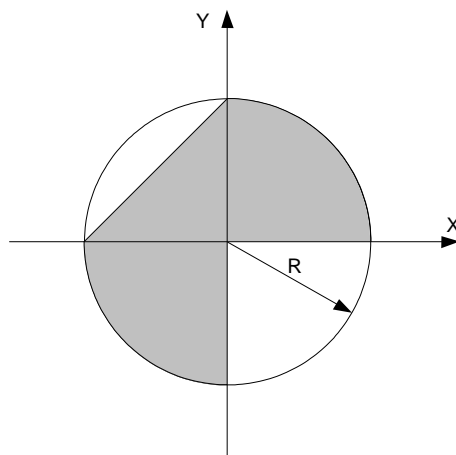
Варіант 3.



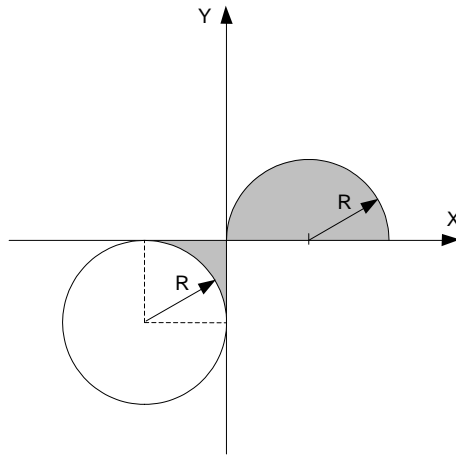
Варіант 4.



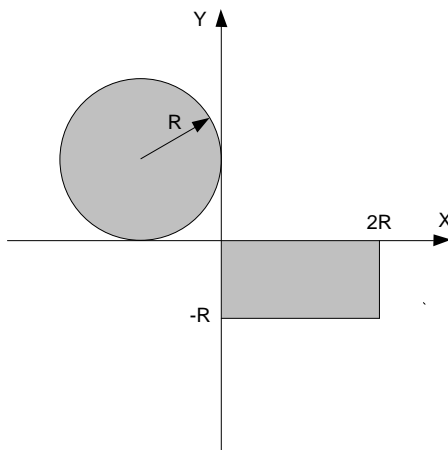
Варіант 5.



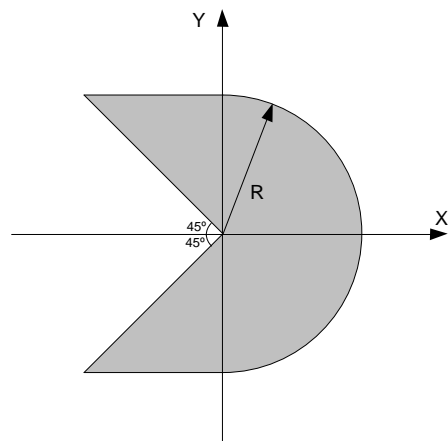
Варіант 6.



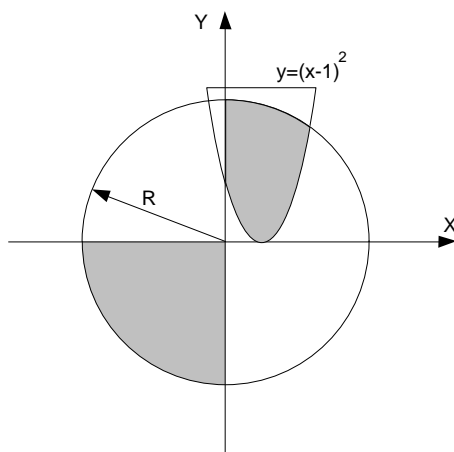
Варіант 7.



