

Confidential

О.В. Костриба

ОСНОВИ ПРОГРАМУВАННЯ

Частина 2



2008



**Основи програмування. Частина 2. TurboPascal 7.0.
/ Костриба О.В. — Білогір'я, Хмельницька область, 2008.**

Схвалено радою районного методичного кабінету №5 від 27.02.2008р.

Упорядник:

**Костриба О.В. – вчитель інформатики Білогірського
НВК.**

У посібнику розміщено теоретичний матеріал та тексти програм з поясненням. Мова програмування – TurboPascal 7.0.

Даний посібник допоможе вчителю підготуватися до уроку та зробити викладання нового матеріалу цікавим і насиченим. Різноманітний добір завдань дозволить оцінити рівень навчальних досягнень учнів.

Посібник може бути корисний для вчителів інформатики, студентам, учням.

На диску розміщено готові проекти, розроблені в середовищі TurboPascal 7.0.

Мова програмування. Середовище програмування.

Мова програмування Паскаль є однією з найбільш поширених мов програмування. Була створена в 1968-1971 рр. швейцарським професором Ніклаусом Віртом, як засіб для навчання програмування. Завдяки принципам структурного програмування і покрокової деталізації, програми мовою Паскаль більш зрозумілі ніж програми іншими мовами.

Мова Паскаль є структурованою мовою програмування, оскільки використовує тільки 4 основні структури:

- 1) вказівки безумовного виконання;
- 2) вказівки розгалуження;
- 3) вказівки повторення;
- 4) виділення допоміжних програм.

Головною відмінністю мови від інших неструктурних мов є деякі обмеження використання вказівок безумовного переходу.

Весь матеріал з мови Turbo Pascal викладений стосовно версій Turbo Pascal 6.0-7.0. В новій версії Turbo Pascal 7.0 є деякі доповнення до попередніх версій що зрівняло її по можливостях з мовою C++.

Програма - впорядкована послідовність дій для ПК, використання якої реалізує алгоритм розв'язку будь-якої задачі.

Транслятор - програма, яка перекладає програму з мови програмування в машинний код.

Переведення програм в машинні коди та їх виконання може здійснюватися шляхом інтерпретації або компіляції.

Під час *інтерпретації* кожна вказівка програми аналізується на предмет виявлення в ній синтаксичних і логічних помилок, переводиться в машинні коди, а потім виконується. Тобто виконання програми проходить шляхом їх покомандної інтерпретації.

Під час *компіляції* в усьому тексті виявляються синтаксичні та логічні помилки, після цього вся програма переводиться в машинний код і лише тоді виконується.

Система програмування - частина базового програмного забезпечення, яка підтримує процес програмування на ПК.

Система програмування містить:

- інтерпретатор (для виконання програм);
- редактор (для складання і введення програм);
- транслятор або компілятор (для перекладу програм з вхідної мови на машинну).

Система програмування Turbo Pascal об'єднує в собі:

- текстовий редактор;
- компілятор;
- компоновник;
- відлагоджувач;
- систему підказки.

Система програмування Turbo Pascal складається з великої кількості файлів. Головні з них:

- 1) Turbo.exe - забезпечує роботу в середовищі програмування;
- 2) Turbo.tpl- містить бібліотеку стандартних підпрограм;
- 3) Turbo.hlp-забезпечує введення на екран підказки.

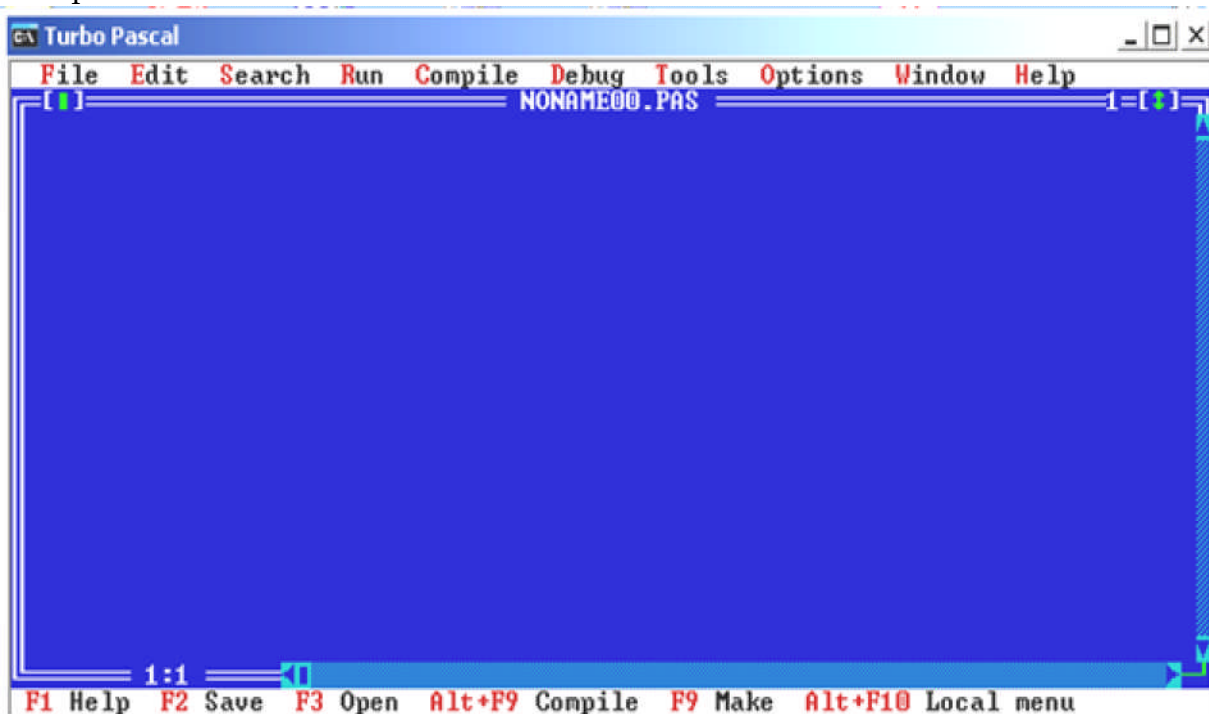


Поняття про середовище програмування TurboPascal.

Щоб увійти в середовище TurboPascal, потрібно:

1. Ввійти в каталог системи програмування.
2. Запустити на виконання командний файл turbo.exe.

На екрані з'явиться головне меню системи.



У верхньому рядку подано команди головного меню, а в нижньому - рядок статусу, в якому перелічені імена функціональних клавіш, призначених для виконання деяких операцій в конкретній ситуації.

Щоб перейти в головне меню, потрібно натиснути клавішу *F10*. Щоб вийти з головного меню і повернутись в редактор тексту - *Esc*.

Виконати команду означає: з допомогою клавіш управління курсором виставити курсор на потрібну команду і натиснути клавішу *Enter*. (Виконати команду можна за допомогою "миші" або комбінацією *Alt + буква*, яка виділена в назві команди).

Кожна команда головного меню має одне або декілька підменю.

Типовий порядок створення нового файлу:

- увійти в головне меню;
- виконати команду *File*;
- вибрати і виконати команду *New*. На екрані з'явиться порожнє вікно редактора з умовною назвою *Noname.pas*;
- набирати текст програми. В кінці кожного рядка натиснути *Enter*;
- виконати програму. Натиснути *Ctrl + F9*. Або увійти в головне меню і вибрати і виконати команду *Run*;
- виправити при необхідності помилки;
- знову виконати програму;
- переглянути результати, натиснувши *Alt + F5*;
- записати програму в файл на диск, увійти в головне меню виконати команду *File*;
- Виконати команду *Save*. Ввести ім'я і натиснути *Enter*.

Введення, виведення даних. Вказівка присвоєння.**Теоретичний матеріал****Алфавіт мови Turbo Pascal**

Алфавіт мови Turbo Pascal:

1) 26 великих і малих букв латинського алфавіту, знак підкреслення

2) Цифри 0...9

3) Спеціальні символи:

+	<	@	`
-	>	.	#
*	[]	,	\$
/	()	:	^
=	{ }	;	

Проміжок - незаповнена позиція в тексті, займає одну позицію.

4. Комбінації спеціальних символів утворюють складені символи:

:=	→ присвоєння	(.)	→ альтернатива{ }	(.)	→ альтернатива[]
..	→ діапазон значень	<>	→ не дорівнює	<=	→ менше або дорівнює
>=	→ більше або рівне				

5. Службові слова.

Ідентифікатори

Для позначення констант, змінних, типів, процедур і функцій, файлів і програм використовують *імена (ідентифікатори)*.

Правила запису ідентифікаторів:

1. Ідентифікатори починаються тільки з букви або знаку підкреслення.
2. Ідентифікатор може складатися з букв, цифр, і знаку підкреслення.
3. Між двома ідентифікаторами повинен бути хоча б один проміжок.
4. Максимальна довжина ідентифікатора 127 символів, але ЕОМ розрізняє тільки 63.
5. Ідентифікатор не повинен співпадати з службовими словами.
6. Великі і малі букви не розрізняються.

Константи і змінні

Константи - величини, значення яких встановлені в описовій частині програми і в процесі виконання програми не змінюються.

Для оголошення констант в програмі використовується службове слово - **CONST**.

CONST < Ідентифікатор> = < значення константи>;

Наприклад:

```
CONST a = 13.5;
      f1 = 15;
```

Константам в програмі не можна присвоювати нові значення після того як вони були описані.

Змінні - величини значення яких змінюється в процесі виконання програм.

Для опису змінних використовується службове слово **VAR**.

VAR <список ідентифікаторів>: < тип>;

Наприклад:

```
VAR i,x,r1,d : real;
    t,ca : integer;
```

Тип - множина значень однакової природи разом із набором операцій, які над ними виконуються.



Кожна змінна повинна бути описана тільки один раз на початку програми після слова **VAR**

Скалярні типи даних

Цілі або цілочисельні типи

Тип	Діапазон
byte	0...255
shortint	-128...127
integer	-32768...32767
word	0...65535
longint	-2147483648...2147483647

При виході значень даних цілого типу за вказаний діапазон помилки виконання програми не буде, але результат виводиться неправильний.

Дійсні типи

Тип	Діапазон
Real	2.9*10E-39...1.7*10E38
Singl	1.5*10E-45...3.4*10E38
double	5.0*10E-324...1.7*10E308
extended	1.9*10E-4951...1.1*10E4932
comp	-2E+63+1...2E+63-1

Літерний тип

Тип	Діапазон
Char	кодова таблиця ПЕОМ

Булевий тип

Тип	Діапазон
boolean	True , False

Структуровані типи даних

Рядковий тип.

Рядок - послідовність символів кодової таблиці ЕОМ.

Для опису даного типу використовується ідентифікатор **String**, після якого в квадратних дужках записується значення максимальної довжини рядка. Якщо значення не вказується, то довжина рядка дорівнює 255 байт.

Наприклад: Var a,c :String[25];
v : String;

Інші типи даних.

В Паскаль використовуються такі типи даних: масиви, множини, записи, файли, процедурні типи, об'єкти. Всі вони потребують окремого детального вивчення.

Арифметичні операції

Цілі типи

На множині цілих чисел визначені такі операції:

- + → додавання; * → множення;
- → віднімання; / → ділення;
- Div → ділення націло; Mod → остача від ділення націло;

Математичні функції:

- ABS(x) → |x|
- ARCTAN(x) → arctg(x)
- COS(x) → cos(x)
- SIN(x) → sin(x)
- EXP(x) → e^x

LN(x) → ln(x)

SQR(x) → x²

SQRT(x) → \sqrt{x}

Результат виконання операції + ; - ; * ; Div ; Mod ; Abs(x); Sqr(x) над цілими числами є цілим числом.

Результат виконання операції /, а також всіх інших математичних функцій на множині цілих чисел, є дійсним числом.

Приклад: При діленні -25/5 отримуємо не ціле число, а дробове число -5.0

1. DIV - цілочисельне ділення.

Наприклад: -25 Div 5.

Між знаком Div і числами, які беруть участь у діленні, повинно бути хоча б по одному проміжку.

2. MOD - остача від цілочисельного ділення.

Наприклад. 35 Mod 6; -1 Mod 2.

Значення відповідних виразів відповідно дорівнюють: 5; -1.

3. RANDOM(x) - випадкове число на проміжку - 0...X.

Наприклад. for I:=1 to 4 do
write(RANDOM(10):2);

Результат: 7 3 8 1

Дійсні типи

Дійсні числа - це десяткові дроби і, в окремому випадку, цілі числа, записані у вигляді десяткового дробу.

Наприклад.

Числа 5; 0 - цілі числа,

Числа 5.0;0.0 - дійсні числа.

Дійсні числа можуть бути записані двома способами:

- з фіксованою крапкою (5.45; 9.23);

- з плаваючою крапкою (3E+5; -8.1E-4);

Запис числа з фіксованою крапкою.

Ціла і дробова частини дійсного числа розділяються десятковою крапкою, а не комою!

Наприклад: 0.39; -37.4

Число не може починатися з крапки і не може нею закінчуватися. Записи 0. і .89 недопустимі.

Запис числа з плаваючою крапкою.

У випадку, коли необхідно записати число в стандартному вигляді, використовується буква E, справа від якої знаходиться показник степеня (8.76*10⁻² - 8.76E-02). Букву E, яка входить до складу числа, потрібно читати, як "помножити на 10 в степені...".

Для дійсних чисел визначені такі операції:

+ → додавання; * → множення;

- → віднімання; / → ділення;

Математичні функції:

ABS(x) → |x|

ARCTAN(x) → arctg(x)

COS(x) → cos(x)

SIN(x) → sin(x)

EXP(x) → e^x

LN(x) → ln(x)

SQR(x) → x²

SQRT(x) → \sqrt{x}



Результатом виконання даних операцій та функцій є дійсне число. Крім даних функцій для дійсних чисел визначені також наступні функції:

INT(X)	→	задає цілу частину числа x.
TRUNC(X)	→	відкидає дробову частину числа x.
FRAC(X)	→	задає дробову частину числа x.
ROUND(X)	→	заокруглює число x.

Літерний тип

Функції для роботи з літерним типом:

ORD(S) - визначає код символу з заданої кодової таблиці. Результат даної функції належить до цілого типу.

CHR(I) - визначає символ коду якого дорівнює I. Результат даної функції належить до літерного типу.

SUCC(S) - визначає символ, який знаходиться після символу S в кодовій таблиці.

PRED(S) - визначає символ, який знаходиться перед символом S в кодовій таблиці.

Булевий тип

$$ODD(X) = \begin{cases} TRUE, & \text{якщо } X \text{ непарне} \\ FALSE, & \text{якщо } X \text{ парне} \end{cases}$$

Результат даної функції належить до булевого типу

Результат виконання операцій порівняння та операцій відношення над даними одного типу належить до булевого типу.

Арифметичні вирази. Пріоритет операцій.

Арифметичний вираз задає послідовність виконання дій над елементами даних.

Виконання кожної операції здійснюється в порядку врахування її пріоритету:

1. Виконуються дії в дужках
2. Обчислюються стандартні математичні функції
3. Виконуються дії - @, NOT
4. *, /, div, mod, and
5. +, -, or, xor
6. =, <>, <, >, <=, >=

де not(ні), and(і), or(або, чи), xor -логічні операції.

Операції з рівним пріоритетом виконуються зліва направо з можливістю регулювання порядку їх виконання за допомогою дужок.

Приклади виразів:

МАТЕМАТИКА	TURBO PASCAL
$ax^2 + b$	$a*Sqr(x)+b$
$\frac{a + b}{ c - d }$	$(a+b)/Abs(c-d)$
$\sqrt{x * 2 + 1}$	$Sqrt(x*2+1)$
$e^x \ln(x+2)$	$Exp(x)*Ln(x+2)$
a^b	$Exp(b*Ln(a))$

Вказівка присвоювання

Вказівка присвоювання має вигляд:

<ім'я змінної>:=<вираз>

Змінна і вираз повинні бути одного типу. Виняток становить випадок дійсної змінної, коли вираз може бути і цілого типу.

Виконується вказівка присвоєння таким чином: спочатку обчислюється значення виразу в правій частині вказівки присвоювання і це значення присвоюється змінній в лівій частині (заноситься в пам'ять ЕОМ).

Наприклад: $y:=sqr(x)+2*x/(x-3.14)$

Програма — алгоритм розв'язання конкретної задачі записаний мовою програмування. Вона задає послідовність виконання конкретних вказівок для досягнення поставленої мети.

Структура програми.

Програма складається з заголовку і головної частини, яка в свою чергу поділяється на розділи.

Заголовок програми починається службовим словом **PROGRAM** та імені програми.

Наприклад:

```
program a1;
program grafik;
```

Ім'я програми може бути довільним і записується згідно правил запису ідентифікаторів.

Головна частина програми може містити від одного до семи розділів, які записуються в наступному порядку:

USES ім'я модуля; - розділ підключення модулів системної бібліотеки.

Модуль - це бібліотека констант, типів даних, змінних, підпрограм, спеціалізованих процедур і функцій. В мові Паскаль можна використовувати наперед визначені модулі (**CRT**, **DOS**, **GRAPH**, **OVERLAY**, **PRINTER**, **SYSTEM**, **TURBO3**, **GRAPH3**), а також власні.

crt - набір програм для управління екраном, вікнами, звуком.

dos - підтримує деякі функції MS DOS.

graph - підтримує графічний режим.

printer - керує друкуючим пристроєм.

LABEL ...; - розділ опису міток

CONST ...; - розділ опису констант

TYPE ...; - розділ опису типів даних

VAR ...; - розділ опису змінних

PROCEDURE (FUNCTION) ім'я процедури (функції): - розділ опису процедур і функцій

Begin

...; - розділ вказівок програми (тіло програми)

...;

End.

Службові слова - *Begin* і *End* називають операторними дужками, вони виконують таку саму роль, що і круглі дужки в математиці розділяючи групи вказівок.

Розділ **Uses** завжди записується після заголовка програми. Інші розділи опису можуть зустрічатися в програмі по декілька раз і в довільному порядку (всі описи об'єктів роблять до їх використання в програмі). Окремі розділи програми можуть бути відсутніми крім розділу вказівок.

