

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ  
по виконанню лабораторних робіт  
з дисципліни «*Інформатика з основами програмування*»

*Частина – «Основи програмування»*

Рівень вищої освіти – «Бакалавр»

Спеціальність –\_1\_101 "Екологія"

183 "Технології захисту навколишнього середовища"

## **Зміст**

<b>Передмова .....</b>	<b>3</b>
<b>Лабораторна робота №1.....</b>	<b>5</b>
<b>Лабораторна робота №2.....</b>	<b>12</b>

## Передмова

Метою дисципліни є формування у студентів теоретичних знань по алгоритмізації процесів обробки інформації та практичних навичок по основам програмування.

Мови ЕОМ виступають як засіб спілкування. Якщо людська мова - це засіб спілкування людей, то мова ЕОМ - засіб спілкування людини й машини. Мова Pascal була розроблена професором Ніклаусом Віртом (Швейцарська Вища технічна школа, м. Цюріх). Названо мову було на честь французького математика XVII століття Блеза Паскаля, що в 1640 році створив арифметичну (або рахункову) машину. Незважаючи на те, що він був розроблений як засіб для навчання техніці програмування, згодом він став популярним і поза сферою освіти.

На відміну від людського, мова ЕОМ не допускає двозначностей і невизначеностей. Кожна мова має строго певну граматику, названу синтаксисом. Якщо пропозиція програми (оператор) не відповідає синтаксису мови, то він не має змісту. З іншого боку, синтаксично правильний оператор має однозначне трактування.

Програма для ЕОМ – це просто послідовність операторів. Оператори являють собою команди ЕОМ, впливаючи яким вона вирішує завдання. Метод, використовуючи який програма вирішує завдання, називається алгоритмом; справа програміста вибрати алгоритм, щоб скласти по ньому програму. Алгоритм може являти собою просто математичне рівняння (наприклад, квадратне рівняння) або складну процедуру.

Внаслідок вивчення дисципліни студенти повинні знати: технічні засоби обчислювальних систем, мови програмування, методи алгоритмізації та програмування типових прикладних задач.

Студент повинен володіти навиками підготовки задач до рішення на ПЕОМ, розробки та відлагодження програм, використання засобів обчислювальної техніки у вирішенні наукових та інженерних задач.

По кожній лабораторній роботі студент повинен скласти **звіт**, якій містить в собі:

1. Назву роботи. Мету.
2. Алгоритм розв'язання задачі у вигляді блок-схеми.
3. Текст (лістинг) програми.
4. Відповідь на контрольні питання.

Оформлений звіт захищається студентом усно.

Варіант індивідуального завдання надається викладачем самостійно

### **Правила техніки безпеки та охорона праці**

Згідно з «Правилами техніки безпеки в лабораторіях інформатики» студентам забороняється:

- з'являтися та знаходитись приміщенні в нетверезому стані;
- ставити поруч з клавіатурою ємності з рідиною;
- перебувати в приміщенні в верхній одежі та завалювати нею робочі столи та стільці;
- працювати в лабораторії більше 6-ти годин на день (для вагітних жінок – більше 4-х годин);
- за власною ініціативою змінювати закріплені за ними робочі місця та знаходитись в приміщенні під час роботи іншої учбової групи;
- самостійно виконувати вмикання електроживлення лабораторії та заміну складових частин ПК, що вийшли із ладу.

У випадку виявлення несправностей обчислювальної техніки студент повинен сповістити про це викладача чи будь-кого з навчально-допоміжного персоналу лабораторії.

## Лабораторна робота №1

*Алфавіт, лексика, структура програми.*

*Організація введення/виводу. Програмування завдань лінійної структури.*

**Мета роботи:** Вивчення елементарних понять – алфавіт, зарезервовані слова, типи, змінні й т.п. Знайомство з основними етапами процесу розробки програми, з поняттям алгоритму, методами запису алгоритмів.

**Постановка завдання:** розробити алгоритм рішення завдання за заданим варіантом, скласти програму мовою Pascal, відлагодити її, виконати розрахунки для кількох варіантів вхідних даних, відповісти на контрольні питання.

### Теоретичні відомості:

Алфавіт мови Pascal включає букви, цифри, шістнадцятирічні цифри, спеціальні символи, пробіли й зарезервовані слова.