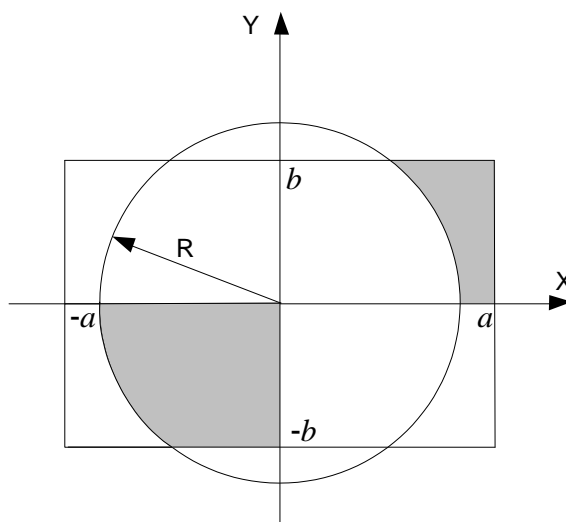
Варіант 8.



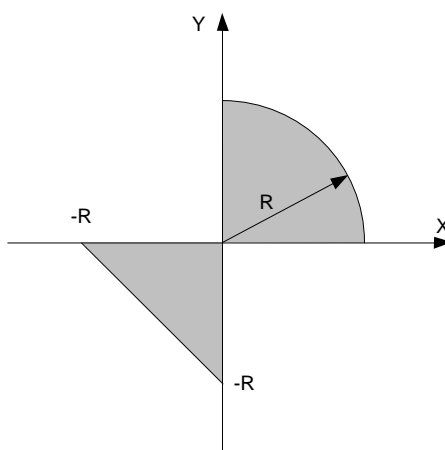
Варіант 9.



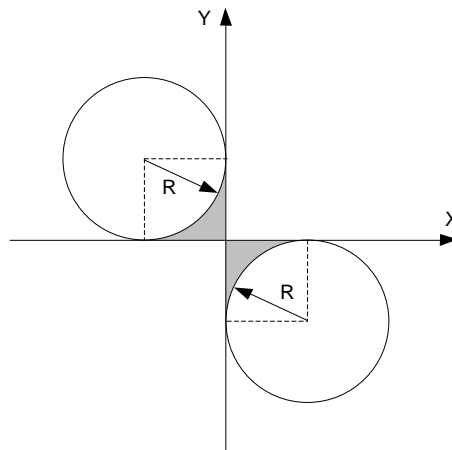
Варіант 10.



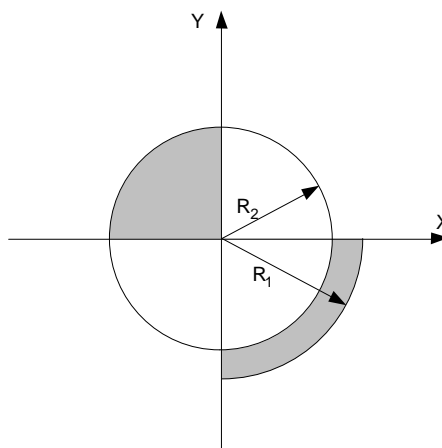
Варіант 11.



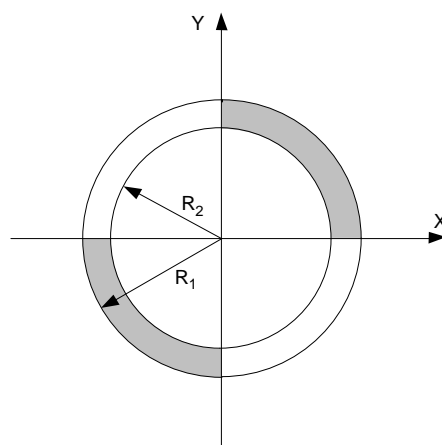
Варіант 12.



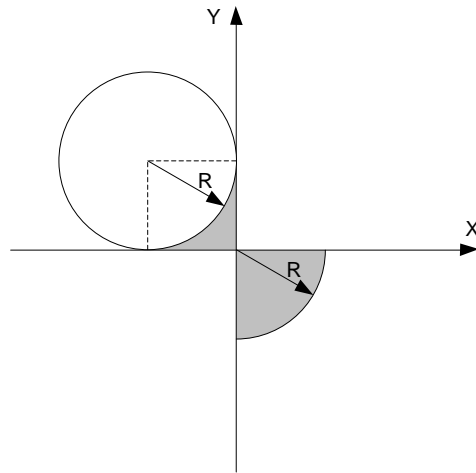
Варіант 13.