Між будь-якими вказівками (групами вказівок) ставиться крапка з комою, яка відокремлює їх одну від одної. В програмі використовуються коментарі, які беруться в фігурні дужки. Коментарі дозволяють краще зрозуміти програму не впливаючи на її виконання. Програма закінчується словом **End** після якого ставиться крапка.

Вказівка введення даних

Виведення даних виконується вказівками **Read** і **Readln**.

Загальний вигляд:

```
Read(список);
```

```
Readln(список);
```

В списку перелічуються через кому імена змінних, значення яких вводяться з клавіатури.



Після введення значення змінної натискається “Enter”. Потім значення другої і знову “Enter” і т.д. Можна записати значення змінних в одному рядку відділивши їх проміжками і натиснути “Enter”.

Різниця між вказівками `Read` і `Readln` полягає в тому, що при виконанні вказівки `Read` значення змінних читаються з одного рядка (для зчитування даних з файла), а при виконанні вказівки `Readln` значення кожної змінної читається з іншого рядка, тобто після зчитування значення першої змінної буде зроблено перехід у наступний рядок і т.д (для зчитування даних з файла).

Наприклад:

```
Read(a,b,c);  
Readln(f,g);
```

Вказівка виведення даних

Виведення даних виконується вказівками `Write` і `Writeln`.

Загальний вигляд цих вказівок:

```
Write(список);  
Writeln(список);
```

У списку перелічується через кому імена змінних і т.д. Якщо у списку є вираз, то він попередньо обчислюється, а вже потім результат виводиться на екран.

Різниця між вказівками `Write` і `Writeln` полягає в тому, що при виконанні вказівки `Write` значення змінних виводиться в одному рядку, так як вони записані, а при виконанні вказівки `Writeln` значення кожної змінної виводиться в окремому рядку, тобто після виведення значення першої змінної буде зроблено перехід у наступний рядок і т.д.

Для зручності введення даних користуються поєднанням вказівок `Read` і `Write`.

Друк результатів за допомогою принтера в цілому співпадає з виведенням даних на екран, тільки у вказівках `Write` або `Writeln` потрібно записати ім'я файла `Lst` і в розділі `Uses` вказати ім'я модуля `Printer`.

Існує ще один спосіб керування виведенням інформації — можна вказувати ширину поля для кожного елементу виведення, форму запису числових значень. Здійснюється це з допомогою двох цілих чисел, які записуються в вказівці `Write` після відповідного елементу виведення і відділяється від нього і між собою двокрапкою “:”.

Наприклад:

```
Write(f,a);  
Writeln(b:8:5);  
Writeln(b:5);  
Writeln('y=',y:4:2);  
Writeln('сума =',c);
```

Приклади розв'язку задач:

1. Задано сторону квадрату. Скласти програму для знаходження його периметру: $P = 4 \cdot a$

Program Begin_1;

```
var
  a:real;
  p:real;
begin
  write('Введіть довжину сторони(m) = ');
  readln(a);
  p:=4*a;
  writeln('P=',p:0:2,'м.');
```

end.

Ім'я програми

Розділ оголошення змінних

Змінна: a – дійсного типу

Змінна: p – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення периметру квадрату

Виведення на екран отриманого результату

Кінець програми

2. Задано сторони прямокутника. Скласти програму для знаходження його площі: $S = a \cdot b$, та периметру: $P = 2 \cdot (a + b)$.

Program Begin_2;

```
var
  a,b:real;
  p,s:real;
begin
  write('Введіть довжин сторін a і b (м) = ');
  readln(a,b);
  p:=2*(a+b);
  S:=a*b;
  writeln('P=',p:0:2,'м.');
```

writeln('S=',S:0:2,'м.кв.');

end.

Ім'я програми

Розділ оголошення змінних

Змінні: a,b – дійсного типу (аргументи)

Змінна: P,S – дійсного типу (результати)

Початок програми

Вивід текстової підказки на екран

Введення числових значень через "пропуск"

Обчислення периметру

Обчислення площі прямокутника

Виведення на екран отриманого значення периметру

Виведення на екран отриманого значення площі

Кінець програми

3. Задано діаметр кола. Скласти програму для знаходження довжини кола: $L = \pi \cdot d$ та площі круга $S = \frac{\pi \cdot d^2}{4}$ обмеженого даним колом.

Program Begin_3;

```
const
  pi=3.14;
var
  d:real;
  L,S:real;
begin
  write('Введіть діаметр кола(m) = ');
  readln(d);
  L:=pi*d;
  S:=pi*sqr(d)/4;
  writeln('L=',L:0:2,'м.');
```

writeln('S=',S:0:2,'м.кв.');

end.

Ім'я програми

Розділ оголошення констант – постійних величин

Константі PI присвоюється значення 3.14

Розділ оголошення змінних

Змінна: d – дійсного типу

Змінні: S,L – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення довжини кола

Обчислення площі круга

Виведення на екран значення змінної → L

Виведення на екран значення змінної → S

Кінець програми



4. Задано довжину ребра куба – a . Скласти програму для знаходження його об'єму: $V = a^3$ та площі поверхні: $S = 6 \cdot a^2$.

Program Begin_4;

```
var
  a:real;
  V,s:real;
begin
  write('Введіть довжину ребра куба (м) = ');
  readln(a);
  V:=a*a*a;
  S:=6*SQR(a);

  writeln('V=',V:0:2,'м.куб. ');
  writeln('S=',S:0:2,'м.кв. ');

end.
```

Ім'я програми

Розділ оголошення змінних

Змінні: a – дійсного типу (аргумент)Змінна: V, S – дійсного типу (результати)**Початок програми**

Вивід текстової підказки на екран

Введення числового значення довжини ребра куба

Обчислення об'єму

Обчислення площі поверхні. SQR – піднесення до квадрату

Виведення на екран отриманого значення об'єму

Виведення на екран отриманого значення площі

:0:2 – кількість символів на все число: кількість символів після коми

Кінець програми

5. Дано два числа a і b . Скласти програму, яка обчислює середнє арифметичне: $SA = \frac{a+b}{2}$ та середнє геометричне: $SG = \sqrt{a \cdot b}$ заданих чисел з точністю до тисячних.

Program Begin_5;

```
var
  a,b:real;
  sa,sg:real;
begin
  write('Введіть довільні додатні числа a і b = ');
  readln(a,b);
  sa:=(a+b)/2;
  sg:=SQRT(a*b);
  writeln('sa=',sa:0:3);
  writeln('sg=',sg:0:3);

end.
```

Ім'я програми

Розділ оголошення змінних

Змінні: a, b – дійсного типу (аргументи)Змінна: sa, sg – дійсного типу (результати)**Початок програми**

Вивід текстової підказки на екран

Введення числових значень через "пропуск"

Обчислення середнього арифметичного двох чисел

Обчислення середнього геометричного двох чисел

Виведення на екран отриманого значення SA

Виведення на екран отриманого значення SG

Кінець програми

6. Скласти програму обчислення значення виразу $Y = \frac{2 \cdot (x-3)+1}{x^2+2}$, для довільного цілого числа X .

Program Begin_6;

```
var
  x:integer;
  y:real;
begin
  write('Введіть ціле число = ');
  readln(x);
  y:=(2*(x-3)+1)/(SQR(x)+2);
  writeln('Y(',x,')=',Y:0:2);

end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типуЗмінна: y – дійсного типу**Початок програми**

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення значення виразу

Виведення на екран отриманого результату

Кінець програми

7. Скласти програму обчислення значення виразу $Y = \frac{\sqrt{x^4 + 2} - 3.14 \cdot x}{|x + \ln(10 + x)| + 3} - 2 \cdot x$,

для $x > 0$.

Program Begin_7;

```
var
  x:real;
  y:real;
begin
  write('Введіть число = ');
  readln(x);
  y:=(sqrt(sqr(sqr(x))+2)-3.14*x)/(abs(x+ln(10+x))
+3)-2*x;
  writeln('Y(',x,')=',Y:0:2);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – дійсного типу

Змінна: y – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення значення виразу

Виведення на екран отриманого результату

Кінець програми

8. Дано значення кута в радіанах ($0 < x < 2\pi$). Скласти програму для вираження значення заданого кута в градусах, враховуючи, що $180^\circ = \pi$ радіан. В якості значення π використовувати 3.14.

Program Begin_8;

```
const
  pi=3.14;
var
  x:real;
  y:real;
begin
  write('Введіть значення кута в радіанах = ');
  readln(x);
  y:=x*180/pi;
  writeln('Y(',x,')=',Y:0:2);
end.
```

Ім'я програми

Розділ оголошення постійних величин

Постійній величині PI присвоюються значення 3,14

Розділ оголошення змінних

Змінна: x – дійсного типу

Змінна: y – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення значення кута в градусах

Виведення на екран отриманого результату

Кінець програми

9. Задано масу тіла в кілограмах. Скласти програму, яка виражає масу тіла в тоннах(т) + центнерах(ц) + кілограмах(кг) + грамах(г).

Наприклад: 2450,78кг → 2т+4ц+50кг+780г

Program Begin_9;

```
var
  x:real;
  d_z,t,c,kg:integer;
  gr: real;
begin
  write('Введіть масу тіла в кілограмах = ');
  readln(x);
  gr:=frac(x)*1000;
  d_z:=trunc(x);
  t:=d_z div 1000;
  c:=(d_z-t*1000) div 100;
  kg:=d_z mod 100;
  writeln(x,'кг=' ,t,'т+',c,'ц+',kg,'кг+',gr:0:3,'г');
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – дійсного типу — маса тіла

Змінні: d_z — додаткова змінна – ціла частина від маси тіла, t — к-сть тонн, c — к-сть центнерів, kg — к-сть кілограм, gr — кількість грам.

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення кількості грам

Відкидаємо дробову частину числа

Обчислюємо кількість тонн

Обчислюємо кількість центнерів

Обчислюємо кількість кілограм

Виведення на екран отриманого результату

Кінець програми



10. Задано трьохзначне число. Скласти програму, яка підраховує суму і добуток цифр заданого числа.

Program Begin_10;

```
var
  x:integer;
  a,b,c:integer;
  s,d:integer;
begin
  write('Введіть трьохзначне число = ');
  readln(x);
  a:=x div 100;
  b:=(x div 10) mod 10;
  c:=x mod 10;
  s:=a+b+c;
  d:=a*b*c;
  writeln('S(',x,')=',s);
  writeln('d(',x,')=',d);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типу — задане число

Змінні: a,b,c – цілого типу — відповідні цифри.

Змінні: s,d – цілого типу — сума і добуток

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення першої цифри – кількість сотень

Обчислення другої цифри – кількість десятків

Обчислення третьої цифри – кількість одиниць

Обчислюємо суму цифр

Обчислюємо добуток цифр

Виведення на екран суми

Виведення на екран добутку

Кінець програми

11. З початку доби пройшло N секунд. Скласти програму, яка виводить поточний час в форматі гг:хх:сс, формат годин - 24h.

Program Begin_11;

```
var
  n:integer;
  h,m,s:integer;
begin
  write('Введіть кількість секунд = ');
  readln(n);
  h:=n div 3600;
  s:=n-h*3600;
  m:=s div 60;
  s:=s-m*60;
  writeln(h:2,':',m:2,':',s:2);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: n – цілого типу — кількість секунд

Змінні: h,m,s – цілого типу — години, хвилини, сек.

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення кількості годин

Обчислення кількості секунд(відкидаємо години)

Обчислення кількості хвилин

Обчислення кінцевої кількості секунд

Виведення на екран результату в заданому форматі

Кінець програми

12. Дано тризначне число. Вивести число, отримане при прочитанні початкового числа справа наліво.

Program Begin_12;

```
var
  x:integer;
  a,b,c:integer;
  d:integer;
begin
  write('Введіть трьохзначне число = ');
  readln(x);
  a:=x div 100;
  b:=(x div 10) mod 10;
  c:=x mod 10;
  d:=c*100+b*10+a;
  writeln('d(',x,')=',d);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типу — задане число

Змінні: a,b,c – цілого типу — відповідні цифри.

Змінна: d – цілого типу — шукане число

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення першої цифри – кількість сотень

Обчислення другої цифри – кількість десятків

Обчислення третьої цифри – кількість одиниць

Обчислення заданого числа

Виведення на екран шуканого числа

Кінець програми

13. Дані цілі позитивні числа A , B , C . На прямокутнику розміру $A * B$ розміщена максимально можлива кількість квадратів із стороною C (без накладень, довжини сторін – цілі числа). Знайти кількість квадратів, розміщених на прямокутнику, а також площу незайнятої частини прямокутника.

Program Begin_13;

```
var
  a,b,c:integer;
  k,s:integer;
begin
  write('Введіть довжини сторін прямокутника = ');
  readln(a,b);
  write('Введіть довжину сторони квадрату = ');
  readln(c);
  k:=(a div c)*(b div c);
  s:=a*b-k*c*c;
  writeln('k=',k);
  writeln('s=',s);
end.
```

Ім'я програми

Розділ оголошення змінних
Змінні: a,b,c – цілого типу — відповідні довжини.
Змінні: d,s – цілого типу — к-сть квадратів та площа незайнятої частини

Початок програми

Вивід текстової підказки на екран

Введення конкретних числових значень

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення кількості квадратів

Обчислення площі незайнятої частини

Виведення на екран кількості квадратів

Виведення на екран площі незайнятої частини

Кінець програми

14. Даний номер деякого року (ціле позитивне число). Визначити відповідний йому номер сторіччя, враховуючи, що, наприклад, початком 20 сторіччя було 1900 рік.

Program Begin_14;

```
var
  n:integer;
  h:integer;
begin
  write('Введіть номер року = ');
  readln(n);
  h:=n div 100+1;
  writeln('Заданому рокові відповідає ',h,' ст');
end.
```

Ім'я програми

Розділ оголошення змінних

Змінні: n – цілого типу — номер року.

Змінна: h – цілого типу — століття

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Обчислення століття

Виведення на екран кінцевого результату

Кінець програми

15. Поміняти місцями вміст змінних A і B і вивести нові значення A і B .

Program Begin_15;

```
var
  a,b:real;
begin
  write('Введіть A і B = ');
  readln(a,b);
  a:=a+b;
  b:=a-b;
  a:=a-b;
  writeln(' A=',a:0:2,' B=',b:0:2);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінні: a,b – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення конкретних числових значень

Обмін зачень змінних A і B без використання додаткових змінних

Виведення на екран кінцевого результату

Кінець програми



Практична робота

"Створення і реалізація найпростіших лінійних програм "

Мета роботи:

1. Вивчення порядку дій при обчисленні арифметичних виразів.
2. Розвиток навиків запису виразів мовою програмування і використання стандартних функцій.
3. Розвиток практичних навиків роботи в середовищі програмування.

Завдання:

Скласти програму розв'язку задачі згідно варіанту. В зошити записати текст програми та результат виконання.

Скласти і виконати програму, задавши вхідні дані самостійно.

1. Квіткова клумба має форму круга. Обчислити її периметр і площу за заданим радіусом.
2. Обчислити периметр і площу прямокутного трикутника за даним катетом та гострим кутом.
3. Обчислити довжину кола і площу круга за заданим діаметром.
4. Ділянка лісу має форму рівнобічної трапеції. Обчислити її периметр і площу за заданими сторонами.
5. Ресторан закупає щодня масло m_1 кг по 8.50 грн. за кілограм, сметану m_2 кг по 2.40 грн., вершки m_3 кг по 4.10 грн. Визначити суми, потрібні для купівлі окремих продуктів, і загальну суму.
6. Скільки секунд мають доба, тиждень, рік?
7. Обчислити кінетичну $E = mv^2/2$ та потенціальну $P = mgh$ енергії тіла заданої маси m , яке рухається на висоті h зі швидкістю V .
8. Ціни на два види товарів зросли на p відсотків. Вивести старі та нові ціни.
9. Обчислити площу поверхні $S = 4\pi r^2$ та об'єм $V = 4\pi r^3/3$ сфери за заданим радіусом r .
10. Швидкість світла 299792 км/с. Яку відстань долає світло за годину, добу?
11. Увести врожайність трьох сортів пшениці (36, 40, 44 т/га) і розміри трьох відповідних полів (y га). Скільки зібрали пшениці з кожного поля і з трьох полів разом?
12. Радіус Місяця 1740км. Обчислити площу поверхні $S = 4\pi r^2$ та об'єм планети $F = (4/3)\pi r^3$.
13. Обчислити довжину гіпотенузи та площу прямокутного трикутника за заданими двома катетами.
14. Обчислити об'єм та площу бічної поверхні куба, якщо відоме ребро.
15. Увести продуктивності роботи трьох труб, які наповнюють басейн, і час їхньої роботи. Скільки води набрано в басейні?
16. Яку площу і периметр матиме квадрат, описаний навколо круга заданої площі S .

17. Тіло падає з прискоренням g . Визначити пройдений тілом шлях $h=gt^2/2$ після першої та другої секунд падіння.
18. Обчислити периметр і площу прямокутного трикутника за заданими катетами.
19. Телефонні розмови з трьома населеними пунктами коштують c_1, c_2, c_3 коп/хв. Розмови тривали t_1, t_2, t_3 хв відповідно. Яку суму нарахує комп'ютер до оплати за кожною і всі розмови?
20. Обчислити площу бічної поверхні $S=2\pi rh$ та об'єм $V=\pi r^2 h$ діжки за заданою висотою h та радіусом основи r .
21. Квіткова клумба має форму квадрата. Обчислити її периметр і площу за заданою стороною.
22. Обчислити катет та площу прямокутного трикутника за заданими гіпотенузою та другим катетом.
23. Обчислити сторону та площу $S=d^2/2$ квадрата, якщо відома його діагональ d .
24. Поїзд їхав t_1 год. зі швидкістю v_1 км/год., t_2 год. зі швидкістю v_2 і t_3 год. зі швидкістю v_3 . Визначити пройдені шляхи з різною швидкістю і повний шлях.
25. Обчислити площу бічної поверхні $S=\pi rl$ та об'єм $V=\pi r^2 h/3$ конуса за заданою висотою h , твірною l та радіусом основи r .

Приклад програми:

Варіант 25

Program PR2;

const p=3.14;

var r,l,h,s,v : real;

begin

write('Введіть висоту конуса =');

readln(h);

write('Введіть твірну конуса =');

readln(l);

write('Введіть радіус основи конуса =');

readln(r);

S:=P*R*L;

V:=P*sqr(R)*H/3;

write(' Площа бічної поверхні конуса =', S:6:2);

write(' Об'єм конуса =', V:6:2);

readln;

end.



Вказівка розгалуження

Теоретичний матеріал

Вказівка розгалуження.

IF логічний вираз THEN вказівка1(u) ELSE вказівка2(u);

Читається так: **ЯКЩО** значення логічного виразу істинне **ТО** виконується вказівка1(u) **ІНАКШЕ** виконується вказівка2(u).

Існує короткий запис вказівки розгалуження (неповне розгалуження):

IF логічний вираз THEN вказівка(u);

Відмінність заключається в тому, що при набутті логічним виразом значення **FALSE** виконуються вказівки, які стоять після вказівки розгалуження.

Якщо після слів **THEN** або **ELSE** стоїть декілька вказівок то вони обмежуються операторними дужками - **Begin** і **End**.

Примітка Якщо логічний вираз складається з декількох умов то вони беруться в дужки, які об'єднують за допомогою службових слів **AND** (і), **OR** (чи).

Наприклад:

```
IF (a=b) and (a+c>r) THEN ...
```

Вказівка безумовного переходу

Вказівки в програмі виконуються, як правило в порядку їх запису. Змінити цей порядок можна за допомогою вказівки переходу, яка записується у вигляді:

GOTO мітка;

Вказівка безумовного переходу може бути записана в будь-якому місці програми. Вона дозволяє перейти до мітки передавши управління вказівці, що знаходиться за міткою. Мітки повинні бути описані в розділі **LABEL**.

Наприклад:

```
K1 : c:=c+2;
    if c>10 then Goto K1;
```

Вказівка вибору

Вказівка розгалуження використовується, коли в залежності від умови розв'язок розгалужується на дві гілки. І хоча дану вказівку можна використовувати для розгалуження на більше гілок, вигідніше використати в такому випадку вказівку вибору.

Вказівка вибору записується у вигляді:

CASE вираз OF

мітка1 : вказівка1;

мітка2 : вказівка2;

.....

міткаN : вказівкаN

ELSE вказівка

END;

Вираз який знаходиться між словами **CASE** і **OF** повинен належати до одного з типів даних крім дійсного або рядкового.

Мітки варіантів - константи, тип яких повинен співпадати з типом виразу. Якщо одній вказівці відповідає декілька міток то їх перераховують через кому.

Виконання вказівки вибору проходить наступним чином: спочатку обраховується значення виразу, а тоді виконується вказівка мітка якої співпала з отриманим значенням. Якщо такої мітки нема то виконується вказівка, яка стоїть після слова **ELSE**.

Приклади розв'язку задач:

1. Дано два числа. Вивести на екран більше з них.

<pre> Program IF_1; var a,b:real; begin write('Введіть два числа= '); readln(a,b); IF a>b then writeln('MAX=',a:0:2) else writeln('MAX=',b:0:2); end. </pre>	<p>Ім'я програми Розділ оголошення змінних Змінні: a,b – дійсного типу</p> <p>Початок програми Вивід текстової підказки на екран Введення конкретних числових значень Визначення більшого числа та виведення на екран отриманого результату</p> <p>Кінець програми</p>
--	---

2. Дано три числа. Вивести на екран найменше з них.

<pre> Program IF_2; var a,b,c:real; min: real; begin write('Введіть три числа= '); readln(a,b,c); IF a<b then min:=a else min:=b; IF c< min then min:=c; writeln('MIN=',min:0:2); end. </pre>	<p>Ім'я програми Розділ оголошення змінних Змінні: a,b,c – дійсного типу Змінна: min – дійсного типу – найменше з чисел</p> <p>Початок програми Вивід текстової підказки на екран Введення конкретних числових значень Визначення більшого числа та виведення на екран отриманого результату</p> <p>Кінець програми</p>
--	--

3. Скласти програму, яка перевіряє чи може існувати трикутник з кутами А,В,С.

<pre> Program IF_3; var a,b,c:real; begin write('Введіть значення кутів трикутника = '); readln(a,b,c); IF a+b+c=180 then writeln('Існує') else writeln('не існує'); end. </pre>	<p>Ім'я програми Розділ оголошення змінних Змінні: a,b,c – дійсного типу</p> <p>Початок програми Вивід текстової підказки на екран Введення конкретних числових значень Сума кутів трикутника повинна дорівнювати 180 градусів</p> <p>Кінець програми</p>
---	--

4. Скласти програму, яка перевіряє чи може існувати трикутник з сторонами А,В,С.

<pre> Program IF_4; var a,b,c:real; begin write('Введіть значення сторін трикутника = '); readln(a,b,c); IF (a+b>c) and (b+c>a) and (c+a>b) then writeln('Існує') else writeln('не існує'); end. </pre>	<p>Ім'я програми Розділ оголошення змінних Змінні: a,b,c – дійсного типу</p> <p>Початок програми Вивід текстової підказки на екран Введення конкретних числових значень Сума двох будь-яких сторін трикутника повинна бути більша за третю сторону.</p> <p>Кінець програми</p>
---	---



5. Складіть програму, яка знаходить розв'язки квадратного рівняння виду: $ax^2+bx+c = 0$.

Program IF_5;

```
var
  a,b,c,d,x1,x2:real;
begin
  write('Введіть значення коефіцієнтів квадратного рівняння= ');
  readln(a,b,c);
  D:=b*b-4*a*c;
  IF (d<0) then writeln(' Задане рівняння не має розв'язків');
  IF (d=0) then
  begin
    x1:=(-b+sqrt(d))/(2*a);
    writeln('x=',x1);
  end;
  IF (d>0) then
  begin
    x1:=(-b+sqrt(d))/(2*a);
    writeln('x1=',x1);
    x2:=(-b-sqrt(d))/(2*a);
    writeln('x2=',x2);
  end
end.
```

Ім'я програми

Розділ оголошення змінних
Змінні: a,b,c,d,x1,x2 – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення конкретних числових значень
Обчислення дискримінанту заданого рівняння
Якщо дискримінант заданого рівняння менший за нуль то рівняння не має розв'язків.

Якщо дискримінант рівняння рівний нулю, то
Початок вказівки розгалуження

Обрахунок кореня рівняння

Вивід отриманого значення на екран

Кінець вказівки розгалуження

Якщо дискримінант рівняння більший за нуль, то

Початок вказівки розгалуження

Обрахунок першого кореня рівняння

Вивід отриманого значення на екран

Обрахунок другого кореня рівняння

Вивід отриманого значення на екран

Кінець вказівки розгалуження

Кінець програми

6. Скласти програму знаходження модуля дійсного числа без використання функції ABS.

Program IF_6;

```
var
  a:real;
begin
  write('Введіть число = ');
  readln(a);
  IF a<0 then a:=-a;

  writeln('Модуль(a)=',a);
end.
```

Ім'я програми

Розділ оголошення змінних
Змінна: a – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення конкретного числового значення

Якщо a менше за нуль, то замінюємо його числом з знаком "-".

Вивід отриманого результату на екран

Кінець програми

7. Скласти програму обчислення значення виразу: $Y = \frac{\sqrt{2 \cdot x - 0.1 \cdot x^2}}{2 \cdot x - 1}$. Використати вказівку розгалуження. Врахувати, що при деяких значеннях змінної X вираз обрахувати неможливо.

Program IF_7;

```
var
  x,y:real;
begin
  write('Введіть значення змінної X = ');
  readln(x);
  IF (2*x-1=0) OR (2*x-0.1*sqr(x)<0) then
```

Ім'я програми

Розділ оголошення змінних
Змінні: x,y – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення конкретних числових значень

Якщо знаменник =0 або підкореневий вираз менший

```
writeln('Обрахунок неможливий')
else
begin
y:=sqrt(2*x-0.1*sqrt(x))/(2*x-1);
writeln('Y(',x,')=',y);
end;
end.
```

за нуль то
вивести – "Обрахунок неможливий"
Інакше
Початок
Обрахунок значення функції
Вивід отриманого результату
Кінець вказівки розгалуження
Кінець програми

8. Скласти програму обчислення значення виразу: $Y = \frac{\sqrt{|x-1|+2}}{x+100}$. Використати вказівку розгалуження. Врахувати, що при деяких значеннях змінної X вираз обрахувати неможливо.

```
Program IF_8;
var
x,y:real;
begin
write('Введіть значення змінної X = ');
readln(x);
IF (x+100=0) then writeln('Обрахунок неможливий')
else
begin
y:=sqrt(abs(x-1)+2)/(x+100);
writeln('Y(',x,')=',y);
end;
writeln;
end.
```

Ім'я програми
Розділ оголошення змінних
Змінні: x,y – дійсного типу
Початок програми
Вивід текстової підказки на екран
Введення конкретних числових значень
Якщо знаменник =0 то вивести – "Обрахунок неможливий"
Інакше
Початок
Обрахунок значення функції
Вивід отриманого результату
Кінець вказівки розгалуження
Вивід пустої стрічки
Кінець програми

9. На площині дано дві точки (x1; y1) та (x2; y2). Визначити, яка з них знаходиться на більшій відстані від початку координат.

```
Program IF_9;
var
x1,y1,x2,y2,d1,d2:real;
begin
write('Введіть координати 1-ої точки = ');
readln(x1,y1);
write('Введіть координати 2-ої точки = ');
readln(x2,y2);
d1:=sqrt(sqrt(x1-0)+sqrt(y1-0));
d2:=sqrt(sqrt(x2-0)+sqrt(y2-0));
IF d1<d2 then
writeln('Ближче знаходиться 1-ша точка')
else
IF d2<d1 then writeln('Ближче знаходиться 2-га точка')
else writeln('Точки знаходяться на однаковій відстані');
writeln;
end.
```

Ім'я програми
Розділ оголошення змінних
Змінні: x1,y1,x2,y2,d1,d2 – дійсного типу
Початок програми
Вивід текстової підказки на екран
Введення координат 1-шої точки
Вивід текстової підказки на екран
Введення координат 2-гої точки
Відстань від 1-ої точки до центру координат
Відстань від 2-ої точки до центру координат
Якщо d1<d2 то
вивести – 'Ближче знаходиться 1-ша точка'
Інакше
Якщо d2<d1 то вивести – 'Ближче знаходиться 2-га точка'
Інакше вивести - 'Точки знаходяться на однаковій відстані'
Вивід пустої стрічки
Кінець програми



10. Обчислити значення функції $Y = \begin{cases} x^2, & \text{якщо } x \leq 0; \\ \sqrt{x+1}, & \text{якщо } x > 0 \end{cases}$;

Program IF_10;

```
var
  x,y : real;
begin
  write('Введіть X = ');
  readln(x);
  IF x<=0 then
    y:=sqr(x)
  else
    y:=sqrt(x+1);
  writeln('Y(',x:0:2,')=',y:0:4);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінні: x,y – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення числового значення змінної X

Якщо $x \leq 0$ тоЗмінній Y присвоюємо значення $-x^2$

Інакше

Присвоюємо значення кореня квадратного з $-x+1$

Вивід отриманого результату на екран

Кінець програми

11. Обчислити значення функції $Y = \begin{cases} x, & \text{якщо } x \leq -5 \\ x^2, & \text{якщо } -5 < x < 0; \\ \sqrt{x}, & \text{якщо } x \geq 0 \end{cases}$

Program IF_11;

```
var
  x,y : real;
begin
  write('Введіть X = ');
  readln(x);
  IF x<=-5 then
    y:= x
  else IF x>=0 then y:=sqrt(x)
        else y:= sqr(x);
  writeln('Y(',x:0:2,')=',y:0:4);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінні: x,y – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення числового значення змінної X

Якщо $x \leq -5$ тоЗмінній Y присвоюємо значення $-x$ Інакше якщо $x \geq 0$ то $Y = \text{корінь квадратний з } -x$ Інакше $Y = x*x$

Вивід отриманого результату на екран

Кінець програми

12. Обчислити значення функції $Y = \begin{cases} x, & \text{якщо } x \leq -5 \\ x^2, & \text{якщо } -1 < x < 0; \\ \sqrt{x}, & \text{якщо } x \geq 0 \end{cases}$

Program IF_12;

```
var
  x,y : real;
begin
  write('Введіть X = ');
  readln(x);
  IF (x>-5) and (x<=-1) then writeln('функція не ви-
значена')
  else
  IF x<=-5 then
    y:= x
  else IF x>=0 then y:=sqrt(x)
        else y:= sqr(x);
  writeln('Y(',x:0:2,')=',y:0:4);
```

Ім'я програми

Розділ оголошення змінних

Змінні: x,y – дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення числового значення змінної X

Якщо $(x > -5)$ and $(x \leq -1)$ то вивести - 'функція не визначена'

інакше

Якщо $x \leq -5$ тоЗмінній Y присвоюємо значення $-x$ Інакше якщо $x \geq 0$ то $Y = \text{корінь квадратний з } -x$ Інакше $Y = x*x$

Вивід отриманого результату на екран

end.

Кінець програми

13. Скласти програму, яка у відповідності до введеного номера дня тижня виводить його назву.

Program IF_13;

```
var
  x : integer;
begin
  write('Введіть X = ');
  readln(x);
  Case x of
    1: writeln('Понеділок');
    2: writeln('Вівторок');
    3: writeln('Середа');
    4: writeln('Четвер');
    5: writeln('П'ятниця');
    6: writeln('Субота');
    7: writeln('Неділя');
  else writeln('Такого дня не існує')
end;
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x - цілого типу

Початок програми

Вивід текстової підказки на екран

Введення числового значення змінної X

В залежності від значення X здійснюється вибір

Якщо X=1 то вивести -'Понеділок'

Якщо X=1 то вивести -'Вівторок'

Якщо X=1 то вивести -'Середа'

Якщо X=1 то вивести -'Четвер'

Якщо X=1 то вивести -'П'ятниця'

Якщо X=1 то вивести -'Субота'

Якщо X=1 то вивести -'Неділя'

Інакше вивести -'Такого дня не існує'

Кінець вказівки вибору

Кінець програми

14. Скласти програму, яка у відповідності до введеного знака арифметичної дії виконує відповідну операцію для чисел - c, p.

Program IF_14;

```
var
  x : char;
  c,p,y : real;
begin
  write('Введіть C = ');
  readln(C);
  write('Введіть P = ');
  readln(P);
  write(' Знак арифметичної операції= ');
  readln(X);
  Case x of
    '+': y:=c+p;
    '-': y:=c-p;
    '*': y:=c*p;
    '/': y:=c/p;
  else writeln('Такої операції не існує')
end;
writeln(c,x,p,'=',y);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x - символного типу

Змінні: c,p,y - дійсного типу

Початок програми

Вивід текстової підказки на екран

Введення числового значення змінної C

Вивід текстової підказки на екран

Введення числового значення змінної C

Вивід текстової підказки на екран

Введення числового значення змінної X

В залежності від значення X здійснюється вибір

Якщо X= '+' то y:=c+p

Якщо X= '-' то y:=c-p

Якщо X= '*' то y:=c*p

Якщо X= '/' то y:=c/p

Інакше вивести -'Такої операції не існує'

Кінець вказівки вибору

Вивід результату на екран

Кінець програми

15. Скласти програму, яка у відповідності до введеного номера обраховує: 1 – довжину кола, 2- площу круга радіуса R.

Program IF_15;

```
var
  x : integer;
  r,s,d : real;
```

Ім'я програми

Розділ оголошення змінних

Змінна: x - цілого типу

Змінні: r,s,d - дійсного типу

**begin**

```

write('Введіть R = ');
readln(R);
write('Введіть номер операції= ');
readln(X);
Case x of
  1 : begin d:=6.28*r; writeln('L=',d:0:2) end;
  2 : begin S:=3.14*r*r; writeln('S=',S:0:2) end;
else writeln('Такої операції не існує')
end;
end.

```

Початок програми

Вивід текстової підказки на екран
Введення числового значення змінної R
Вивід текстової підказки на екран
Введення числового значення змінної X
В залежності від значення X здійснюється вибір
Якщо X= 1 то обчислюємо довжину кола
Якщо X= 2 то обчислюємо площу круга
Інакше вивести – 'Такої операції не існує'
Кінець вказівки вибору

Кінець програми

16. Задано два довільних числа. Вивести на екран номер меншого з заданих чисел. Якщо числа рівні – вивести номер другого з них.

Program IF_16;

```

var
x,y:real;
begin
write('Введіть два різних довільних числа X та Y=');
readln(x,y);
IF x<y then
  writeln('Порядковий номер меншого числа=',1)
else
  writeln('Порядковий номер меншого числа=',2);
end.

```

Ім'я програми

Розділ оголошення змінних
Змінні: x,y – дійсного типу

Початок програми

Вивід текстової підказки на екран
Введення числових значень для змінних X та Y
Якщо X<Y то вивести – 'Порядковий номер меншого числа=',1
Інакше вивести – 'Порядковий номер меншого числа=',2

Кінець програми

17. Дано натуральне число x ($x < 99$). Визначити, чи правильне твердження, що x^2 дорівнює кубу суми цифр цього числа.

Program IF_17;

```

var
x,a,b : integer;
begin
write('Введіть двохцифрове число X =');
readln(x);
a:= x div 10;
b:= x mod 10;
IF x*x=(a+b)*(a+b)*(a+b) then
  writeln('ТАК')
else
  writeln('НІ');
end.

```

Ім'я програми

Розділ оголошення змінних
Змінні: x,a,b - цілого типу

Початок програми

Вивід текстової підказки на екран
Введення значення змінної X
Обчислення першої цифри числа
Обчислення другої цифри числа
Якщо x^2 дорівнює кубу суми цифр числа то вивести – 'ТАК'
Інакше вивести – 'НІ'

Кінець програми

18. Вияснити чи утворюють задані три числа неспадну послідовність.

Program IF_18;

```

var
x,y,z : real;
begin
write('Введіть три числа X,Y,Z =');

```

Ім'я програми

Розділ оголошення змінних
Змінні: x,y,z - дійсного типу

Початок програми

Вивід текстової підказки на екран


```

readln(x,y,z);
IF (X<=Y) and (Y<=Z) then
  writeln('ТАК')
else
  writeln('НІ');
end.

```

Введення значення змінних X,Y,Z
Якщо числа утворюють неспадну послідовність то вивести – 'ТАК'
Інакше вивести – 'НІ'
Кінець програми

19. Вияснити чи утворюють задані три числа неспадну послідовність.