Букви - це букви латинського алфавіту від **a** до **z** і від **A** до **Z**, а також знак підкреслення “\_”. У Turbo Pascal немає розходження між прописними й малими літерами алфавіту, якщо тільки вони не входять у символні й строкові вираження.

Цифри – арабські цифри від 0 до 9.

Спеціальні символи Turbo Pascal - це символи + - \* / = , ' . : ; < > [ ] ( ) { } ^ @ \$ # . До спеціальних символів ставляться також наступні пари символів: < > <= >= := (\* \*) ( . ) У програмі ці пари символів не можна розділяти пробілами, якщо вони використовуються як знаки операцій відносини або обмежники коментарю.

Ідентифікатори в Turbo Pascal - це імена констант, змінних, міток, типів, об'єктів, процедур, функцій, модулів, програм і полів у записах. Ідентифікатор завжди починається буквою, за якої можуть слідувати букви й цифри. Буквою вважається також символ підкреслення, тому ідентифікатор може починатися цим символом і навіть складатися тільки з одного або декількох символів підкреслення. Пробіли й спеціальні символи алфавіту не можуть входити в ідентифікатор.

Зарезервовані слова використовуються для опису операторів, даних і інших язикових конструкцій. Зарезервовані слова не можуть використовуватись як ідентифікатори. Вони надають тексту програми більш «читабельний» вигляд, наближаючи його до тексту, написаному на природній англійській мові.

*Приклади правильних ідентифікаторів:*

a ALPHA MyProgram \_beta

*Приклади неправильних ідентифікаторів:*

1Program { починається з цифри }

block#1 { містить спеціальний символ }

My Prog { містить пробіл }

mod { зарезервоване слово }

У кожній програмі для обчислення потрібних результатів широко використовуються змінні. Це величина, значення якої змінюється в процесі роботи програми. Коли змінної привласнюється нове значення, її старе значення втрачається. Для оголошення змінної необхідно вказати ім'я змінної і її тип. Однотипні змінні можуть перераховуватися через кому перед вказівкою їхнього типу.

*Приклад оголошення змінних:*

a: integer;

b: byte;

c: boolean;

У прикладі використані змінні *a* типу Integer, *b* типу Byte і *c* типу Boolean. Тип визначає безліч припустимих значень, які може мати той або інший об'єкт, а також безліч припустимих операцій, які застосовні до нього. Перші два типи призначені для цілих даних, причому Integer може містити як додатні, так і від'ємні числа в діапазоні від  $-32768$  до  $+32767$ , а Byte тільки додатні в діапазоні від 0 до 255. Ці типи вважаються сумісними.

Константа – це величина, значення якої не змінюється в процесі роботи програми. Як константи в Pascal можуть використатися цілі, дійсні й шістнадцятиричні числа, логічні константи, символи, рядки символів.

Цілі числа записуються зі знаком або без нього за звичайними правилами.

Дійсні числа записуються зі знаком або без нього з використанням десяткової крапки й/або експонентної частини. Експонентна частина починається символом **e** або **E**, за яким можуть впливати знаки «+» або «-» і десятковий порядок. Символ **e** (**E**) означає десятковий порядок і має сенс «помножити на 1.0 у ступені». Наприклад,  $3.14E5$  - 3.14 помножити на 10 у ступені 5.

Pascal замислювався автором як навчальна мова структурного програмування. Як наслідок цього програми, на ньому написані, мають досить просту, але тверду структуру. Вона така:

{ частина оголошення ім'я програми: }

**Program** NameOfProgram;

{ частина оголошення списку модулів, що підключають: }

**Uses** { список модулів через кому };

```

{ частина оголошення констант:}
Const { список констант і їхніх значень через ; };
{ частина оголошення типів користувача:}
Type { список типів користувача через ; };
{ частина оголошення змінних:}
Var { список змінних і їхніх типів через ;}
{ частина основної програми }
Begin { крапка входу }
{ оператори основної програми }
End. { основна крапка виходу й кінець програми }

```

Частини Program і Uses не є обов'язковими, однак повинні бути першими і єдиними в програмі. Const, Type і Var частин у програмі може існувати безліч, їхній порядок не нормується й визначається вимогами програмування. Частина основної програми завжди є останньою, ознакою її завершення є ключове слово End із крапкою після нього. Будь-яка інформація після завершення основної частини ігнорується компілятором.