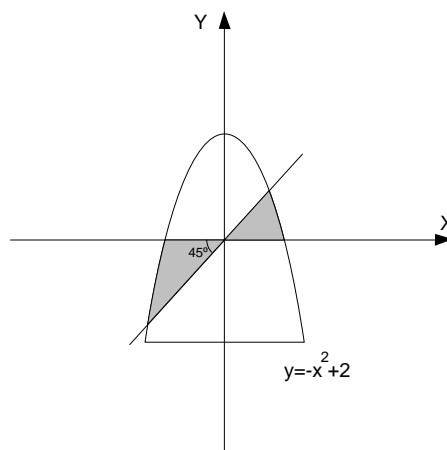
Варіант 14.



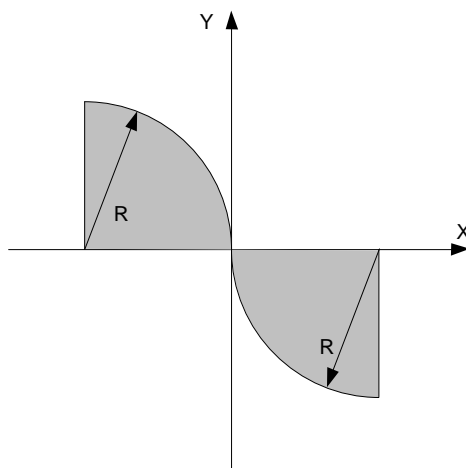
Варіант 15.



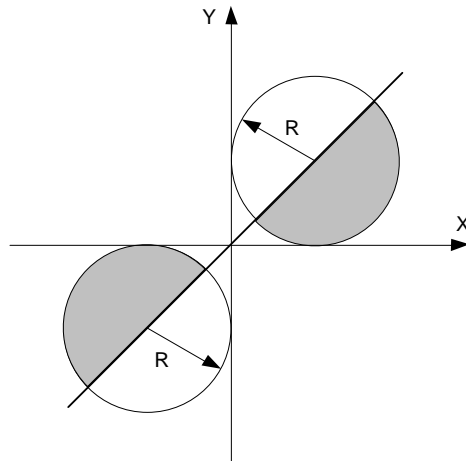
Варіант 16.



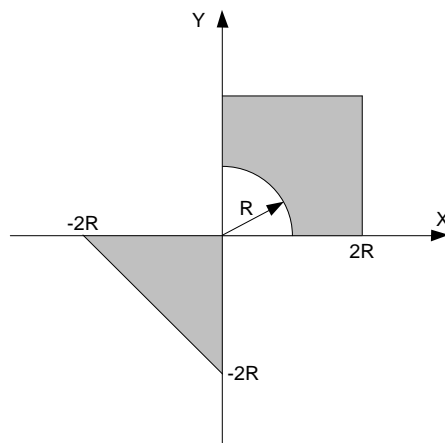
Варіант 17.



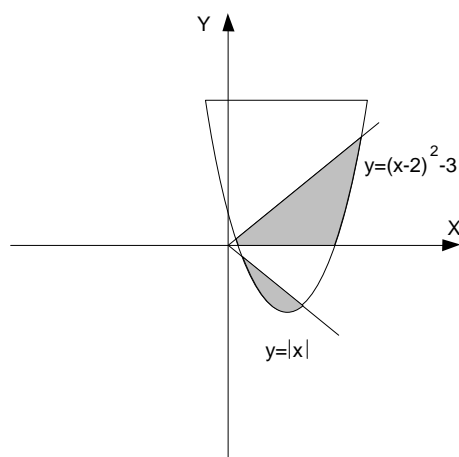
Варіант 18.