```

Program IF_19;
var
  x,y,z : real;
begin
  write(' Введіть координати точки X,Y =');
  readln(x,y);
  write(' Введіть радіус кола =');
  readln(z);
  IF sqrt(sqr(x)+sqr(y))<z then
    writeln('ТАК')
  else
    writeln('НІ');
end.

```

Ім'я програми
Розділ оголошення змінних
Змінні: x,y,z - дійсного типу
Початок програми
Вивід текстової підказки на екран
Введення значення змінних X,Y
Вивід текстової підказки на екран
Введення значення змінної Z
Якщо точка знаходиться в межах заданого кола то вивести – 'ТАК'
Інакше вивести – 'НІ'
Кінець програми

20. Вияснити, що більше для трьох заданих додатних чисел: середнє арифметичне чи середнє геометричне.

```

Program IF_20;
var
  x,y,z,sa,sg : real;
begin
  write('Введіть три числа X,Y,Z =');
  readln(x,y,z);
  sa:=(x+y+z)/3;
  sg:=sqrt(x*y*z);
  IF sa<sg then
    writeln(' середнє геометричне ')
  else
    writeln(' середнє арифметичне ');
end.

```

Ім'я програми
Розділ оголошення змінних
Змінні: x,y,z, sa,sg - дійсного типу
Початок програми
Вивід текстової підказки на екран
Введення значення змінних X,Y,Z
Обчислення середнього арифметичного
Обчислення середнього геометричного
Якщо sa<sg то вивести – середнє геометричне
Інакше вивести – середнє арифметичне
Кінець програми

Практична робота

"Створення і реалізація програм з розгалуженням"

Мета роботи:

1. Отримання навиків використання умовного оператора.
2. Розвиток практичних навиків роботи в середовищі програмування.

Завдання:

Скласти програму розв'язку задачі згідно варіанту. В зошити записати текст програми та результат виконання. Врахувати, що при деяких значеннях X обрахувати значення функції неможливо.



Таблиця функцій:

№	Функція Y_1	Функція Y_2	Функція Y_3	Функція Z
1.	$x+2/x$	$x^5+x^{-4}+x^{1/4}$	$\sin^2(x)+2,15x/ \operatorname{tg}x-2 $	$4Y_1+Y_2+Y_3$
2.	$x-5/x$	$3x^5+x^{-2}+x^{1/2}$	$3\sin x^2+(8-x^{1.4}+x)/(2x-3)$	Y_1/Y_2+Y_3
3.	$x+5,6/x$	$x^{15}+x^{-4}+3,12x^{1/4}$	$8\operatorname{ctg}x-3,18x^{-2}/(x+3\cos x)$	Y_1+Y_2/Y_3
4.	$3/(x+1)$	$x^5+x^{-4}/(8+x^{1/4})$	$4,22x^{1/3}+45\cos x-2/(x^{-3}+2)$	$2Y_1+Y_2-Y_3$
5.	$x-1/2x$	$(x^5+x^{-4}+1)/x^{1/4}$	$\sin x \cos x-23x/\operatorname{tg}x-4,5x$	Y_1+3Y_2/Y_3
6.	$x-7/(x+1)$	$x^{0.6}+x^{-1/6}+x^{1/3}$	$\sin^2(x)+(8-x^{1.4}+x)/(2x-3)$	$Y_1*Y_2+2Y_3$
7.	$x/(2-x)$	$x^5+x^{-4}+x^{7/x}$	$3\sin x^2+2,15x/ \operatorname{tg}x-2 $	Y_1/Y_2*Y_3
8.	$5x/(x+2)$	$x^{5x}+x^{-4}+x^{1/4}$	$8\operatorname{ctg}x+45\cos x-2/(x^{-3}+2)$	$Y_1-Y_2+3Y_3$
9.	$x^2/(x+5)$	$x^5+(x-2)^{-4}+x^{1/4}$	$4,22x^{1/3}+3,18x^{-2}/(x+3\cos x)$	$Y_1/(3Y_2)+Y_3$
10.	$2-x/(2+x)$	$x^5+x^{-4}+x+5^{1/2}$	$\sin x \cos x-45\cos x-2/(x^{-3}+2)$	$Y_1+Y_2+Y_3$
11.	$3,67x/(2+x)$	$(x-3)^{-5}+x^4+x^{14}$	$\sin^2(x)+23x/\operatorname{tg}x-4,5x$	$Y_1+2Y_2*Y_3$
12.	$2/(2x-3)$	$x^5(x+2)^{-4}+x^4$	$3\sin x^2+3,18x^{-2}/(x+3\cos x)$	$Y_1*Y_2+3Y_3$
13.	$x/(2x+3)$	$x^3+x^{-4}+x^{1/2}$	$8\operatorname{ctg}x-2,15x/ \operatorname{tg}x-2 $	$0.5Y_1+Y_2+Y_3$
14.	$4x/(x+1)$	$x^3+(2-x)^{-4}+x^{1/2}$	$4,22x^{1/3}+(8-x^{1.4}+x)/(2x-3)$	$Y_1+2Y_2-Y_3$
15.	$2,85/(4x+1)$	$x^3+x^{-4}+(3+x)^{1/2}$	$\sin x \cos x-3,18x^{-2}/(x+3\cos x)$	$Y_1*Y_2+2Y_3$
16.	$3+x/(x+3)$	$(2-x)^3+x^{-4}+3x^{1/2}$	$\sin^2(x)+45\cos x-2/(x^{-3}+2)$	Y_1-2Y_2/Y_3
17.	$4x/(x-2)/3$	$x^3+8x^{-4}+4x^{1/2}$	$3\sin x^2+23x/\operatorname{tg}x-4,5x$	$Y_1*Y_2*Y_3$
18.	$2x/(3-4x)$	$x^3+(8-x)^{-4}+4x^{1/2}$	$8\operatorname{ctg}x-(8-x^{1.4}+x)/(2x-3)$	$Y_1/Y_2/Y_3$
19.	$(x+2)/(2x-1)$	$x^{-3}+8x^{-4}+(4-x)^{1/2}$	$4,22x^{1/3}+2,15x/ \operatorname{tg}x-2 $	Y_1-Y_2/Y_3
20.	$(x-1)/(x+4)$	$x^3+3x^{-4}+(x-4)^{1/2}$	$\sin x \cos x-(8-x^{1.4}+x)/(2x-3)$	$2(Y_1+Y_2)/Y_3$
21.	$3x+2/(x-8)$	$x^3+8x^{-2}+4x^{1/6}$	$\sin^2(x)+3,18x^{-2}/(x+3\cos x)$	$Y_1/(Y_2+Y_3)$
22.	$2+5/(x+6/x)$	$x^{-4}+2-x^{-2}+4x^{1/4}$	$3\sin x^2+4,5\cos x-2/(x^{-3}+2)$	$(Y_1+Y_2)/2*Y_3$
23.	$2x/(3x+1)/(3x)$	$3x^3+8+x^{-4}+4x^{1/2}$	$8\operatorname{ctg}x+45\cos x-2/(x^{-3}+2)$	$Y_1-Y_2*Y_3$
24.	$8x/(x+5)$	$4x^3+x^{-4}+4-x^{1/2}$	$4,22x^{1/3}+23x/\operatorname{tg}x-4,5x$	Y_1*Y_2/Y_3
25.	$4,3(x+2)/(x+4/x)$	$(x-3)^3+(x+7)^{-4}+(x-4)^{1/6}$	$\sin x \cos^2 x-2,15x/ \operatorname{tg}x-2 $	$Y_1+3Y_2/2Y_3$

Початковий рівень

Обрахувати значення функції Y_3 (згідно варіанту).

Середній рівень

Обрахувати значення функції Y_2 (згідно варіанту).

Достатній рівень

Обрахувати значення функції Y_1 (згідно варіанту).

Високий рівень

Обрахувати значення функції Z (згідно варіанту).

Додаткові задачі (CASE) :

Скласти програму для розв'язування наведеного нижче завдання двома способами, використовуючи: 1) команду case; 2) команду if. Придумати і задати вхідні дані так, щоб вибір був з 4—7 альтернатив.

1. Ввести номер студента зі списку. Вивести його прізвище.
2. Є дані про автомобілі чотирьох моделей. Як вхідне дане ввести номер моделі і отримати характеристики: рік випуску і ціну.
3. Ввести номер поїзда. Вивести назву пункту призначення.
4. Ввести першу букву назви країни. Вивести назву її столиці.
5. Ввести номер дня тижня. Вивести його назву.
6. Ввести номер трамвая. Вивести назви його кінцевих зупинок.
7. Ввести першу букву назви країни. Вивести назву континенту.

8. Ввести номер місяця. Вивести назву пори року.
9. Ввести номер студента у списку. Вивести його ім'я.
10. Ввести першу букву назви міста. Вивести довідку про кількість населення і площу міста.
11. Ввести номер місяця. Вивести назву місяця і номер кварталу.
12. Ввести номер автобуса. Вивести кількість зупинок його маршруту.
13. Ввести першу букву назви країни. Вивести кількість населення і кількість міст цієї країни.
14. Ввести телефонний код райцентру. Вивести його назву.
15. Ввести номер дня тижня. Вивести кількість пар (уроків) у цей день.

Приклад програми:

Варіант 25. Достатній рівень.

```
Program PR3;
```

```
Var
```

```
  x,y: real;
```

```
Begin
```

```
  Readln( x );
```

```
  if (cos(x)=0) or (abs(sin(x)/cos(x)-2)=0) then
```

```
    writeln(' Обрахунок неможливий ')
```

```
  else
```

```
    begin
```

```
      y:=sin(x)*cos(2*x)-2.15*x/abs(sin(x)/cos(x)-2);
```

```
      Writeln( 'y=',y:8:4 );
```

```
    end;
```

```
  readln;
```

```
end.
```



Вказівка повторення

Теоретичний матеріал

Вказівка повторення з передумовою (WHILE).

Вказівка повторення з передумовою записується у вигляді:

WHILE логічний вираз **DO** вказівка;

або

WHILE логічний вираз **DO**

BEGIN

вказівка ;

.....;

вказівка ;

END;

Читається так: **ПОКИ** логічний вираз істиний **ВИКОНУВАТИ** вказівку(и). Виконувану частину часто називають тілом циклу. Працює дана конструкція наступним чином. Спочатку обчислюється значення логічного виразу. Якщо воно рівне **FALSE** (ложь) то тіло циклу не виконується ні разу. Якщо при початковій перевірці значення логічного виразу **TRUE** (істина) то виконується тіло циклу після службового слова **DO**, після чого здійснюється повернення на початок циклу і все повторюється спочатку. В тілі циклу необхідно передбачити зміну даних для логічного виразу, інакше його значення може постійно бути **TRUE**, що призведе до зациклювання (цикл виконується бескінечно довго).

Наприклад: Обчислити суму парних чисел на проміжку від 0 до 100.

Program SUMMA;

var i,s: integer;

Begin

s:=0;

i:=2; {початкові значення }

While i<= 100 do

begin

s:=s+i;

i:=i+2; {зміна даних для лог. виразу}

end;

write('S=',s)

End.

Вказівка повторення з післяумовою (REPEAT).

Вказівка повторення з післяумовою записується у вигляді:

REPEAT

вказівка(*u*);

UNTIL логічний вираз;

Читається так: **ПОВТОРЯТИ** вказівку(и) **ДО** ТИХ ПІР, **ПОКИ** логічний вираз не стане істиним (**TRUE**). Тобто даний цикл завершується, коли логічний вираз приймає значення **TRUE**, а не **FALSE**, на відміну від цикла **WHILE**. Так як умова виходу з циклу перевіряється в кінці, то тіло циклу виконується, як мінімум один раз. Якщо логічний вираз завжди буде приймати значення **FALSE** то відбудеться зациклювання.

Цикл **REPEAT** використовується в ситуаціях, коли виконується деяка група вказівок, а потім, в залежності від результату, або продовжується робота програми далі, або дана група вказівок виконується знову. Класичним прикладом є обчислення виразу з заданою точністю (точність досягнута - закінчуємо обчислення, ні - продовжуємо).

Наприклад: Поділити задане трьохзначне число на 3 з точністю до цілих не використовуючи дії ділення.

```
Program DIL;
  Var   x : real;
        i : integer;
Begin
  readln(x);
  i:=0;
Repeat
  x:=x-3;
  i:=i+1
until x<3;
write (i)
End.
```

Вказівка повторення (FOR - TO - DO).

Вказівка FOR записується у вигляді:

FOR ім'я змінної=вираз1 TO вираз2 DO

```
Begin
  вказівка(u);
End;
```

Читається так: ДЛІЯ змінної від виразу1 ДО виразу2 ВИКОНУВАТИ вказівку(u).

В даному циклі не перевіряється умова закінчення циклу, а просто рахується скільки разів виконується вказівка(u), (зациклювання неможливе) тобто цикл виконується певну кількість разів. Рахунок здійснюється за допомогою спеціального параметра - лічильника. Значення змінної (лічильника) збільшується від виразу1 до виразу2 з кроком 1. Як тільки значення лічильника стає рівним виразу2 виконання циклу припиняється.

Примітка Цикл FOR можна зациклити, якщо тілі циклу переозначити значення змінної(лічильника) так, щоб воно не стало рівне виразу2.

Наприклад: Вивести на екран всі цілі числа та їх квадрати на проміжку від 10 до 20 включно.

```
Program Kvadrat;
  var i,s:integer;
Begin
  For i:=10 to 20 do
  begin
  s:=sqr(i);
  writeln('i=',i:3,' i*i=',s)
  end
End.
```

Вказівка повторення (FOR - DOWNTO - DO).

Вказівка FOR записується у вигляді:

FOR ім'я змінної=вираз1 DOWNTO вираз2 DO

```
Begin
  вказівка(u);
```

```
End;
```

Читається так: ДЛІЯ змінної від виразу1 ДО виразу2 ВИКОНУВАТИ вказівку(u).

Відмінність від попереднього циклу заключається в тому, що значення виразу1 повинно бути більшим значення виразу2. Тобто значення змінної(лічильника) зменшується від виразу1 до виразу2 з кроком -1 (крок можна змінити).



Приклади розв'язку задач:

1. Вивести на екран таблицю значень функції $Y=2x^2$. Змінна X змінюється від 0 до 50 з кроком 1.

```

Program FOR_1;
var
  x : integer;
  y : real;
begin
  for x:=1 to 50 do
  begin
    y:=2*sqr(x);
    writeln('Y(',x,')=',y:0:3);
  end;
end.

```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типу

Змінна: y – дійсного типу

Початок програми

Для змінної X від 1 до 50 з кроком 1 виконувати

Початок циклу(вказівки повторення)

Обчислення значення функції Y для поточного значення змінної X

Вивід отриманого значення функції для конкретного значення X

Кінець циклу

Кінець програми

2. Вивести на екран таблицю значень функції $Y=\sin^2(x)$. Змінна X змінюється від 0 до 10 з кроком 0,5.

```

Program FOR_2;
var
  x : real;
  y : real;
begin
  x:=1;
  while x<=10 do
  begin
    y:=sqr(sin(x));
    writeln('Y(',x,')=',y:0:3);
    x:=x+0.5;
  end;
end.

```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типу

Змінна: y – дійсного типу

Початок програми

встановлення початкового значення змінної X

Поки $x \leq 10$ виконувати

Початок циклу(вказівки повторення)

Обчислення значення функції Y для поточного значення змінної X

Вивід отриманого значення функції для конкретного значення X

Обчислення наступного значення змінної X

Кінець циклу

Кінець програми

3. Вивести на екран таблицю значень функції $Y=2x+1$. Змінна X змінюється від 0 до 10 з кроком 0,5.

```

Program FOR_3;
var
  x : real;
  y : real;
begin
  x:=1;
  repeat
    y:=2*x+1;
    writeln('Y(',x,')=',y:0:3);
    x:=x+0.5;
  until x>10;
end.

```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типу

Змінна: y – дійсного типу

Початок програми

встановлення початкового значення змінної X

Повторювати (початок циклу)

Обчислення значення функції Y для поточного значення змінної X

Вивід отриманого значення функції для конкретного значення X

Обчислення наступного значення змінної X

До тих пір поки значення змінної X не стане біль-

end.

шим за 10 (кінець циклу)
Кінець програми

4. Обчислити суму та кількість цілих, додатних чисел, кратних 3 і менших за 100.

Program FOR_4;

```
var
  x, k : integer;
  s : real;
begin
  s:=0;
  k:=0;
  for x:=1 to 100 do
    if x mod 3=0 then
      begin
        k:=k+1;
        s:=s+x;
      end;
    writeln('K=',k);
    writeln('S=',s:0:1);
  end.
```

Ім'я програми

Розділ оголошення змінних
Змінні: x, k – цілого типу
Змінна: s – дійсного типу

Початок програми

Встановлення початкового значення для кількості
Встановлення початкового значення для суми
Для змінної X від 1 до 100 з кроком 1 виконувати
Якщо залишок від ділення числа на 3 дорівнює 0, то
Початок
Кількість збільшуємо на 1
Суму збільшуємо на значення змінної X
Кінець вказівки розгалуження
Вивід отриманого значення для кількості на екран
Вивід отриманого значення для суми на екран
Кінець програми

5. Обчислити добуток парних, додатних чисел менших 100.

Program FOR_5;

```
var
  x : integer;
  d : real;
begin
  x:=2;
  d:=1;
  while x<=100 do
    begin
      d:=d*x;
      x:=x+2;
    end;
  writeln('D=',d:0:2);
end.
```

Ім'я програми

Розділ оголошення змінних
Змінна: x – цілого типу
Змінна: d – дійсного типу

Початок програми

Встановлення початкового значення змінної X
Встановлення початкового значення добутку
Поки x<=100 виконувати
Початок циклу(вказівки повторення)
Обчислення поточного значення добутку
Обчислення наступного значення змінної X
Кінець циклу
Вивід отриманого значення добутку
Кінець програми

6. Обчислити $\sum_{x=5}^{20} \left(x + \frac{1}{x} \right)$.

Program FOR_6;

```
var
  x : integer;
  s : real;
begin
  s:=0;
  for x:=5 to 20 do
    s:=s+(x+1/x);
  writeln('S=',s:0:1);
end.
```

Ім'я програми

Розділ оголошення змінних
Змінна: x – цілого типу
Змінна: s – дійсного типу

Початок програми

Встановлення початкового значення суми
Для x від 5 до 20 з кроком 1 виконувати
Обчислення поточного значення суми
Вивід отриманого значення суми
Кінець програми



7. Обчислити $n!$ (факторіал – $3!=3*2*1$).