Для знайомства з мовою Pascal спробуємо скласти нескладну програму, що здійснює вивід якого-небудь повідомлення на екран ПК. Нехай це буде фраза «Я програмую на Турбо Паскалі» (фраза може бути іншою). Можливий варіант такої програми:

*Приклад:*

```

Program My_First_Program;
begin
  WriteLn('Я програмую на Турбо Паскалі');
end.

```

Тепер спробуйте виконати програму. Для цього послу набору її тексту натисніть **Ctrl-F9**. Якщо Ви не помилилися при уведенні тексту, то через кілька секунд побачите швидку зміну зображень на екрані: відразу після завантаження програми Pascal очищає екран, надаючи його в розпорядження працюючої програми користувача. Такий екран називається вікном програми. Якщо в тексті були помилки, виправте їх (див. **Додаток 4**) і запустите програму заново. Після завершення прогону (робота програми часто називається її прогоном або компіляцією) на екрані знову з'явиться вікно редактора з текстом програми. Якщо Ви не встигли розглянути зображення вікна програми, натисніть **Alt-F5**. Після натискання на будь-яку клавішу середовище поверне екран у режим відтворення вікна редактору.

Обговоримо її єдиний оператор

```
WriteLn();
```

Цікаво, що в Pascal взагалі й Turbo Pascal, зокрема, немає спеціальних операторів вводу-виводу. Для обміну інформацією з навколишнім світом у програмах, написаних мовою Pascal, використовуються спеціальні стандартні процедури. Таким чином, по своїй суті оператор *WriteLn()* є оператором звертання до вбудованої процедури виводу даних (своя назва вона одержала від *WriteLine* - записати рядок).

Поняття процедури - одне із центральних понять Pascal. Процедура - це деяка послідовність операторів, до якої можна звернутися по імені. Щоразу, коли ми називаємо в операторі ім'я процедури, ініціюється послідовність запрограмованих у ній дій.

Процедура (або оператор) *Write* здійснює вивід на екран або друкувальний пристрій. Оператор може виводити повідомлення або значення змінної. Повідомлення записуються в апострофах. Для виводу значення змінної вказується ім'я змінної. Повідомлення й змінні можна чергувати в одному списку, розділяючи комами. Курсор залишається за останнім виведеним даним. Оператор *writeln* виконує аналогічні дії й переводить курсор на наступний рядок. Оператори *write* і *writeln* допускають вивід даних із форматкуванням.

```
Write(a:5:2);
```

Перше число вказує, скільки екранних знаків приділяється під вивід. Друге - вказує кількість знаків після коми в числі й може бути відсутнє.

Оператор *read* здійснює введення даних з клавіатури. Дані, що вводять, розміщуються як значення змінних, імена яких перераховані в круглих дужках за оператором *read*.

```
Read (a, b, c);
```

Уводяться дані теж списком, у якому вони розділяються пробілом. Уведення закінчується натисканням Enter. Курсор, що відзначає позицію наступного уведення/виводу, залишається за останнім уведеним даним. Оператор *readln* виконує аналогічні дії й переводить курсор на наступний рядок.

Частина програми, що виконується, складається з одного й більше операторів. Оператор описує деяку дію, що повинна виконати програма. Переважна більшість операторів містить зарезервовані слова, що пояснюють відповідну дію.

Оператор присвоєння - один з найбільш часто використовуваних операторів обробки даних. Його вигляд:

```
Змінна := вираз;  
Y:= sin(x);
```



І змінна, і вираз повинні бути того самого типу або належати до сумісних типів. Знак операції складається із двох символів, які не можна розривати пробілом.

Складений оператор. Якщо необхідно при деякій умові виконати відразу групу операторів, їх поєднують в одному складеному операторі за допомогою операторних дужок begin – end.

Те, із чим ми мали справу вище, лише умовно можна назвати програмою. Спробуємо створити щось більш складне й корисне.