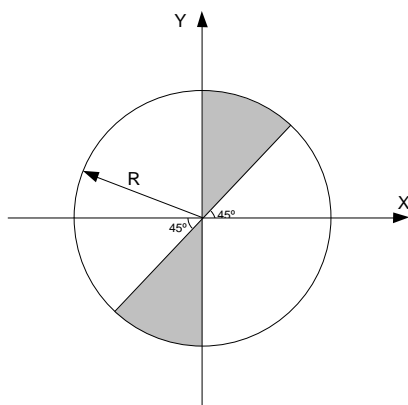
Варіант 19.



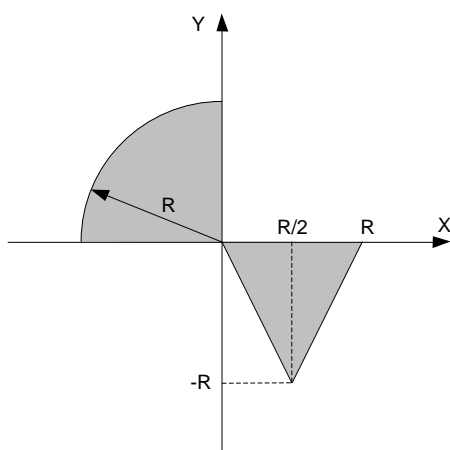
Варіант 20.



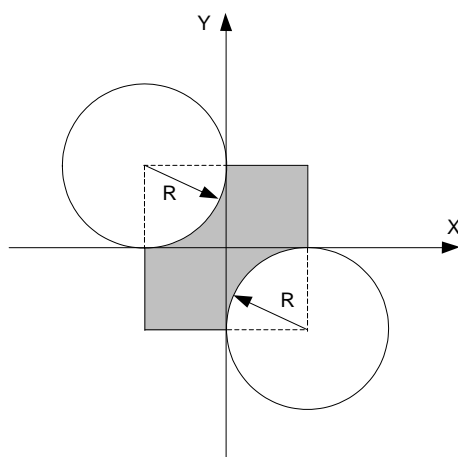
Варіант 21.



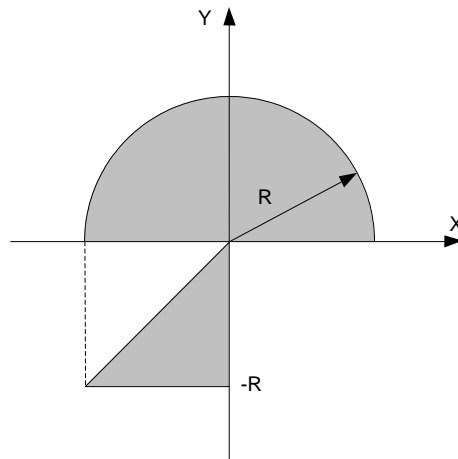
Варіант 22.



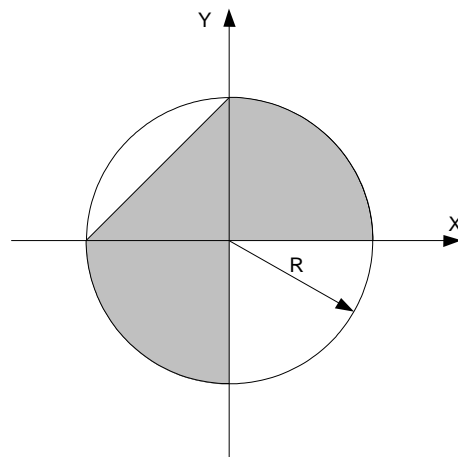
Варіант 23.



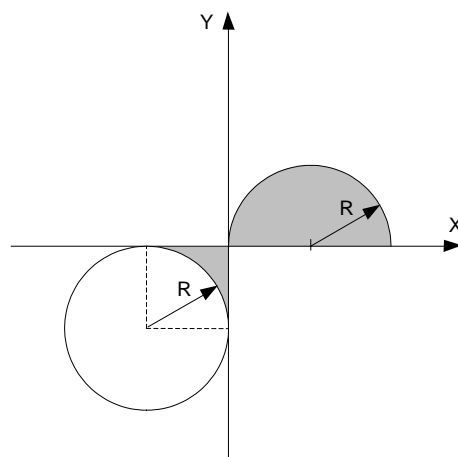
Варіант 24.