```

Program FOR_7;
var
  n,x : integer;
  f : real;
begin
  f:=1;
  readln(n);
  for x:=n downto 1 do
    f:=f*x;
  writeln('n!=',f:0:0);
end.

```

Ім'я програми

Розділ оголошення змінних

Змінні: n,x – цілого типу

Змінна: f – дійсного типу

Початок програми

Встановлення початкового значення факторіалу

Введення конкретного значення - N

Для X від N до 1 з кроком (-1) виконувати

Обчислення поточного значення факторіалу

Вивід отриманого значення факторіалу

Кінець програми

8. Обчислити $(n-1) + (n-1-2) + \dots + (n-1-2-\dots-(n-1))$, для $n = 10$.

```

Program FOR_8;
var
  n,x : integer;
  f,s : integer;
begin
  n:=10;
  f:=n-1;
  s:=f;
  for x:=2 to n-1 do
    begin
      f:=f-x;
      s:=s+f;
    end;
  writeln('S=',S);
end.

```

Ім'я програми

Розділ оголошення змінних

Змінні: n,x – цілого типу

Змінні: f,s – цілого типу

Початок програми

Встановлення початкового значення

Встановлення початкового значення

Встановлення початкового значення суми

Для X від 2 до N-1 з кроком 1 виконувати

Початок циклу

Обчислення поточного значення доданку

Обчислення поточного значення суми

Кінець циклу

Вивід отриманого значення факторіалу

Кінець програми

9. Знайти точку перетину графіків функцій $Y=x+1$, $F=5-2,5x$ з точністю до 0,5 на проміжку від -10 до 10.

```

Program FOR_9;
var
  x : real;
  y,f : real;
begin
  x:=-10;
  while x<=100 do
    begin
      y:=x+1;
      f:=5-2*x;
      if abs(y-f)<0.5 then
        begin
          writeln('Y(',x:0:1,')=',y:0:2);
          writeln('F(',x:0:1,')=',f:0:2);
        end;
      x:=x+0.1;
    end;
end.

```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типу

Змінні: y,f – дійсного типу

Початок програми

Встановлення початкового значення змінної X

Поки $x \leq 100$ виконувати

Початок циклу(вказівки повторення)

Обчислення поточного значення функції Y

Обчислення поточного значення функції F

Якщо різниця значень F та Y менша за 0,5 то

Початок

Вивід значення функції Y

Вивід значення функції F

Кінець

Обчислення наступного значення змінної X

Кінець циклу

Кінець програми

10. Вивести на екран всі пари трьохзначних чисел результат від ділення яких становить число $\pi = 3,1415$, з точністю до 0,0001.

Program FOR_10;

```
var
a,b: integer;
begin
for a:=100 to 999 do
for b:=100 to 999 do
if abs(a/b-3.1415)<0.0001 then
writeln(a,'/',b,'=',a/b:0:4);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінні: a,b – дійсного типу

Початок програми

Для А від 100 до 999 з кроком 1, виконувати

Для В від 100 до 999 з кроком 1, виконувати

Якщо різниця від ділення a/b по модулю менша за 0,0001 то вивести числа А та В та результат їх ділення

Кінець програми**Практична робота**

"Створення і реалізація програм з повторенням"

Мета роботи:

1. Отримання навиків використання вказівок повторення.
2. Розвиток практичних навиків роботи в середовищі програмування.

Завдання:

Скласти програму розв'язку задачі згідно варіанту. В зошити записати текст програми та результат виконання.

Задача 1

Таблиця мір. Побудувати таблицю відповідностей між мірами. Початкове значення міри, крок зміни цього значення та кількість рядків у таблиці (10-15) задати самостійно у режимі діалогу. Оформити таблицю якнайкраще, використовуючи формати виведення.

1. 1 унція = 28.353495 г = 142 карати;
2. 1 драхм = 1.77185 г = 0.06249 унцій;
3. 1 карат = 0.2 г = 2.9412 гран;
4. 1 гран = 0.068 г = 0.038378 драхм;
5. 1 пайп = 54.18 пек = 477.33 л;
6. 1 галон (брит.) = 1.2 галон (СІЛА) = 4.546 л;
7. 1 галон (США) = 0.0347 сак = 3.785 л;
8. 1 чарка = 0.0568 л = 0.00012 пайпа;
9. 1 квартет = 291 л = 5123.24 чарок;
- 10.1 страйк = 72.73 л = 1280.46 чарок;
- 11.1 челдрон = 1.309 л = 0.149 пека;
- 12.1 сак = 109 л = 1.499 страйка;
- 13.1 пек = 8.81 л = 0.07929 сака;
- 14.1 корд малий = 3.624 куб. м = 128 куб. футів;
- 15.1 стандарт = 4.672 куб. м = 0.165 рода;
- 16.1 род = 28.3 куб. м = 1000 куб. футів;
- 17.1 чейн будівельний = 30.48 м = 100 футів;
- 18.1 фінгер = 11.4 см = 4.5 дюймів;
- 19.1 нейл = 5.7 см = 2.25 дюймів;
- 20.1 фут = 0.3048 м = 12 дюймів;
- 21.1 ярд = 0.9144 м = 3 фути;
- 22.1 кабельт Брит. = 0.183 км = 680 футів;
- 23.1 кабельт США = 219.5 м = 720 футів; 24.1 дюйм = 2.54 см = 12 ліній;
- 25.1 морська миля = 1.852 км = 6076 футів.

**Приклад програми:**

Варіант 25.

{1 морська миля = 1.852 км = 6076 футів.}

Program Z1_V25;

Const

k1 = 1.852;

k2 = 6076;

Var

f,m, km:real;

Begin

m:=0;

WriteLn(' Милі ',' Кілометри ',' Фути ');

While m<=10 do

Begin

km:=m*k1;

f:=m*6076;

WriteLn(m:5:1,km:10:3,f:12:1);

m:=m+0.5;

end

end.

Задача 2

Обчислення скінчених сум і добутоків. Обчислити значення виразу z для свого варіанта:

- | | | |
|-----------------------------------|--------------------------------------|------------------------------------|
| 1) $z = a+b$; | 10) $z = ab-\pi$; | 19) $z = 12a-\cos(b) $; |
| 2) $z = ab$; | 11) $z = a-2b$; | 20) $z = 2a-b$; |
| 3) $z = \operatorname{tg}(b)-a$; | 12) $z = \operatorname{atg}b$; | 21) $z = \operatorname{tg}(a+b)$; |
| 4) $z = (a+b)^2$; | 13) $z = \cos(ab)$; | 22) $z = \ln a+4b $; |
| 5) $z = 5ab-4$; | 14) $z = a-6 $; | 23) $z = 3ab-\cos(b)$; |
| 6) $z = \sin(a)+b$; | 15) $z = \operatorname{ctg}(2a)-b$; | 24) $z = 4a+e^b$; |
| 7) $z = b\operatorname{tg}a$; | 16) $z = e^{3ab}$; | 25) $z = 5a-2b$. |
| 8) $z = a^2+3b$; | 17) $z = 4ba-b$; | |
| 9) $z = (ab)^{1/4}$; | 18) $z = 2a-b$; | |

якщо

$$a = \sum_{x=i}^{i+8} Y_2(x) \quad , \quad b = \prod_{x=i}^{i+8} Y_3(x)$$

де i - номер варіанта, x - ціле число. Вирази для функцій Y_2 та Y_3 визначити з таблиці (практична робота – розгалуження). Вивести значення x , a , b , z у вигляді таблиці.

Приклад програми:

Варіант 25.

Program Z1_V25;

Var

x : integer;

a,b,z: real;

Begin

a:=0;

b:=1;

writeln(' x ', a ', b ', z ');

```

For x:=25 to 32 do
Begin
a:=a+exp(3*ln(x-3))+exp(-4*ln(x+7))+exp((1/6)*ln(x-4));
b:=b*(4.3*(x+2)/(x-4/x));
z:=5*a-2*b;
writeln(x:4,a:13:2,b:13:2,z:13:2);
end
end.

```

Задача 3. Обчислення нескінченних сум. Утворити нескінченно спадну числову послідовність:

$a_k = (-1)^k Y_3(k) / (k!)$, де i - номер варіанта, x – довільне число з проміжку (0;1), $k=1,2,\dots$

Обчислити суму цієї послідовності з точністю $\epsilon=0.00000001$. Скільки потрібно доданків для досягнення заданої точності? Виконайте програму тричі для різних значень точності.

Приклад програми:

Варіант 25.

```

Program Z3_V25;
Var
n,k : integer;
a,s,e,x,fact,y:real;
Begin
n:=0;
s:=0;
k:=1;
e:=0.00000001;
x:=0.3;
a:=1;
fact:=1;
while abs(a)>e do
begin
y:=4.3*(k+2)/(k+4/k);
a:=exp(-1*ln(k))*y/fact;
s:=s+a;
k:=k+1;
fact:=fact*k;
n:=n+1;
end;
WriteLn('Кількість доданків = ',n);
WriteLn('Сума ряду =',s:12:8);
end.

```

Задача 4. Табулювання функції і пошук даних. Протабулювати функцію $z = Y_1(x)$ на проміжку $[0; i]$ з кроком $h=0.1i$, де i - номер варіанта. Результати обчислень вивести на екран у вигляді таблиці пар чисел x, z . Виконати завдання пошуку даних. Якщо шуканих даних немає, вивести про це повідомлення.

1. Обчислити суми першого та останнього значень функції.
2. Обчислити суму та добуток всіх значень функції y , для яких виконуються нерівності $z < -3,2$ або $z > 0$.



3. Обчислити добуток та кількість усіх значень функції y , для яких виконуються нерівності $z < -3$ або $z > 0,4$.
4. Обчислити добуток усіх від'ємних значень функції y та визначити кількість додатних.
5. Обчислити добуток значень аргументу (x), для яких досягаються мінімальне та максимальне значення функції z .
6. Скільки було від'ємних значень? Визначити максимальне значення.
7. Визначити суму додатних значень функції та кількість від'ємних.
8. Скільки від'ємних та додатних значень має функція z ?
9. Обчислити суму та кількість додатних значень функції z .
10. Обчислити суму квадратів усіх додатних значень функцій z . Визначити, для якого x функція набуває мінімального значення.
11. Обчислити модуль різниці максимального та першого значень z .
12. Обчислити суму усіх значень функції y , для яких виконуються нерівності $z < 1,2$ або $z > 4$. Визначити максимальне значення функції.
13. Обчислити добуток додатних значень та кількість від'ємних.
14. Обчислити добуток від'ємних значень функції z . У якій точці (x) функція набуває максимального значення.
15. Обчислити добуток усіх значень функції z , для яких справджується нерівність $1 < z < 3,1$. Визначити, для якого x функція набуває мінімального значення.
16. Обчислити кількість та добуток усіх від'ємних значень z .
17. Обчислити суму квадратів та добуток усіх значень функції y , для яких справджується нерівність $-2,41 < z < 5$.
18. Обчислити модуль добутку максимального та мінімального значень.
19. Обчислити середнє арифметичне всіх від'ємних значень функції.
20. Обчислити суму кубів всіх додатних значень та їхню кількість.
21. Знайти середнє арифметичне тих значень функції y , для яких виконуються нерівності $z < 0$ або $z > 1$.
22. Знайти мінімальне значення функції, а також визначити значення аргумента, для якого воно досягається.
23. Обчислити суму та кількість тих значень функції y , для яких виконується нерівність $0 < z < 1$.
24. Обчислити кількість та добуток тих значень функції y , для яких виконуються нерівності $1,3 < z < 5$.
25. Яких значень функції більше: додатних чи від'ємних?

Масиви.**Теоретичний матеріал**

Масив - структурний тип даних, який складається з фіксованого числа елементів одного і того ж типу.

Розрізняють одно- і двовимірні масиви.

Одновимірні масиви

Одновимірний масив - це лінійна сукупність елементів одного типу. Кількість елементів задається під час опису масива і не змінюється на протязі виконання програми.

Методи опису масивів.

1) *var ім'я масиву:array[m..n] of тип елементів масиву;*

де m,n - цілі числа.

Наприклад

A:array[1..9] of integer;

C:array[4..12] of real;

2) *type ім'я типу=array[m..n] of тип компонент масиву*

var ім'я масиву:ім'я типу;

Наприклад:

type TAB=array[1..9] of integer;

var A:TAB;

При зверненні до елементів масиву необхідно вказувати ім'я масиву і номер елемента (індекс).

Наприклад:

A[4], A[8], C[11]

Введення і виведення елементів масиву здійснюється поелементно, тому для введення і виведення елементів масиву використовують оператори циклу.

Наприклад: *Знайти добуток елементів масиву*

Program DOBUTOK;

Var A:array[1..9] of real;

d: real;

i: integer;

Begin

For i:=1 to 9 do

begin

writeln('Введіть елементи масиву');

write('A['i,']=');

readln(A[i]);

end;

{Обчислення добутку}

d:=1;

For i:=1 to 9 do d:=d*a[i];

{Вивід добутку на екран}

writeln('d=',d)

End.

Двовимірні масиви

Двовимірні масиви являють собою таблицю, яка складається з m-рядків та n-стовпчиків. Двовимірний масив можна замінити сукупністю двох одновимірних масивів.

Описують двовимірний масив майже так само, як і одновимірний.

var ім'я масиву:array[m..k,n..h] of тип елементів масиву;



Наприклад:

```
var A[1..4,1..9] of real;
```

При зверненні до елементів такого масиву необхідно вказувати два індекси - номер рядка і номер стовпчика. Опрацювання елементів масиву здійснюється за допомогою двох циклів.

Наприклад: *Знайти добуток елементів двовимірного масиву*

```
Program DOBUTOK;
```

```
Var A:array[1..9,1..6] of real;
```

```
    d: real;
```

```
    i,j: integer;
```

```
Begin
```

```
  For i:=1 to 9 do
```

```
    For j:=1 to 6 do
```

```
      begin
```

```
        write('A[',i,',',j,']=');
```

```
        readln(A[i,j]);
```

```
      end;
```

```
    d:=1;
```

```
  For i:=1 to 9 do
```

```
    For j:=1 to 6 do
```

```
      d:=d*a[i,j];
```

```
    writeln('d=',d)
```

```
End.
```

Впорядкування елементів масиву.

Іноколи виникає необхідність впорядкувати елементи масиву в порядку зростання чи спадання. Для розв'язування даних задач використовують декілька методів:

1. Метод бульбашок

Метод "бульбашок" оснований на порівнянні двох сусідніх елементів, якщо числа в парі розміщені в порядку зростання (спадання взаємності від умови задачі) то залишаємо їх без змін, якщо ні, то міняємо їх місцями. Недоліком даного методу являється надмірна кількість переглянутих пар елементів. Щоб уникнути цього використовують допоміжну змінну в якій фіксується кількість перестановок, якщо ця кількість рівна нулю перегляд закінчується - масив впорядковано.

```
Program Sort1;
```

```
var  a:array[1..15] of real;
```

```
    j,i,n:integer;
```

```
    m:real;
```

```
begin
```

```
  read(n);
```

```
  for i:=1 to n do
```

```
    readln(a[i]);
```

```
  for j:=1 to n-1 do
```

```
    begin
```

```
      for i:=1 to n-j do
```

```
        if a[i]>a[i+1] then
```

```
          begin
```

```
            m:=a[i];
```

```
            a[i]:=a[i+1];
```

```
            a[i+1]:=m
```

```
          end;
```

```
end;
for i:=1 to n do writeln (a[i]:1);
end;
```

Впорядкування елементів масиву методом вибору мінімального елемента.

Сортування вибором проводять в наступному порядку:

1. Визначити номер мінімального елемента масиву.
2. Поміняти місцями мінімальний і перший елементи масиву.
3. Повторити виконання пунктів 1 і 2 для залишку масиву (без першого елемента).
4. Виконувати пункти 1 і 2 поки залишок масиву не скоротиться до одного елемента.

Розглянемо порядок сортування для масиву $A[5]$, який містить елементи - 4,5,1,2,3.

```

4 5 1 2 3
1 5 4 2 3
1 2 4 5 3
1 2 3 5 4
1 2 3 4 5
```

Підкреслено область пошуку найменшого елемента.

Program Sort2;

```
var F:array[1..5] of integer;
    i,j,min,k:integer;
```

```
begin
```

```
{Блок введення елементів масиву}
```

```
writeln('Введіть елементи масиву');
```

```
for i:=1 to 5 do
```

```
begin
```

```
    write('F[' ,i ,']=');
```

```
    readln(f[i]);
```

```
end;
```

```
{Блок сортування елементів масиву}
```

```
for i:=1 to 5 do
```

```
begin
```

```
    min:=f[i];
```

```
    k:=i;
```

```
    j:=i+1;
```

```
    while j<=5 do
```

```
    begin
```

```
        if min > f[j] then
```

```
        begin
```

```
            min:=f[j];
```

```
            k:=j;
```

```
        end;
```

```
        j:=j+1;
```

```
    end;
```

```
    f[k]:=f[i];
```

```
    f[i]:=min;
```

```
end;
```

```
{виведення відсортованого масиву}
```

```
Writeln('Масив після сортування');
```

```
for i:=1 to 5 do
```

```
write(f[i]:4);
```

```
End.
```



Приклади розв'язку задач:

1. Обчисліть суму елементів масиву А [1..10]. Елементи масиву довільні числа.

```

Program ARRAY_1;
var
  i : integer;
  A : array[1..10] of real;
  s: real;
begin
  for i:=1 to 10 do
  begin
    write ('A[',i,']=');
    readln(a[i]);
  end;
  s:=0;
  for i:=1 to 10 do
    s:=s+a[i];
    writeln('S=',s:0:3);
end.