*Приклад*: розробити програму, яка обчислить значення Y:

$$Y = \frac{\ln(1 + a\sqrt{1 + a^2})}{b + \sqrt{1 + \sin(c)}} \text{ при цьому, } b = 3,18; \text{ а й } c \text{ необхідно вводити із клавіатури}$$

Program Primer1; {Оголошення імені програми, це - Primer1}

//не забуваємо про те, що кожний рядок програми повинен закінчуватися крапкою з комою, крім того, текст програми може супроводжуватися коментарями, укладеними у фігурні дужки або відділеними // .

Var a, c, Y: real; {перелік із трьох змінних, використовуваних у програмі з оголошенням типу}

Const b=3.18; {оголошення постійної, використовуваної в програмі}

Begin {початок програми, після нього крапка з комою не ставиться!}

Write('Введіть значення A:'); {Вивід на екран текстового повідомлення, пропозиція ввести A}

Read (A); {Оператор читання, у цьому випадку служить для уведення значення змінної A с клавіатури. На практиці - програма припиняє роботу й від користувача програми потрібно ввести значення відповідних змінних.}

Write('Введіть значення C: ');

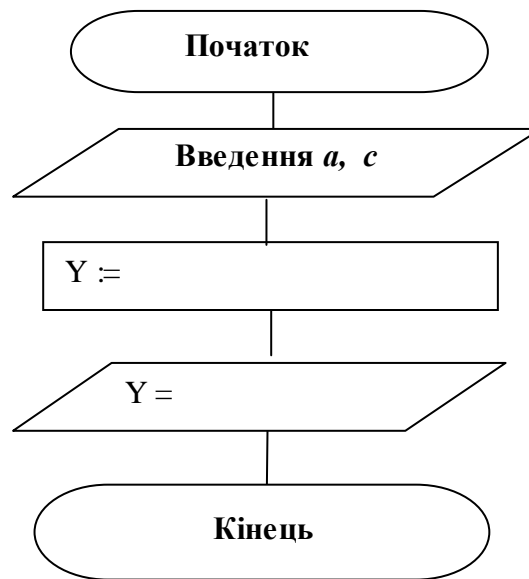
Read (c); {Введення змінної C}

Y:= (ln(1+a\*sqrt(1+sqr(a)))/(b+sqrt(1+sin(c)))) { := оператор присвоєння. В правій частині стоїть вираз, якій необхідно обчислити. Потім отримане значення привласнюється змінній, що стоїть в лівій частині оператора}

Writeln('Y=', Y); {Виводимо на екран значення Y; спочатку йде повідомлення в апострофах, символи між ними будуть виведені на екран у тім виді, у якому вони представлені в тексті програми, після йде вивід знайденого значення}

End. {Кінець програми, після нього завжди ставиться крапка}

Схема алгоритму рішення такого завдання буде мати такий вигляд:



**Варіанти завдань для самостійної роботи:**

1. Дано два числа  $a$  і  $b$ . Одержати їхню суму, різницю й добуток.

2. Дані  $x, y, z$ . Обчислити  $a, b$ , якщо:

$$a = \frac{\sqrt{|x-1|} - \sqrt{|y|}}{1+x^2/2+x^2/4}, b = x(\arctg(z) + e^{-(x+3)})$$

3. Дано дійсні числа  $x$  і  $y$ . Одержати  $(|x| - |y|) / (1 + |x*y|)$ .

4. Дані  $x, y, z$ . Обчислити  $a, b$ , якщо:

$$a = \frac{3 + e^{y-1}}{1+x^2|y-tg(z)|}, b = 1 + |y-x| + \frac{(y-x)^2}{2} + \frac{(y-x)^3}{3}$$

5. Дано довжину ребра куба. Знайти площу грані, площа повної поверхні й об'єм цього куба.

6. Дано  $x, y, z$ . Обчислити  $a, b$ , якщо:

$$a = (1+y) \frac{x+y/(x^2)+4}{e^{-x-2}+1/(x^2+4)}, b = \frac{1+\cos(y-2)}{x^4/2+\sin^2 z}$$

7. Дано два дійсних додатних числа. Знайти середнє арифметичне й середнє геометричне цих чисел.

8. Дано  $x, y, z$ . Обчислити  $a, b$ , якщо:

$$a = y + \frac{x}{y^2 + \left| \frac{x^2}{y + x^3/3} \right|}, b = (1 + \operatorname{tg}^2 \frac{z}{2})$$

9. Дано катети прямокутного трикутника. Знайти його гіпотенузу й площу.