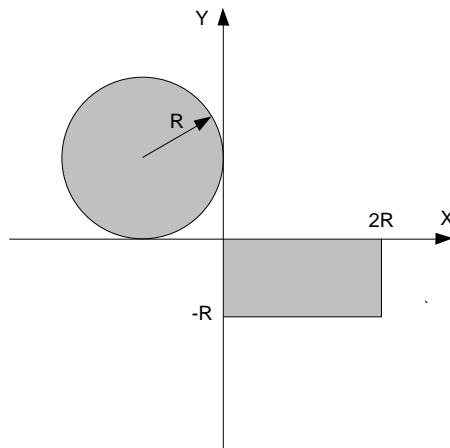
Варіант 25.



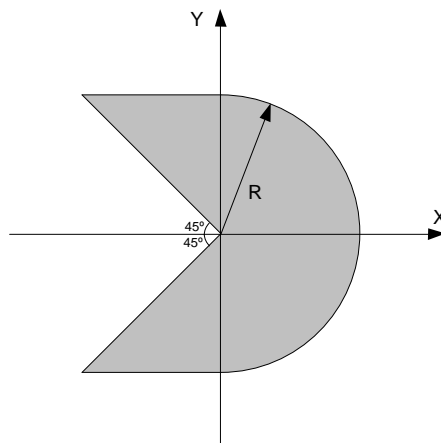
Варіант 26.



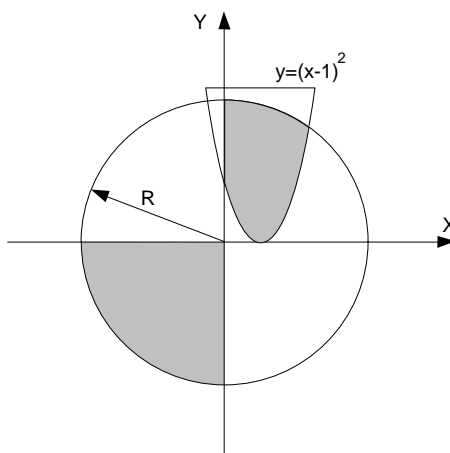
Варіант 27.



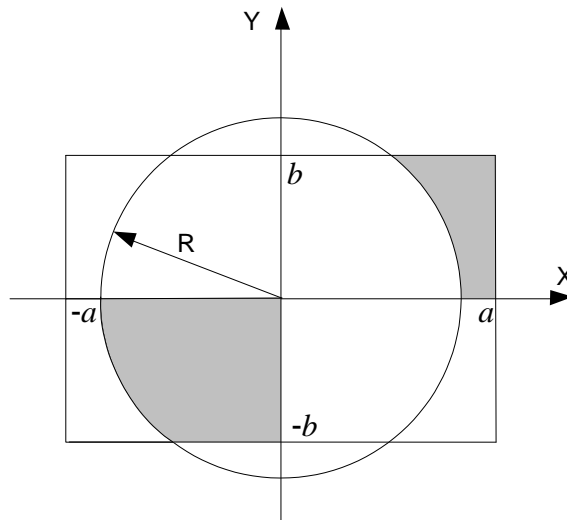
Варіант 28.



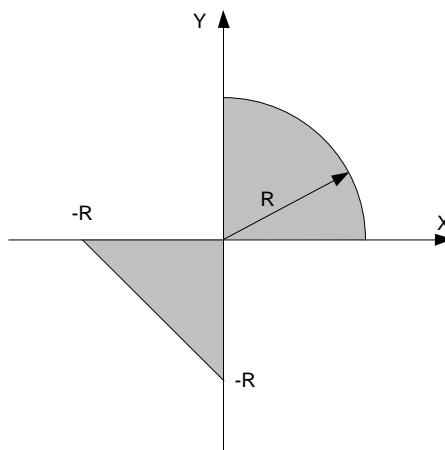
Варіант 29.