```

Ім'я програми

Розділ оголошення змінних

Змінна: i – цілого типу

Змінна: А – масив з 10 елементів дійсного типу

Змінна: s – дійсного типу

Початок програми

Для змінної I від 1 до 10 з кроком 1 виконувати

Початок циклу(вказівки повторення)

Вивід текстової підказки

Введення поточного елемента масиву

Кінець циклу

Встановлення початкового значення суми

Для змінної I від 1 до 10 з кроком 1 виконувати

Обчислення суми елементів масиву

Вивід кінцевого значення суми

Кінець програми

2. Обчисліть добуток додатних елементів масиву А [1..10]. Елементи масиву довільні числа.

```

Program ARRAY_2;
var
  i : integer;
  A : array[1..10] of real;
  d: real;
begin
  for i:=1 to 10 do
  begin
    write ('A[',i,']=');
    readln(a[i]);
  end;
  d:=1;
  for i:=1 to 10 do
    if a[i]>0 then d:=d*a[i];
    writeln('D=',d:0:3);
end.

```

Ім'я програми

Розділ оголошення змінних

Змінна: i – цілого типу

Змінна: А – масив з 10 елементів дійсного типу

Змінна: d – дійсного типу

Початок програми

Для змінної I від 1 до 10 з кроком 1 виконувати

Початок циклу(вказівки повторення)

Вивід текстової підказки

Введення поточного елемента масиву

Кінець циклу

Встановлення початкового значення добутку

Для змінної I від 1 до 10 з кроком 1 виконувати

Обчислення добутку додатних елементів масиву

Вивід кінцевого значення добутку

Кінець програми

3. Обчисліть добуток та кількість додатних, парних елементів масиву С [1..8]. Елементи масиву довільні цілі числа.

```

Program ARRAY_3;
var
  i,k : integer;
  C : array[1..8] of integer;
  d: real;
begin
  for i:=1 to 8 do
  begin
    write ('C[',i,']=');

```

Ім'я програми

Розділ оголошення змінних

Змінні: i,k – цілого типу

Змінна: С – масив з 8 елементів цілого типу

Змінна: d – дійсного типу

Початок програми

Для змінної I від 1 до 8 з кроком 1 виконувати

Початок циклу(вказівки повторення)

Вивід текстової підказки


```

readln(c[i]);
end;
d:=1;
k:=0;
for i:=1 to 8 do
  if (c[i]>0) and (c[i] mod 2=0)
  then
  begin
    d:=d*c[i];
    k:=k+1;
  end;
  writeln('D=',d:0:0);
  writeln('K=',k);
writeln;
end.

```

Введення поточного елемента масиву
Кінець циклу
Встановлення початкового значення добутку
Встановлення початкового значення кількості
Для змінної I від 1 до 8 з кроком 1 виконувати
Якщо елемент масиву >0 і парний,
То
Початок
Обчислення добутку
Обчислення кількості
Кінець вказівки розгалуження
Вивід кінцевого значення добутку
Вивід кінцевого значення кількості
Вивід пустої стрічки
Кінець програми

4. Обчисліть кількість додатних елементів масиву C [1..8], що стоять на парних місцях. Елементи масиву довільні цілі числа.

```

Program ARRAY_4;
var
  i,k : integer;
  C : array[1..8] of integer;
begin
  for i:=1 to 8 do
  begin
    write ('C[',i,']=');
    readln(c[i]);
  end;
  k:=0;
  for i:=1 to 8 do
    if (c[i]>0) and (i mod 2=0)
    then
      k:=k+1;
    writeln('K=',k);
  writeln;
end.

```

Ім'я програми
Розділ оголошення змінних
Змінні: i,k – цілого типу
Змінна: C – масив з 8 елементів цілого типу
Початок програми
Для змінної I від 1 до 8 з кроком 1 виконувати
Початок циклу(вказівки повторення)
Вивід текстової підказки
Введення поточного елемента масиву
Кінець циклу
Встановлення початкового значення кількості
Для змінної I від 1 до 8 з кроком 1 виконувати
Якщо елемент масиву >0 і має парний, номер
То
Кількість збільшуємо на 1
Вивід кінцевого значення кількості
Вивід пустої стрічки
Кінець програми

5. Заповнити масив C[1..20] значеннями функції $Y=\sin(2x)$. X змінюється від 1 до 20 з кроком 1.

```

Program ARRAY_5;
var
  x : integer;
  C : array[1..20] of real;
begin
  for x:=1 to 20 do
    c[x]:=sin(2*x);
  for x:=1 to 20 do
    writeln('Sin(2*,x,)=',c[x]:0:4);
  writeln;
end.

```

Ім'я програми
Розділ оголошення змінних
Змінна: x – цілого типу
Змінна: C – масив з 20 елементів дійсного типу
Початок програми
Для змінної X від 1 до 20 з кроком 1 виконувати
Заповнення масиву значеннями функції
Для змінної X від 1 до 20 з кроком 1 виконувати
Вивід значень функції
Вивід пустої стрічки
Кінець програми



6. Заповнити масив $C[1..20]$ довільними числами за формулою $C(x) = 1/x * \text{random}(100) - x * \text{random}(50)$. Вивести на екран максимальний елемент заданого масиву

Program ARRAY_6;

```
var
  x : integer;
  C : array[1..20] of real;
  max: real;
begin
  for x:=1 to 20 do
    c[x]:= 1/x*random(100)-x*random(50);
  max:=c[1];

  for x:=2 to 20 do
    if max<c[x] then max:=c[x];

  writeln('Max=',max:0:4);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типу

Змінна: C – масив з 20 елементів дійсного типу

Змінна: max – дійсного типу

Початок програми

Для змінної X від 1 до 20 з кроком 1 виконувати

Заповнення масиву значеннями функції

задаємо початкове значення, позначивши за максимальний – 1-ий елемент масиву

Для змінної X від 1 до 20 з кроком 1 виконувати

Якщо максимальний е-т менший за поточний е-т масиву то його значення присвоюємо максимальному

Вивід максимального елемента масиву

Кінець програми

7. Заповнити масив $C[1..20]$ довільними числами за формулою $C(x) = 1/x * \text{random}(100) - x * \text{random}(50)$. Вивести на екран мінімальний елемент заданого масиву

Program ARRAY_7;

```
var
  x : integer;
  C : array[1..20] of real;
  m: real;
begin
  for x:=1 to 20 do
    c[x]:= 1/x*random(100)-x*random(50);
  m:=c[1];

  for x:=2 to 20 do
    if m>c[x] then m:=c[x];

  writeln('MIN=',m:0:4);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – цілого типу

Змінна: C – масив з 20 елементів дійсного типу

Змінна: m – дійсного типу

Початок програми

Для змінної X від 1 до 20 з кроком 1 виконувати

Заповнення масиву значеннями функції

задаємо початкове значення, позначивши за мінімальний – 1-ий елемент масиву

Для змінної X від 2 до 20 з кроком 1 виконувати

Якщо поточний максимальний елемент менший за поточний елемент масиву то його значення присвоюємо максимальному

Вивід максимального елемента масиву

Кінець програми

8. Заповнити масив $C[1..20]$ довільними числами за формулою $C(x) = 1/x * \text{random}(100) - x * \text{random}(50)$. Вивести на екран максимальний елемент заданого масиву та його номер

Program ARRAY_8;

```
var
  x,n : integer;
  C : array[1..20] of real;
  max: real;
begin
  for x:=1 to 20 do
    c[x]:= 1/x*random(100)-x*random(50);
  max:=c[1];
```

Ім'я програми

Розділ оголошення змінних

Змінні: x, n – цілого типу

Змінна: C – масив з 20 елементів дійсного типу

Змінна: max – дійсного типу

Початок програми

Для змінної X від 1 до 20 з кроком 1 виконувати

Заповнення масиву значеннями функції

Задаємо початкове значення, позначивши за макси-

```

n:=1;
for x:=2 to 20 do
  if max<c[x] then
    begin
      n:=x;
      max:=c[x];
    end;
writeln('N_max=',n);
writeln('Max=',max:0:4);
end.

```

мальний – 1-ий елемент масиву
Для змінної X від 2 до 20 з кроком 1 виконувати
Якщо поточний максимальний елемент менший за поточний елемент масиву то його значення присвоюємо максимальному, фіксуємо номер максимального елемента – змінна - n

Вивід номеру максимального елемента
Вивід максимального елемента масиву
Кінець програми

9. Заповнити масив C[1..20] довільними числами за формулою $C(x) = 1/x * \text{random}(100) - x * \text{random}(50)$. Відсортувати створений масив в порядку спадання. Вивести на екран відсортований масив.

```

Program ARRAY_9;
var
  x,i : integer;
  C : array[1..20] of real;
  m: real;
begin
  for x:=1 to 20 do
    c[x]:= 1/x*random(100)-x*random(50);
  for i:=1 to 20 do
    for x:=2 to 20 do
      if c[x-1]<c[x] then
        begin
          m:=c[x];
          c[x]:=c[x-1];
          c[x-1]:=m;
        end;
    for x:=1 to 20 do
      writeln('C[' ,x, ']=',c[x]:0:4);
end.

```

Ім'я програми

Розділ оголошення змінних

Змінні: x, i – цілого типу

Змінна: C – масив з 20 елементів дійсного типу

Змінна: m – дійсного типу

Початок програми

Для змінної X від 1 до 20 з кроком 1 виконувати

Заповнення масиву значеннями функції

Для змінної I від 1 до 20 з кроком 1 виконувати (зовнішній цикл)

Для змінної X від 1 до 20 з кроком 1 виконувати (внутрішній цикл)

Якщо попередній елемент менший за поточний елемент масиву то проводимо обмін значень елементів масиву з використанням додаткової змінної – m.

Змінна – m використовується для збереження проміжних значень

Вивід відсортованого масиву

Вивід відсортованого масиву

Кінець програми

10. Заповнити масив C[1..20] довільними числами за формулою $C(x) = x * \text{random}(100) - x * \text{random}(50)$. Підрахувати кількість елементів, що падають в проміжок від -10 до 10.

```

Program ARRAY_10;
var
  x,i : integer;
  C : array[1..20] of real;
begin
  for x:=1 to 20 do
    c[x]:= x*random(100)-x*random(50);
  for x:=1 to 20 do
    if (c[x]<=-10) and (c[x]<=10) then i:=i+1;
  writeln('Кількість=',i);
end.

```

Ім'я програми

Розділ оголошення змінних

Змінні: x, i – цілого типу

Змінна: C – масив з 20 елементів дійсного типу

Початок програми

Для змінної X від 1 до 20 з кроком 1 виконувати

Заповнення масиву значеннями функції

Для змінної X від 1 до 20 з кроком 1 виконувати

Якщо елемент задовільняє умові то кількість збільшуємо на 1

Вивід результатів обчислення кількості

Кінець програми



11. Задано масив ABC, що містить 15 елементів дійсного типу. Заповнити масив випадковими числами від 0 до 100. Знайти найбільший та найменший елементи масиву. Вивести на екран знайдені елементи та їх номери.

```

program ARRAY_11;
var
    max,min: real;
    n_max,n_min,n:integer;
    abc: array[1..15] of real;
begin
    {блок введення даних}
    for n:=1 to 15 do
        begin
            abc[n]:=random(1000)/(random(100)+1);
            writeln('ABC['n,']='abc[n]:0:3);
        end;
    {Блок обчислень}
    max:=abc[1];
    n_max:=1;
    min:=abc[1];
    n_min:=1;
    for n:=2 to 15 do
        begin
            if abc[n]>max then
                begin
                    max:=abc[n];
                    n_max:=n;
                end;
            if abc[n]<min then
                begin
                    min:=abc[n];
                    n_min:=n;
                end;
        end;
    writeln('Max='max:0:2);
    writeln('Номер максимального елемента='n_max);
    writeln('Min='min:0:2);
    writeln('Номер мінімального елемента='n_min);
end.
  
```

Ім'я програми

Розділ оголошення змінних

Змінні: max,min – дійсного типу

Змінні: n_max,n_min,n – цілого типу

Змінна: abc – масив з 15 елементів дійсного типу

Початок програми

Для змінної N від 1 до 15 з кроком 1 виконувати

Початок циклу

Заповнення масиву довільними значеннями

Вивід елементів масиву на екран

Кінець циклу

В якості максимального вибираємо перший елемент

масиву, номер максимального елемента = 1

В якості мінімального вибираємо перший елемент

масиву, номер мінімального елемента = 1

Для змінної N від 1 до 15 з кроком 1 виконувати

Початок циклу

Якщо поточний елемент більший за максимальний, то початок

Максимальному присвоюється значення поточного елемента, номер максимального=номеру поточного

Кінець вказівки розгалуження

Якщо поточний елемент менший за мінімальний, то початок

Мінімальному присвоюється значення поточного елемента, номер мінімального=номеру поточного

Кінець вказівки розгалуження

Кінець циклу(вказівки повторення)

Вивід максимального елемента

Вивід номера максимального елемента

Вивід мінімального елемента

Вивід номера мінімального елемента

Кінець програми

12. Протабулювати функцію $y=x*x$ від 0 до 10 з кроком 0,5. Отримані значення функції занести в двовимірний масив UX[1..2,1..21]..

```

Program ARRAY_12;
var
    ux: array [1..2,1..21] of real;
    n: integer;
    y,x:real;
begin
    x:=0;
    n:=1;
  
```

Ім'я програми

Розділ оголошення змінних

Змінна: ux – масив з 20 елементів дійсного типу

Змінна: n – цілого типу

Змінні: y,x – дійсного типу

Початок програми

Встановлення початкового значення змінної X

Встановлення початкового значення номера елемента

```

while x<=10 do
begin
y:=x*x;
yx[1,n]:=x;
yx[2,n]:=y;
n:=n+1;
x:=x+0.5;
end;
for n:=1 to 21 do
writeln ('x=',yx[1,n]:0:1,'      ','y=',yx[2,n]:0:3);
end.

```

Поки $x \leq 10$ виконувати
Початок циклу
Обчислення значення функції
Заповнення першого рядка масиву – X
Заповнення другого рядка масиву – Y
Наступний номер елемента в рядку
Наступне значення змінної – X
Кінець циклу
Для n від 1 до 21 виконувати
Вивід елементів масиву
Кінець програми

Практична робота

"Створення і реалізація програм з використанням масивів"

Мета роботи:

1. Отримання навиків використання масивів.
2. Розвиток практичних навиків роботи в середовищі програмування.

Завдання:

Скласти програму розв'язку задачі згідно варіанту. В зошити записати текст програми та результат виконання

Задача 1. Одновимірні масиви. Нехай прибуток фірми за N-ий рік обчислюється за формулою $Y = k * \text{random}(100) - i * \text{random}(50)$ умовних одиниць, де $N = 1991, 1992, \dots, 2001$; i — номер варіанта, $k = 1, 2, \dots, 10$. Якщо $y > 0$, то вважатимемо, що фірма у відповідний рік мала прибуток, а у випадку $y < 0$ - збитки. Вивести на екран таблицю: номер року, величина прибутку, примітка(прибуток, збитки).

Пошук даних. Розглянути фінансову діяльність фірми протягом десяти років. Виконати додатково завдання свого варіанта, наведене нижче. Вивести повідомлення, якщо шуканих даних немає, наприклад, якщо збитків чи прибутків не було взагалі.

1. Обчислити суму прибутків фірми. Визначити максимальний збиток (якщо збитки були).
2. Обчислити суму збитків. У якому році збиток був максимальний?
3. Обчислити суми прибутків та збитків фірми та їх різницю. Коли прибуток був максимальний?
4. Скільки років поспіль прибутків було менше, ніж 1000, але більше, ніж 500 у.о? Коли фірма зазнала найбільших збитків?
5. Обчислити суму збитків. У якому році прибуток був найбільший?
6. Обчислити суму прибутків у межах $0 < Y < 710$ (в у.о.). У якому році фірма зазнала найбільших збитків?
7. Скільки років прибутки були в межах від 200 до 700 у.о? Які це були роки?
8. Обчислити суму всіх збитків. У якому році збиток був найбільший? Який це був збиток?
9. Обчислити суму тих збитків, для яких справджуються умови $Y < -650$ або $Y > -150$ (в у.о.). Визначити найбільший прибуток.
10. Визначити суми прибутків та збитків. Скільки років фірма була прибутковою?
11. Обчислити суму прибутків, що були у межах $230 < Y < 8500$ (в у.о.). Скільки років фірма мала такі прибутки?
12. Обчислити суму збитків, які були в межах $-750 < Y < -200$ (в у.о.). Коли дохід був мінімальний?



13. Обчислити суму прибутків та збитків за перші сім років роботи та їх різницю. Визначити максимальний прибуток за цей період.
14. Обчислити суми прибутків, що були в межах $Y < 170$ або $Y > 620$ (в у.о.). Скільки років фірма мала такі прибутки?
15. Обчислити суму збитків і визначити скільки років фірма була збитковою? У якому році збиток був максимальний?
16. Визначити найбільший збиток. У якому році фірма мала найбільший прибуток?
17. У які роки фірма мала найбільші прибуток та збиток?
18. Обчислити суму збитків. Чи був хоч раз нульовий баланс?
19. Обчислити суми прибутків і збитків фірми та їх різницю. Визначити максимальний збиток фірми.
20. Обчислити суму збитків, для яких справджується умова $Y < -590$ або $Y_t > -330$ (в у.о.). Визначити найбільший прибуток і в якому році він був отриманий?
21. Обчислити суму збитків фірми. У якому році прибуток був найменший? Визначити його величину.
22. Обчислити середні арифметичні всіх прибутків та збитків.
23. Обчислити суми прибутків і збитків за перші п'ять років роботи. Скільки років на протязі цього періоду фірма мала прибутки?
24. Обчислити суму прибутків, які були в межах $315 < Y < 958$ (в у.о.). У якому році збитки були найбільші?
25. Коли прибутки були більші, ніж 580 та менші, ніж 100 у.о? Коли був максимальний прибуток?