10. Дано  $x, y, z$ . Обчислити  $a, b$ , якщо:

$$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, b = 1 + \frac{z^2}{3 + z^2/5}$$

11. Визначити периметр правильного  $n$ -кутника, описаного біля окружності радіуса  $r$ .

12. Дані  $x, y, z$ . Обчислити  $a, b$ , якщо:

$$a = \frac{1 + \sin^2(x + y)}{2 + \left| x - 2x/(1 + x^2 y^2) \right|} + x, b = \cos^2(\operatorname{arctg} \frac{1}{z})$$

13. Дано сторону рівностороннього трикутника. Знайти площу цього трикутника.

14. Дано  $x, y, z$ . Обчислити  $a, b$ , якщо

$$a = \ln \left| (y - \sqrt{|x|}) \left( x - \frac{y}{z + x^2/4} \right) \right|, b = x - \frac{x^2}{3} + \frac{x^3}{4}$$

### **Контрольні питання:**

1. Яка структура програми в мові Pascal ?
2. Що таке ідентифікатор?
3. Чим відрізняються константи й змінні? Як вони оголошуються?
4. Що таке зарезервоване слово?
5. Що таке тип даних? Назвіть, які типи даних ви знаєте?
6. Укажіть, як працюють оператори введення/виводу?
7. Чим відрізняється *write* від *writeln* і *read* від *readln*?
8. Які дії реалізує оператор присвоєння?
9. Для чого використовується складений оператор?
10. Як записується коментар в програмі?

## Лабораторна робота №2

### Алгоритмічна структура - розгалуження. Умовний оператор, оператор вибору.

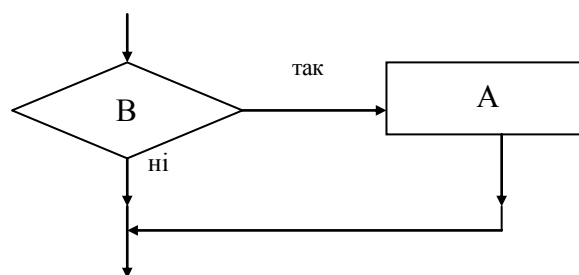
**Мета роботи:** Знайомство з однією з основних алгоритмічних структур – розгалуженням. Вивчення особливостей умовного оператора й оператора вибору.

**Постановка задачі:** розробити алгоритм рішення завдання за заданим варіантом, скласти програму мовою Pascal, відлагодити її, виконати розрахунки для кількох варіантів вхідних даних, відповісти на контрольні питання.

#### Теоретичні відомості:

На практиці рідко зустрічаються задачі лінійної структури. Набагато частіше, залежно від яких-небудь проміжних результатів, необхідно організувати обчислення логічної структури. У мові Pascal є дві реалізації однієї з основних алгоритмічних структур - розгалуження – умовний оператор і оператор вибору. Ці оператори дозволяють на основі аналізу деякої умови вибрати потрібний варіант продовження обчислювального процесу. До операторів розгалуження відносять умовний оператор if та оператор вибору case.

Умовний оператор може мати дві форми: коротку й повну. Алгоритмічна схема короткої форми:



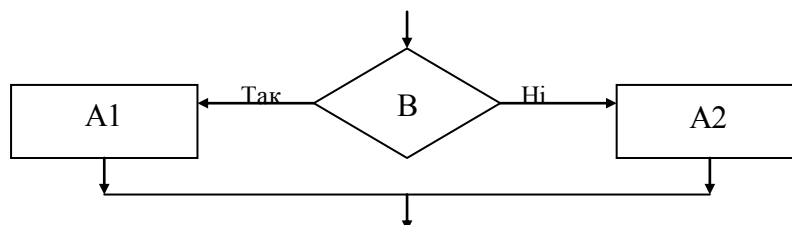
Цієї схемі відповідає такий програмний код:

If B then A;

B – умовний вираз, A - будь-який оператор Турбо Паскаля.

При виконанні оператора аналізується вираз B. Якщо він має значення True, виконується оператор A, у протилежному випадку нічого не відбувається й умовний оператор завершує свою роботу.

Алгоритмічна схема повної форми умовного оператора:



Цієї схемі відповідає такий програмний код:

```
If B then A1
      else A2;
```

B - умовний вираз, A1, A2 - будь-які оператори Pascal.