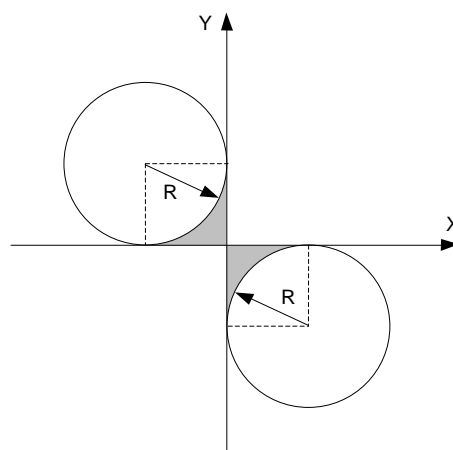
Варіант 30.



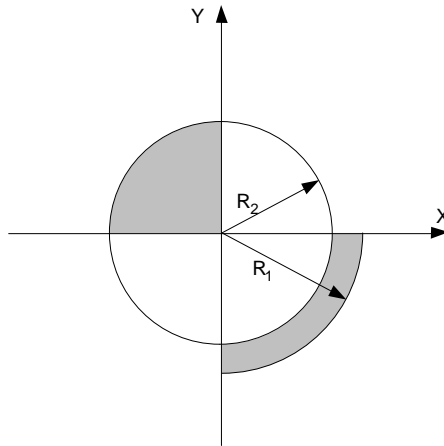
Варіант 31.



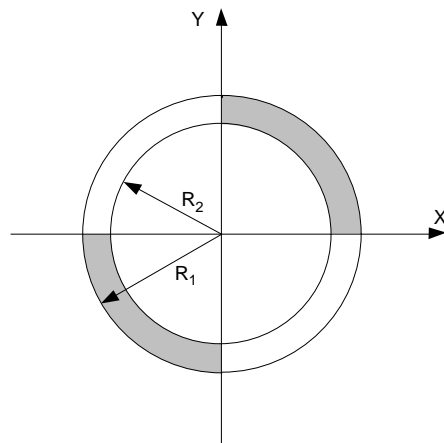
Варіант 32.



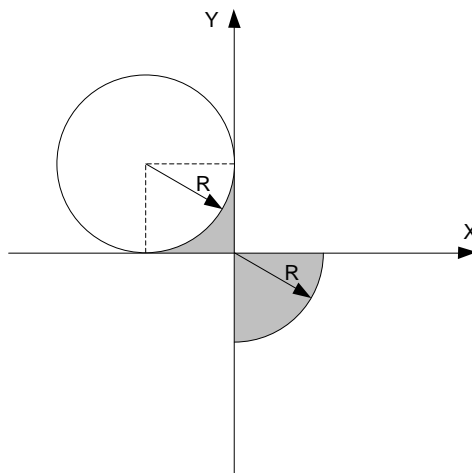
Варіант 33.



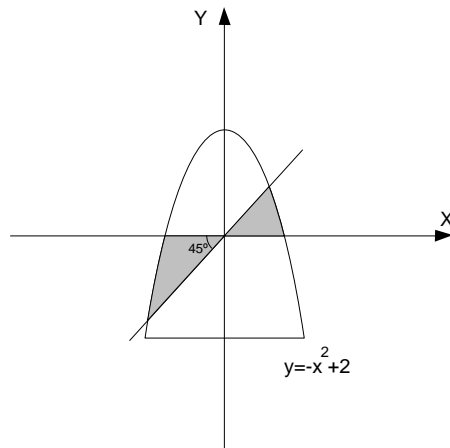
Варіант 34.