Приклад програми:

Варіант 25

Program V_25;

Var

```
y : array [ 1..10 ] of real;
x: array [1..10] of integer ;
i,k,m:integer;
a: string;
max: real;
```

Begin

```
i:=25;
```

```
For k:=1 to 10 do
```

```
begin
```

```
  x[k]:=1990+k;
```

```
  y[k]:=k*random(100)-i*random(50);
```

```
end;
```

```
For k:=1 to 10 do
```

```
begin
```

```
  if y[k]<0 then a:='збитки' else a:='прибуток';
```

```
  writeln(x[k], ' | ', y[k]:5:3, ' | ', a)
```

```
end;
```

```
For k:=1 to 10 do
```

```
begin
```

```
  if y[k]>500 then writeln('>500 ', x[k]);
```

```
  if y[k]<100 then writeln('<100 ', x[k]);
```

```
end;
```

```
max:=y[1]; m:=x[1];
```

```
For k:=2 to 10 do
```

```

if max<y[k] then
  begin
    max:=y[k];
    m:=x[k]
  end;
writeln('max=',max:5:3,' ', m)
end.

```

Задача 2. Одновимірні масиви та складний пошук.

Утворити і вивести масив y з елементами $Y(k) = k * \text{random}(100) - i * \text{random}(50)$, де i - номер варіанта, $k = 1, 12$. Виконати завдання свого варіанта. У разі відсутності шуканих даних, вивести повідомлення про це.

1. Перший додатний елемент поміняти місцями з максимальним.
2. Знайти суму третього та шостого додатних елементів.
3. Другий від'ємний елемент замінити мінімальним.
4. Скільки є елементів з мінімальним значенням серед додатних?
5. Усі додатні елементи масиву, окрім максимального, занести в інший масив.
6. Обчислити суму перших чотирьох від'ємних елементів.
7. Вивести номер передостаннього додатного елемента.
8. Елементи масиву після другого від'ємного занести в інший масив.
9. Знайти добуток другого та четвертого елементів більших, ніж 3.
10. Максимальний елемент поміняти місцями з другим нульовим.
11. Останній від'ємний елемент замінити найбільшим.
12. Обчислити добуток другого від'ємного та п'ятого елементів.
13. Чи третій додатний елемент є останнім у масиві?
14. Вивести номери двох найбільших елементів. Обчислити їх суму.
15. Чи є два елементи серед від'ємних з максимальним значенням?
16. Максимальний елемент поміняти місцями з четвертим
17. Третій додатний елемент замінити максимальним.
18. Визначити номер п'ятого від'ємного елемента.
19. Обчислити добуток перших трьох додатних елементів та визначити їхні номери.
20. Обчислити суму другого додатного та третього елементів.
21. Другий від'ємний елемент поміняти місцями з третім додатним.
22. Утворити масив елементів, значення яких є між значеннями третього та максимального елементів заданого масиву.
23. Вивести добуток номерів двох найменших елементів серед додатних.
24. Визначити суму номерів другого та третього від'ємного елементів.
25. Вивести номери другого та четвертого додатних елементів.

Задача 3. Двовимірні масиви. Простий пошук.

Утворити масив з елементами $a[n,k] = i * k * \text{random}(10) - i * n * \text{random}(10)$, де i - номер варіанта, $k, n = 1, 10$. Вивести його на екран у вигляді таблиці (матриці). Виконати додатково завдання свого варіанта.

1. Визначити індекси мінімального елемента масиву. Обчислити добуток його від'ємних елементів.
2. Обчислити кількість елементів масиву, для яких виконується нерівність $10 < A < 60$.
3. Обчислити добуток значень тих елементів, для яких справджуються нерівності $A < 10$ або $A > 10$.
4. Обчислити кількість додатних елементів та їхній добуток.
5. Обчислити суму квадратів елементів, значення яких більші, ніж 1.



6. Обчислити добуток квадратів тих елементів масиву, для яких виконується нерівність $|A| < 15$.
7. Обчислити кількість тих елементів масиву, для яких виконується нерівність $A > 30$ та суму елементів менших, ніж 9.
8. Обчислити добуток від'ємних елементів. Визначити індекси максимального елемента.
9. Обчислити суму діагональних елементів масиву та кількість від'ємних елементів.
10. Обчислити добуток тих елементів масиву, для яких виконується нерівність $2 < A < 50$.
11. Визначити індекси максимального елемента масиву. Обчислити добуток елементів над головною діагоналлю.
12. Обчислити добуток елементів перших двох рядків.
13. Обчислити суму елементів масиву над головною діагоналлю. Визначити індекси мінімального елемента.
14. Обчислити суму від'ємних елементів. Знайти максимальний елемент.
15. Обчислити добуток мінімального і максимального елементів масиву.
16. Визначити індекси мінімального і максимального елементів масиву.
17. Елементи масиву, що дорівнюють нулю, замінити на 10. Знайти суму елементів під головною діагоналлю.
18. Визначити кількість від'ємних та суму додатних елементів.
19. Обчислити добуток тих елементів, для яких виконуються нерівності $A < -5$ або $A > 30$. Визначити індекси мінімального елемента.
20. Визначити індекси максимального та мінімального елементів масиву. Обчислити їхній добуток.
21. Обчислити добуток елементів над головною діагоналлю матриці та визначити їхню кількість.
22. Обчислити середнє арифметичне додатних елементів масиву.
23. Обчислити суму тих елементів масиву, для яких виконується нерівність $10 < A < 50$. Знайти максимальний елемент.
24. Обчислити суму діагональних елементів матриці та кількість елементів, значення яких менші, ніж 3.
25. Обчислити добуток елементів під головною діагоналлю на суму елементів на головною діагоналлю.

Задача 4. Двовимірні масиви. Задача про вибори.

Нехай шість населених пунктів позначені номерами від 1 до 6 (величина k), а п'ять кандидатів — номерами від 1 до 5 (величина n). Кількість голосів, набраних кандидатами у кожному пункті визначається формулою $a[k,n]=\text{random}(10i+50)$, де i - номер варіанта. Вивести на екран таблицю результатів голосування, де у рядках є дані з населених пунктів, а у стовпцях дані щодо конкретних кандидатів. Визначити і вивести значення величин з додаткового завдання. Створити одновимірний масив з шуканими даними.

1. Які підсумкові результати кожного кандидата? (Підказка: утворити одновимірний масив з сум значень усіх стовпців таблиці).
2. Які номери населених пунктів, де кількість поданих голосів перевищила 150 (утворити одновимірний масив з цих номерів)?
3. Хто з кандидатів набрав максимальну, а хто - мінімальну кількість голосів у четвертому населеному пункті?
4. Яка кількість голосів була подана за першого і третього кандидатів у всіх населених пунктах?

5. В яких населених пунктах другий і четвертий кандидати набрали максимальну кількість голосів?
6. Скільки виборців взяли участь у голосуванні у кожному населеному пункті?
7. Хто з кандидатів має максимальний рейтинг?
8. Хто з кандидатів набрав максимальну кількість голосів у другому населеному пункті?
9. В яких населених пунктах кількість опитаних більша деякого заданого числа n ?
10. За кого з кандидатів подано кількість голосів меншу деякого заданого числа n ?
11. В яких населених пунктах перший кандидат набрав максимальну кількість голосів?
12. В якому населеному пункті проголосувало найбільше людей?
13. Хто з кандидатів набрав найбільше голосів у другому і третьому населених пунктах?
14. В якому населеному пункті перший кандидат набрав мінімальну кількість голосів, а в якому максимальну?
15. У кого з-поміж другого, четвертого і п'ятого кандидатів найвищий рейтинг?
16. Хто набрав максимальну, а хто - мінімальну кількість голосів у першому населеному пункті?
17. У яких населених пунктах перший і п'ятий кандидат набрали більше, ніж 100 голосів?
18. Які номери населених пунктів, де кількість учасників виборів не перевищила 450?
19. У кого з кандидатів рейтинг більший деякого заданого числа M ?
20. В яких містах кількість виборців менша деякого заданого числа?
21. Які кандидати набрали мінімальну кількість голосів в кожному із населених пунктів?
22. Які кандидати набрали максимальну і мінімальну кількість голосів в другому і п'ятому населених пунктах?
23. У кого з кандидатів найменший рейтинг?
24. У скількох кандидатів рейтинг перевищує деяке задане число M ?
25. В яких населених пунктах третій кандидат набрав максимальну кількість голосів?



Процедури та функції.

Теоретичний матеріал

Підпрограма - це логічно завершена частина програми, яка має ім'я і може бути викликана з будь-якого місця основної програми.

В мові програмування Турбо Паскаль підпрограми реалізуються за допомогою процедур і функцій.

Процедура (функція) - це незалежна іменована частина програми, яку можна викликати по імені для виконання наперед визначених дій. Всі процедури і функції поділяють на дві групи: вбудовані і визначені користувачем. Перші входять до складу мови програмування і викликаються для виконання по фіксованому імені (математичні функції, графічні процедури та ін.). Другі розробляються і іменуються користувачем.

Процедури і функції повинні бути описані в розділі опису даних. Всередині кожної процедури (функції) можуть використовуватись інші процедури (функції) але описані вище її по тексту.

Структура процедури(функції) повторює структуру програми, тільки замість слова PROGRAM записується слово PROCEDURE або FUNCTION.

Procedure ім'я(список формальних параметрів);
розділ опису даних;

Begin
тіло процедури;

End;
Function ім'я(список формальних параметрів):тип ;
розділ опису даних;

Begin
тіло функції;

End;
Відмінність між процедурою і функцією заключається в тому, що функція повертає в точку свого виклику одне значення тип якого повинен співпадати з типом функції. В тілі функції повинен бути використаний оператор присвоювання:

ім'я функції := вираз;

В списку формальних параметрів перераховуються імена параметрів (а не конкретні значення) і вказується їх тип.

Наприклад:

```
Procedure Geron(a,b,c: real);
Function factorial(n:integer):integer;
Procedure P2(a,d:real,c:integer,f:char);
```

Після опису процедур(функцій) їх можна викликати замінивши формальні параметри фактичними (конкретними значеннями).

Наприклад:

```
Procedure Geron(3.5,4.2,6);
Function factorial(10);
Procedure P2(-4,2.3,11,'X');
```

Приклад програми:

З трьох заданих чисел знайти максимальне і добути з нього корінь кубічний, потім знайти мінімальне і піднести його до кубу.

Словесний запис алгоритму:

Початок

ввід даних;

пошук максимального і мінімального числа;

обчислення і вивід $\max^{1/3}$;

обчислення і вивід \min^3 ;

Кінець.

Переведемо даний алгоритм на мову програмування:

Program Z;

var x1,y1,z1,min1,max1:real;

Procedure M(x,y,z:real;var min,max:real);

{ процедура пошуку мінімального і } { максимального числа }

begin

max:=x;

if max<y then max:=y;

if max<z then max:=z;

min:=x;

if min>y then min:=y;

if min>z then min:=z;

end;

Function M1(max2:real):real;

begin

M1:=exp(1/3*ln(max));

end;

Function M2(min2:real):real;

begin

M2:=exp(3*ln(min));

end;

BEGIN

read(x1,y1,z1);

M(x1,y1,z1,min1,max1);

M1(min1);

M2(max1);

writeln(min1,' 'max1);

writeln(M1,' 'M2);

END.



Приклади розв'язку задач:

1. Обчисліть площу кільця $R1 < R2$. Для обчислення площі кругів використайте функцію.

```

Program Function_1;
var
  r1, r2 : real;
  s: real;
function Kolo(r:real):real;
begin
  kolo:=3.14*sqr(r);
end;
begin
  write ('Введіть радіус більшого кола =');
  readln(r2);
  write ('Введіть радіус меншого кола =');
  readln(r1);
  s:= kolo(r2)-kolo(r1);
  writeln('S=',s:0:3);
end.

```

Ім'я програми

Розділ оголошення змінних
 Змінні: r1, r2 – дійсного типу
 Змінна: s – дійсного типу
Початок функції Kolo

Обчислення площі круга. R – формальний параметр
 Кінець функції

Початок програми

Вивід текстової підказки
 Введення числового значення
 Вивід текстової підказки
 Введення числового значення
 Обчислення площі кільця
 Вивід площі кільця на екран

Кінець програми

2. Обчисліть площу кільця $R1 < R2$. Для обчислення площі кругів використайте процедуру.

```

Program Function_2;
var
  r1, r2 : real;
  s,s1,s2: real;
Procedure Kolo(r:real);
begin
  s:=3.14*sqr(r);
end;
begin
  write ('Введіть радіус більшого кола =');
  readln(r2);
  write ('Введіть радіус меншого кола =');
  readln(r1);
  kolo(r2);
  s2:=s;
  kolo(r1);
  s1:=s;
  s:=s2-s1;
  writeln('S=',s:0:3);
end.

```

Ім'я програми

Розділ оголошення змінних
 Змінні: r1, r2 – дійсного типу
 Змінні: s,s1,s2 – дійсного типу
Початок процедури Kolo

Обчислення площі круга. R – формальний параметр
 Кінець процедури

Початок програми

Вивід текстової підказки
 Введення числового значення
 Вивід текстової підказки
 Введення числового значення
 Обчислення площі більшого круга
 Фіксація площі більшого круга
 Обчислення площі меншого круга
 Фіксація площі меншого круга
 Обчислення площі кільця
 Вивід площі кільця на екран

Кінець програми

3. Обчисліть площу опуклого чотирикутника, якщо відомо його сторони та діагональ. Для обчислення площі трикутників використайте функцію.

```

Program Function_3;
var
  a,b,c,e,d,p : real;
  s: real;
function Tr(x,y,z:real):real;

```

Ім'я програми

Розділ оголошення змінних
 Змінні: a,b,c,e,d,p – дійсного типу
 Змінна: s – дійсного типу
Початок функції Tr

```

begin
p:=(x+y+z)/2;
Tr:=sqrt(p*(p-x)*(p-y)*(p-z));
end;
begin
write ('Введіть сторони чотирикутника =');
readln(a,b,c,e);
write ('Введіть діагональ чотирикутника =');
readln(d);
s:=tr(a,b,d)+tr(c,e,d);
writeln('S=',s:0:3);
end.

```

Обчислення півпериметра трикутника
 Обчислення площі трикутника з сторонами x,y,z
 Кінець функції
Початок програми
 Вивід текстової підказки
 Введення числового значення
 Вивід текстової підказки
 Введення числового значення
 Обчислення площі чотирикутника
 Вивід площі чотирикутника на екран
Кінець програми

4. Обчисліть площу опуклого чотирикутника, якщо відомо його сторони та діагональ. Для обчислення площі трикутників використайте процедуру.

```

Program Function_4;
var
a,b,c,e,d,p : real;
s,s1,s2: real;
Procedure Tr(x,y,z:real);
begin
p:=(x+y+z)/2;
s:=sqrt(p*(p-x)*(p-y)*(p-z));
end;
begin
write ('Введіть сторони чотирикутника =');
readln(a,b,c,e);
write ('Введіть діагональ чотирикутника =');
readln(d);
tr(a,b,d);
s2:=s;
tr(c,e,d);
s1:=s;
s:=s2+s1;
writeln('S=',s:0:3);
end.

```

Ім'я програми
 Розділ оголошення змінних
 Змінні: a,b,c,e,d,p – дійсного типу
 Змінні: s,s1,s2 – дійсного типу
Початок процедури Tr
 Обчислення півпериметра трикутника
 Обчислення площі трикутника з сторонами x,y,z
 Кінець процедури
Початок програми
 Вивід текстової підказки
 Введення числового значення
 Вивід текстової підказки
 Введення числового значення
 Обчислення площі першого трикутника
 Фіксація площі першого трикутника
 Обчислення площі другого трикутника
 Фіксація площі другого трикутника
 Обчислення площі чотирикутника
 Вивід площі чотирикутника на екран
Кінець програми

5. Обчисліть значення виразу $Y = \text{ctg}^2(x) + 3 \text{ctg}(2x)$. Для обчислення ctg використайте функцію.

```

Program Function_5;
var
x,y: real;
function Ctg(z:real):real;
begin
ctg:=cos(z)/sin(z);
end;
begin
write ('Введіть X =');
readln(x);
y:=sqrt(ctg(x))+3*ctg(2*x);
writeln('Y=',y:0:3);
end.

```

Ім'я програми
 Розділ оголошення змінних
 Змінні: x,y – дійсного типу
Початок функції Ctg
 Обчислення $\text{ctg}(z)$
 Кінець функції
Початок програми
 Вивід текстової підказки
 Введення числового значення
 Обчислення значення виразу
 Вивід результату на екран
Кінець програми



6. Обчисліть значення виразу $U=f(0.5,a)+f(a+b,a-b)$. Де $F(x,y) = \frac{x^2 + xy}{1 + x + y}$

```

Program Function_6;
var
  a,b,u: real;
function F(x,y:real):real;
begin
  F:=(sqr(x)+x*y)/(1+x+y);
end;
begin
  write ('Введіть A =');
  readln(a);
  write ('Введіть A =');
  readln(a);
  u:= f(0.5,a)+f(a+b,a-b);
  writeln('U=',u:0:3);
end.
    
```

Ім'я програми
 Розділ оголошення змінних
 Змінні: a,b,u – дійсного типу
Початок функції F
 Обчислення F(x,y)
 Кінець функції
Початок програми
 Вивід текстової підказки
 Введення числового значення
 Вивід текстової підказки
 Введення числового значення
 Обчислення значення виразу
 Вивід результату на екран
Кінець програми

7. Обчисліть значення виразу $Y=f(a)+f(b)-f^2(a)$. Де $F(x) = 2x^2 - \sqrt{x}$

```

Program Function_7;
var
  a,b,y: real;
function F(x:real):real;
begin
  F:=2*sqr(x)-sqrt(x);
end;
begin
  write ('Введіть A =');
  readln(a);
  write ('Введіть B =');
  readln(B);
  y:= f(a)+f(b)-sqr(f(a));
  writeln('Y=',y:0:3);
end.
    
```

Ім'я програми
 Розділ оголошення змінних
 Змінні: a,b – дійсного типу
Початок функції F(x)
 Обчислення F(x)
 Кінець функції
Початок програми
 Вивід текстової підказки
 Введення числового значення
 Вивід текстової підказки
 Введення числового значення
 Обчислення значення виразу
 Вивід результату на екран
Кінець програми

8. Задано координати вершин трикутника x_1,y_1,x_2,y_2,x_3,y_3 . Обчислити його периметр. Для обчислення довжини сторони трикутника використати підпрограму.

```

Program Function_8;
var
  x1,y1,x2,y2,x3,y3,P: real;
function F(a1,b1,a2,b2:real):real;
begin
  F:=sqrt(sqr(a2-a1)+sqr(b2-b1));
end;
begin
  write ('Введіть X1,Y1 =');
  readln(x1,y1);
  write ('Введіть X2,Y2 =');
    
```

Ім'я програми
 Розділ оголошення змінних
 Змінні: x1,y1,x2,y2,x3,y3,P– дійсного типу
Початок функції F(x)
 Обчислення F(x)
 Кінець функції
Початок програми
 Вивід текстової підказки
 Введення числового значення
 Вивід текстової підказки

```

readln(x2,y2);
write ('Введіть X3,Y3 =');
readln(x3,y3);
p:= f(x1,y1,x2,y2)+ f(x2,y2,x3,y3)+
f(x3,y3,x1,y1);
writeln('P=',p:0:3);
end.