При виконанні оператора аналізується вираз B. Якщо він має значення True, виконується оператор A1, у противному випадку виконується оператор A2.

Умовні оператори можуть бути вкладеними, тобто на місці оператора A в короткій формі й A1(A2) у повній формі можуть стояти умовні оператори. При вкладенні умовних операторів різних форм може виникнути ситуація, коли частин else менше частин then.

*Наприклад:*

```
If X > Y then
  If A = 0 then B:= 0
      else B:= 1;
```

у цьому випадку вважається, що частина else зв'язується з найближчою по ходу написання програми частиною then, що не має частини else. Для наведеного приклада при виконанні умови  $X > Y$  змінна B одержить значення 0 або 1 залежно від виконання умови  $A=0$ . Однак якщо умова  $X > Y$  не виконується, другий умовний оператор не буде працювати й значення B не зміниться.

**Оператор вибору** дозволяє вибрати одне з декількох можливих продовжень програми. Параметром, по якому здійснюється вибір, служить ключ вибору - вираження будь-якого порядкового типу (цілого, символного, логічного).

Структура оператора вибору така:

```
Case <ключ вибору> of <список вибору> [else <оператори>] end
```

case, of, else, end - зарезервовані слова (вибрати, з, інакше, кінець).

Оператор вибору працює в такий спосіб. Спочатку обчислюється значення вираження <ключ вибору>, а потім у послідовності операторів <список вибору> відшукується такий, перед яким йшла константа, рівна обчисленому значенню. Знайдений оператор виконується, після чого оператор вибору завершує свою роботу. Якщо в списку вибору не буде знайдена константа, що відповідає обчисленому значенню ключа вибору, керування передається операторам, що йдуть за словом else. Частина else <оператори> може бути відсутня. Тоді при відсутності в списку вибору потрібної константи нічого не відбудеться, і оператор вибору просто завершить свою роботу.

*Приклад.* Необхідно розробити програму, що буде визначати, який сьогодні день тижня:

```

Program Nedelya;
Var x: integer;
Begin
  Writeln('Введіть номер дня тижня ');
  Read(x);
  Case x of
    1: writeln ('Сьогодні понеділок');
    2: writeln ('Сьогодні вівторок');
    3: writeln ('Сьогодні середа');
    4: writeln ('Сьогодні четвер');
    5: writeln ('Сьогодні п'ятниця');
    6: writeln ('Сьогодні субота');
    7: writeln ('Сьогодні неділя');
  end;
end.

```

На закінчення розглянемо приклад, що знаходить значення функції  $Y$ , використовую умовний оператор if:

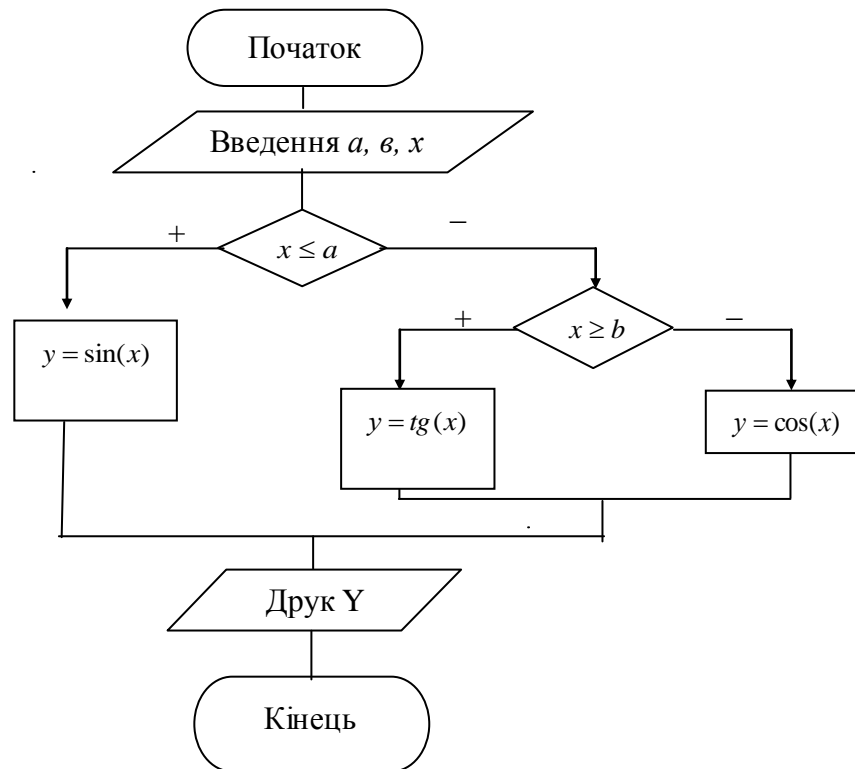
$$Y = \begin{cases} \sin x, & \text{якщо } x \leq a, \\ \cos x, & \text{якщо } a < x < b, \\ \operatorname{tg} x, & \text{якщо } x \geq b. \end{cases}$$

```

Program Primer2;
var x,y: real; { об'явлення початкових даних}
    a, b: integer;
begin
  writeln(' Введіть значення x, a і b '); { друк підказки для введення даних}
  readln ( x, a, b ); {введення початкових даних }
  if x <= a {перевірка першої умови задачі, якщо умова істинна –
обчислюється значення функції}
  then y := sin(x)
  { якщо умова помилкова - перевірка другої умови }
  else if x <= b
  then y:= cos(x) {обчислення значення функції Y }
  else y:= sin(x)/cos(x); {обчислення іншого значення функції Y}
  writeln ('Y=', y); { друк результату }
End.