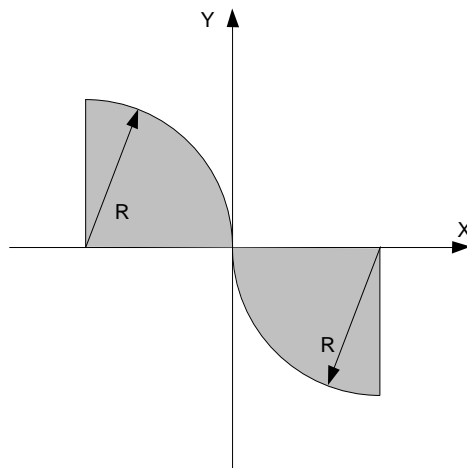
Варіант 35.



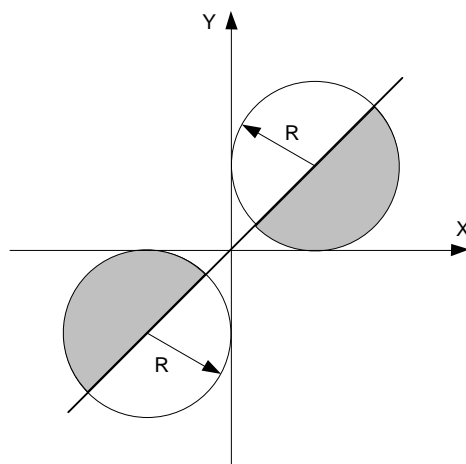
Варіант 36.



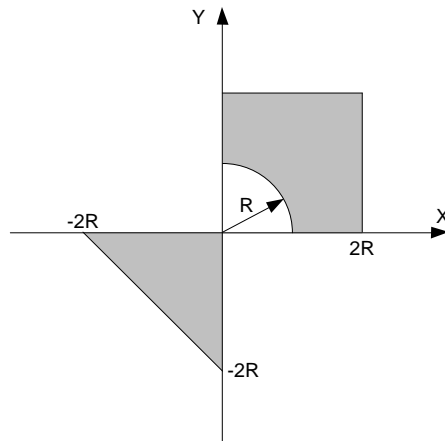
Варіант 37.



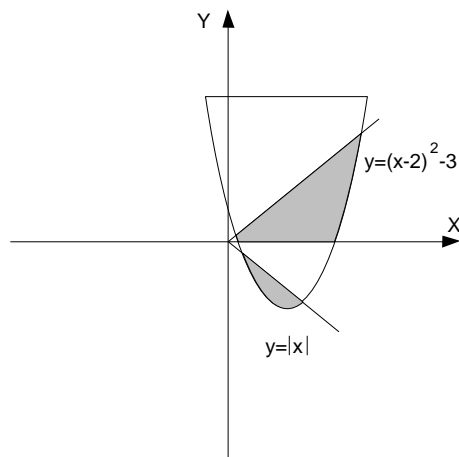
Варіант 38.



Варіант 39.



Варіант 40.



Питання та завдання для контролю знань

1	Яким буде результат виконання фрагменту програми? (впишіть відповідь у праву колонку)	<pre> a = 1; b = -1; if (a > b) a = a - b; else b = b - a; cout << a << endl; cout << b << endl; </pre>	
2	Напишіть фрагмент програми, який обчислює максимальне значення із трьох цілих чисел. (впишіть відповідь у праву колонку)		
3	Поставте у відповідність елементи лівого та правого списків (пронумеруйте відповідні елементи правого списку).		
	1	+	арифметичні операції
	2	–	операції відношення
	3	=	логічні операції
	4	>	операції над розрядами (бітами)
	5	<	операції типу «множення»
	6	>=	операції типу «додавання»
	7	<=	результат – типу <code>boolean</code>
	8	<>	множення
	9	&&	додавання
	10		ділення
	11	!	віднімання
	12	*	зміна знаку
	13	/	ділення націло
	14	%	обчислення остачі від ділення націло
	15	<<	логічне «і»
	16	>>	логічне об'єднуюче «або»
			логічне заперечення
			розрядний зсув вправо
			розрядний зсув вліво
			результат буде цілий
			перевірка на рівність
			перевірка на нерівність
			перевірка, чи більше
			перевірка, чи менше
			перевірка, чи більше або дорівнює
			перевірка, чи менше або дорівнює
			мають найвищий пріоритет
			мають найнижчий пріоритет

4. Яким буде результат виконання фрагменту програми ? (всі необхідні оголошення зроблено)

```
int i=1, j=0, x;

if ( i > 0 )
    if ( j > i )
        x = j;
    else
        x = i;

cout << x << endl;
```

5. Нарисувати блок-схему та UML-діаграму дії до розгалуження із завдання № 4.