```

Алгоритмічна схема рішення буде мати такий вигляд:



### **Варіанти завдань для самостійної роботи:**

1. Дано дійсні числа  $x, y$ . Отримати:  $\max(x, y)$ ;  $\min(x, y)$ ;
2. Дано дійсні числа  $x, y, z$ . Отримати:  $\max(x, y, z)$ ;
3. Дано дійсні числа  $x, y, z$ . Отримати:  $\min(x, y, z)$ ;
4. Дано дійсні числа  $a, b, c$ . Подвоїти ці числа, якщо  $a > b > c$ , і замінити їхніми абсолютними значеннями, якщо це не так.
5. Дано дійсні числа  $x, y$ . Обчислити  $z$ :

$$z = \begin{cases} x - y, & \text{коли } x > y \\ y - x + 1, & \text{в іншому випадку} \end{cases}$$

6. Дано два дійсних числа. Вивести перше число, якщо воно більше другого, і два числа, якщо це не так.
7. Дано два дійсних числа. Замінити перше число нулем, якщо воно менше або дорівнює другому, і залишити без зміни в протилежному випадку.

8. Дано дійсні числа  $x, y$ . Менше із цих двох чисел замінити їхньою напівсумою, а більше - їхнім подвоєним добутком.
9. Визначити, яка із двох фігур - коло або квадрат має більшу площу. Відомо, що сторона квадрата дорівнює  $a$ , радіус кола  $r$ . Вивести на печатку назву й значення площі більшої фігури.
10. Дано дійсне число  $a$ . Обчислити  $f(x)$ , якщо

$$a) \quad f(x) = \begin{cases} x^2, & \text{коли } -2 \leq x \leq 2 \\ 4, & \text{в іншому випадку} \end{cases}$$

$$b) \quad f(x) = \begin{cases} x^2 + 4x + 5, & \text{коли } x \leq 2 \\ \frac{1}{x^2 + 4x + 5}, & \text{в іншому випадку} \end{cases}$$

$$c) \quad f(x) = \begin{cases} 0, & \text{коли } x \leq 0 \\ x, & \text{коли } 0 \leq x \leq 1 \\ x^2, & \text{в іншому випадку} \end{cases}$$

$$d) \quad f(x) = \begin{cases} 0, & \text{якщо } x \leq -5 \\ x^2 - x, & \text{якщо } -5 \leq x \leq 5 \\ x^2 - \sin x^2, & \text{в іншому випадку} \end{cases}$$

### **Контрольні питання:**

1. Які види розгалужень реалізовані в Turbo Pascal?
2. Для чого використовуються розгалуження? Наведіть приклади.
3. Які форми умовних операторів вам відомі?
4. Як працює умовний оператор?



5. Чи можуть умовні оператори бути вкладеними?
6. Укажіть структуру оператора вибору.