6. Яким буде результат виконання фрагменту програми ? (всі необхідні оголошення зроблено)

```
int i=1, j=0, x;

if ( i > 0 )
{
    if ( j > i )
        x = j;
}
else
    x = i;

cout << x << endl;
```

7. Нарисувати блок-схему та UML-діаграму дії до розгалуження із завдання № 6.

8. Яким буде результат виконання фрагменту програми ? (всі необхідні оголошення зроблено)

```
int a=-1, b=1, x;

x = ( a > b ) ? a -= b : b -= a;

cout << a << endl;
cout << b << endl;
cout << x << endl;
```

9. Яким буде результат виконання фрагменту програми ? (всі необхідні оголошення зроблено)

```
int a=-1, b=1;

if ((a++) > (--b))
    a -= b;
else
    b -= a;

cout << a << endl;
cout << b << endl;
```

10. Яким буде результат виконання фрагменту програми ? (всі необхідні оголошення зроблено)

```
int x=1, y;

if ( x == 0 )
    y = 1;
else if ( x != 0 )
    y = 2;
else
    y = 3;

cout << y << endl;
```

11. Яким буде результат виконання фрагменту програми ? (всі необхідні оголошення зроблено)

```
int i=1, j=0, x;

switch (i)
{
    case 0:
        x = j;
    case 1:
        x = 0;
    case 2:
        x = i;
    default:
        x = 5;
}
cout << x << endl;
```

12. Яким буде результат виконання фрагменту програми ? (всі необхідні оголошення зроблено)

```
int i=1, j=0, x;

switch (i)
{
    case 0:
        x = j;
        break;
    case 1:
        x = 0;
        break;
    case 2:
        x = i;
        break;
    default:
        x = 5;
}
cout << x << endl;
```

Предметний покажчик

?		Б	
?:, 15, 28		Блок, 21	
	{	Блок команд, 21	
		Булевий тип, 17	
{}, 21		Булеві операції, 13, 17	
		Вирази відношення, 18	
		Логічні операції, 18	
B		К	
bool, 17		Команда вибору варіанта, 15, 28	
break, 31		Команда переходу, 20	
	G	Команда розгалуження, 13, 22	
		Повна форма, 14, 22	
goto, 20		Скорочена форма, 13, 22	
I		П	
if, 13, 22		Порожня команда, 20	
S		У	
switch, 15, 28		Умовна операція, 15, 28	

Література

Основна

1. Ковалюк Т.В. Основи програмування. – К. ВНУ, 2005. – 384 с.
2. Вирт Н. Алгоритмы + структуры данных = программы. – М.: Мир, 1985.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2007. – 461 с.
4. Павловская Т.А., Щупак Ю.А. С/С++. Структурное программирование: Практикум. СПб.: Питер, 2005. – 239 с.

Додаткова

5. Кнут, Дональд, Эрвин Искусство программирования. 3-е издание. Том1. М.: Вильямс, 2001.
6. Кнут, Дональд, Эрвин Искусство программирования. 3-е издание. Том2. М.: Вильямс, 2001.
7. Кнут, Дональд, Эрвин Искусство программирования. 3-е издание. Том3. М.: Вильямс, 2001.
8. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию. – М.: Наука, 1990.
9. Вирт Н. Систематическое программирование. Введение. – М., Мир, 1977.
10. Ахо А.В., Хопкрофт Дж., Ульман Дж. Д. Структуры данных и алгоритмы. Пер. с англ.: М.: «Вильямс», 2001.– 384 с.: ил.
11. Себест Р. Основные концепции языков программирования. 5-е издание. М.: Вильямс, 2001.