```

Введення числового значення
Вивід текстової підказки
Введення числового значення
Обчислення значення виразу
Вивід результату на екран
Кінець програми

9. Задано координати діагоналі прямокутника x_1, y_1, x_2, y_2 . Обчислити його периметр. Для обчислення довжини сторони трикутника використати підпрограму.

```

Program Function_9;
var
  x1,y1,x2,y2,P,y: real;
function F(a1,b1,a2,b2:real):real;
begin
  F:=sqrt(sqr(a2-a1)+sqr(b2-b1));
end;
begin
  write ('Введіть X1,Y1 =');
  readln(x1,y1);
  write ('Введіть X2,Y2 =');
  readln(x2,y2);
  y:= 2*(f(x1,0,x2,0)+f(y1,0,y2,0));
  writeln('Y=',y:0:3);
end.

```

Ім'я програми
Розділ оголошення змінних
Змінні: x_1, y_1, x_2, y_2, P – дійсного типу
Початок функції F(x)

Обчислення $F(a_1, b_1, a_2, b_2)$
Кінець функції
Початок програми
Вивід текстової підказки
Введення числового значення
Вивід текстової підказки
Введення числового значення
Обчислення значення виразу
Вивід результату на екран
Кінець програми

10. Створити програму для визначення максимального з чотирьох довільних чисел. Створити та використати підпрограму для визначення максимального з двох чисел.

```

Program Function_10;
var
  a,b,c,d,max: real;
function F(x,y:real):real;
begin
  IF x>y then F:=x else F:=y;
end;
begin
  write ('Введіть A,B,C,D через пропуск =');
  readln(a,b,c,d);
  max:=F(F(a,b),F(c,d));
  writeln('Max=',max:0:3);
end.

```

Ім'я програми
Розділ оголошення змінних
Змінні: a, b, c, d, max – дійсного типу
Початок функції F(x)

Обчислення максимального з двох чисел
Кінець функції
Початок програми
Вивід текстової підказки
Введення числового значення
Обчислення значення виразу
Вивід результату на екран
Кінець програми

Практична робота

"Створення і реалізація програм з використанням підпрограм"

Мета роботи:

1. Отримання навиків використання підпрограм.
2. Розвиток практичних навиків роботи в середовищі програмування.

Завдання:



Скласти програму розв'язку задачі згідно варіанту. В зошити записати текст програми та результат виконання.

Задача 1. Підпрограми.

У підрозділі Y є 15 співробітників, а в G - 20. Протягом місяця вони відпрацювали деяку кількість днів, яка задана як випадкова величина зі значеннями $Y(n)=\text{random}(31)$, $n = 1,15$, $G(k)=\text{random}(31)$, $k = 1,20$. Денна оплата праці d у.о. Податкова ставка 20%. Використовуючи підпрограми, утворити масиви u , g , вивести значення їх елементів на екран і виконати завдання пошуку даних для кожного підрозділу.

Вивести повідомлення, якщо шуканих даних немає.

1. Скільки осіб працювали у кожному підрозділі більше 15 днів?
2. Хто найменше заробив у кожному підрозділі?
3. Кому нараховано більше, ніж 100 у.о. у кожному підрозділі?
4. Скільки людино-днів було відпрацьовано у кожному підрозділі?
5. Який середній заробіток у кожному підрозділі?
6. Скільки осіб отримали більше, ніж 50 і менше, ніж 120 у.о.?
7. Скільки осіб працювали менше, ніж 10 днів?
8. Яка сума податку була сплачена у кожному підрозділі?
9. Хто сплатив найбільший податок у кожному підрозділі?
10. У скількох осіб податок перевищив 20 у.о.?
11. Який середній податок був у кожному підрозділі?
12. У якому підрозділі більший середній заробіток?
13. Хто сплатив найменший податок у кожному підрозділі?
14. Скільки осіб працювали лише один день у кожному підрозділі?
15. У скількох осіб заробіток вищий за середній?
16. У якому підрозділі менший середній заробіток?
17. У скількох осіб заробіток відхиляється від середнього менше, ніж на 10%?
18. У якому підрозділі був зафіксований найбільший заробіток?
19. Скільки осіб працювали більше, ніж 5 і менше, ніж 12 днів?
20. Який середній заробіток перших п'яти осіб?
21. У скількох осіб заробіток був менший за середній?
22. Який середній заробіток останніх чотирьох осіб?
23. У якому підрозділі було відпрацьовано більшу кількість людино-днів?
24. Хто заробив більше, ніж 100 і менше, ніж 200 у.о.?
25. Скільки осіб працювали 2, 3 або 4 днів?
26. Завдання підвищеної складності. Яка кількість відпрацьованих днів найчастіше була зафіксована у кожному підрозділі?

Рядкові величини.**Теоретичний матеріал****Рядкові величини та їх опис**

Символьний тип даних (CHAR) дозволяє програмісту працювати з окремими символами тексту. Для обробки більш великих текстових одиниць - стрічок в Турбо-Паскалі введено особливий тип даних - STRING (стрічка).

Значеннями даного типу є стрічки до яких можуть входити будь-які символи, наприклад: 'Школа', 'Turbo Pascal 6.0', '145.76'.

Максимальна довжина рядка не повинна перевищувати 255 символів.

Рядкові величини описуються в блоці VAR, для цього використовують службове слово **String** після якого в квадратних дужках вказують максимальне значення довжини рядка.

Наприклад:

```
Var  a:string[15];
      b,b1:string[8];
      v:string;
```

Якщо довжина рядка не вказана, то вона автоматично приймає значення - 255.

Рядкові величини можна описувати в блоці Const.

Наприклад:

```
Const v='школа';
       k='Turbo Pascal';
```

В пам'яті змінні типу string займають на одну комірку більше ніж їх максимальна довжина. Звертання до елементів стрічки здійснюється так само як і до елементів масивів - по індексу. Індекс змінюється від 0 до максимальної довжини. В нульовому елементі зберігається інформація про довжину рядка, починаючи з першого - сам рядок.

Вказівки і функції опрацювання рядкових величин.

Для роботи з рядковими величинами визначено ряд операцій і функцій.

1.Функція **Concat** - з'єднує рядки в один в порядку їх запису.

Приклад:

```
Concat(x,a,s);
```

де x,a,s - рядкові величини

Дану операцію можна замінити використавши операцію "+" - $y:=x+a+s$;

2.Функція **Length** - видає фактичну довжину рядкової величини.

Приклад:

```
Length(s1);
```

Дану операцію можна замінити на $ord(s1[0])$;

3.Функція **Copy** - копіює фрагмент рядка S починаючи з позиції N довжиною L.

Приклад:

```
Copy(s,n,l);
```

```
Copy(s1,5,3);
```

4.Функція **Pos** - виводить номер позиції, з якої стрічка s1 входить перший раз в стрічку s2 -> $Pos(s1,s2)$;

Приклад:

```
Pos('a','інформатика') -> 7
```

5.Процедура **Insert** - вставляє задане слово в рядок S починаючи з позиції N.

Приклад:

```
s:='МП Pascal';
```

```
Insert('Turbo',s,4);
```

Результат -> МП TurboPascal



6. Процедура **Delete** - вилучає з рядка S починаючи з позиції N, K символів.

Приклад:

```
s:='Turbo';
```

```
Delete(s,2,3);
```

Результат 'To'.

7. Процедура **Str(x,s)** - переводить числове значення - X в відповідний рядок і присвоює його змінній S.

Приклад:

```
X:=195;
```

```
Str(x,s);
```

Результат S -> '195'

8. Процедура **Val(s,x,c)** - виконує дію обернену до Str.

Приклад програми:

Скласти програму, яка перевіряє чи являється задана літерна величина "перевертисем".

```
program qw;
var a:string;c:char;
n,k,i:integer;
label 1,2,3;
begin
  readln(a);
  n:=length(a);
  if n/2=trunc(n/2) then k:=(n div 2) else k:=trunc(n/2);
  for i:=1 to k do
  begin
    if copy(a,i,1)=copy(a,n+1-i,1)then goto 1 else goto 2;
  1:end;writeln("Так");goto 3;
  2:writeln("НІ");
  3:readln;
end.
```

Приклади розв'язку задач:

1. Скласти програму що підраховує кількість літер "а" в заданій рядковій величині X.

<pre>Program String_1; var x:string; k,d,i:integer; begin write ('Введіть X ='); readln(x); d:=length(x); k:=0; For i:=1 to d do IF copy(x,i,1)='a' then k:=k+1; writeln(' K=',k); end.</pre>	<p>Ім'я програми Розділ оголошення змінних Змінна: x – рядкового типу Змінні: k,d,i – цілого типу</p> <p>Початок програми Вивід текстової підказки Введення рядка Визначення довжини рядка Введення початкового значення для кількості Змінюємо номер символу в рядку Якщо скопійований символ – "а". то збільшуємо кількість на 1. Виведення на екран отриманого результату</p> <p>Кінець програми</p>
---	--

2. Скласти програму що знищує всі літери "а" в заданій рядковій величині X.

Program String_2;

```
var
  x:string;
  k,d,i:integer;
begin
  write ('Введіть X =');
  readln(x);
  d:=length(x);
  k:=0;
  For i:=1 to d do
    IF copy(x,i,1)='a' then
      begin
        delete(x,i,1);
        i:=i-1;
        d:=d-1;
      end;

  writeln(' X=',x);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – рядкового типу

Змінні: k,d,i – цілого типу

Початок програми

Вивід текстової підказки

Введення рядка

Визначення довжини рядка

Введення початкового значення для кількості

Змінюємо номер символу в рядку

Якщо скопійований символ – "a", то

Знищуємо символ

Повертаємось на символ назад

Зменшуємо довжину рядка на один символ

Виведення на екран отриманого результату

Кінець програми

3. Скласти програму що подвоєє літери "а" в заданій рядковій величині X і знаходить їх кількість.

Program String_3;

```
var
  x:string;
  k,d,i:integer;
begin
  write ('Введіть X =');
  readln(x);
  d:=length(x);
  k:=0;
  For i:=1 to d+2 do
    IF copy(x,i,1)='a' then
      begin
        k:=k+1;
        insert('a',x,i);
        i:=i+1;
        d:=d+1;
      end;

  writeln(' K=',k);
  writeln(' X=',x);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – рядкового типу

Змінні: k,d,i – цілого типу

Початок програми

Вивід текстової підказки

Введення рядка

Визначення довжини рядка

Введення початкового значення для кількості

Змінюємо номер символу в рядку

Якщо скопійований символ – "a", то

Кількість збільшуємо на 1

Вставляємо символ - a

Перескакуємо через вставлений символ

Збільшуємо довжину рядка на один символ

Виведення на екран отриманого результату

Кінець програми

4. Скласти програму що визначає номер останньої літери "а" в заданій рядковій величині X.

Program String_4;

```
var
  x:string;
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – рядкового типу



```
k,d,i:integer;
```

```
begin
```

```
  write ('Введіть X =');
```

```
  readln(x);
```

```
  d:=length(x);
```

```
  k:=0;
```

```
  For i:=1 to d do
```

```
    IF copy(x,i,1)='a' then k:=i;
```

```
    writeln(' K=',k);
```

```
end.
```

Змінні: k,d,i – цілого типу

Початок програми

Вивід текстової підказки

Введення рядка

Визначення довжини рядка

Введення початкового значення для номеру.

Змінюємо номер символу в рядку

Якщо скопійований символ – "a", то фіксуємо номер

Виведення на екран отриманого результату

Кінець програми

5. Скласти програму що замінює літеру "a" на літеру "o" в заданій рядковій величині X.

Program String_5;

```
var
```

```
  x:string;
```

```
  k,d,i:integer;
```

```
begin
```

```
  write ('Введіть X =');
```

```
  readln(x);
```

```
  d:=length(x);
```

```
  For i:=1 to d do
```

```
    IF copy(x,i,1)='a' then
```

```
      begin
```

```
        delete(x,i,1);
```

```
        insert('o',x,i);
```

```
      end;
```

```
    writeln(' X=',x);
```

```
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: x – рядкового типу

Змінні: k,d,i – цілого типу

Початок програми

Вивід текстової підказки

Введення рядка

Визначення довжини рядка

Змінюємо номер символу в рядку

Якщо скопійований символ – "a", то

Знищуємо літеру "a"

Вставляємо літеру "o"

Виведення на екран отриманого результату

Кінець програми

Практична робота

"Створення і реалізація програм з використанням рядкових величин"

Мета роботи:

1. Отримання навиків використання рядкових величин.
2. Розвиток практичних навиків роботи в середовищі програмування.

Завдання:

Скласти програму розв'язку задачі згідно варіанту. В зошити записати текст програми та результат виконання.

Задача 1. Дані типу рядки (string).

Ввести прізвище, ім'я та по батькові як одне дане типу рядок. Визначити довжину рядка і кількість букв "a" у ньому. Виконати додатково завдання свого варіанта двома способами:

а) розглядаючи рядок як масив символів;

б) застосовуючи до рядка функції та процедури.

1. Вивести ім'я та кількість букв у третьому слові.
2. Визначити скільки букв 'a' є у прізвищі.
3. Вивести три букви - свої ініціали з крапками.
4. Вивести довжини прізвища та імені.
5. Вивести прізвище та ініціали.
6. Вивести ім'я та кількість букв у прізвищі.
7. Визначити скільки букв 'o' є в імені.

8. Вивести найдовше слово.
9. Вилучити усі букви 'а' та 'о' з прізвища.
10. Вивести ім'я у стовпчик.
11. Чи починається хоч би одне слово з букви 'М'?
12. Усі букви 'і' в імені продублювати.
13. Вивести прізвище та кількість букв у імені.
14. Вивести ім'я у зворотному порядку.
15. Вивести прізвище у стовпчик.
16. Вивести ім'я та по батькові та кількість букв у імені.
17. Вивести найкоротше слово.
18. Вивести дане без пропусків. Скільки букв е в імені?
19. Вивести довжини трьох слів.
20. Вивести ім'я та кількість букв у прізвищі.
21. Вивести ім'я, прізвище.
22. Кожну букву імені продублювати.
23. Вивести прізвище у зворотному порядку.
24. Визначити скільки букв 'а' та 'б' є у прізвищі.
25. Вивести третє слово та кількість букв у прізвищі.

Задача 2. Дані типу рядки (string). Криптографія.

Придумати та описати словесне власний спосіб шифрування тексту. Скласти програму для введення тексту як даного типу string (до 255 символів), його шифрування, дешифрування і виведення результатів. Якщо вхідний і дешифрований тексти збігаються, вивести "ОК".



Графічні вказівки. Побудова графіків функцій.

Теоретичний матеріал

Графічний режим

Для створення програм з використанням графіки на мові Турбо Паскаль необхідно підключити модуль графічних процедур - *Graph*, а також включити графічний режим роботи дисплея за допомогою команди - *InitGraph*..

Команда *InitGraph* має три параметри -

InitGraph(ім'я драйвера, режим, шлях до драйвера)

Ім'я драйвера:

Detect = 0;

CGA = 1;

EGA = 3;

IBM8514 = 6;

VGA = 9;

Режим -

VGALO = 0;

VGAMED = 1;

VGAHI = 2;

Шлях до драйвера - вказується шлях до каталогу в якому знаходиться відповідний драйвер підтримки вибраного режиму.

Наприклад:

```
InitGraph(vga,2,'c:\tp\bgi');
```

```
InitGraph(3,0,'c:\tp');
```

Якщо тип адаптера невідомий або програма виконуватиметься на різних комп'ютерах, потрібно забезпечити автоматичне тестування. На місці третього параметра вказати порожній рядок - "" (драйвер знаходиться в каталозі в одному каталозі з компілятором Turbo Pascal). Драйвери - це спеціалізовані файли з розширенням *.bgi* (*cga.bgi*, *vga.bgi*). Після завершення роботи в графічному режимі його закривають процедурою *CloseGraph* для переходу в текстовий режим роботи.

Графічні процедури

PutPixel(X,Y,C) - виводить на екран точку з координатами *X*, *Y* кольором *C*.

Line(X1,Y1,X2,Y2) - виводить на екран відрізок між точками з координатами *X1*, *Y1*, *X2*, *Y2*. Колір і стиль поточні.

LineTo(X,Y) - проводить відрізок від поточної точки до точки з координатами *X*, *Y*.

LineRel(DX,DY) - проводить відрізок від поточної точки з приростом *DX*, *DY*.

Rectangle(X1,Y1,X2,Y2) - виводить на екран прямокутник.

Bar(X1,Y1,X2,Y2) - виводить на екран прямокутник зафарбований поточним стилем. *X1*, *Y1*- координати лівого верхнього кута, *X2*, *Y2*- координати правого нижнього кута

Circle(X,Y,R) - виводить на екран коло радіуса *R* і центром в точці з координатами *X*, *Y*. Колір поточний.

Arc(X,Y,a,b,R) - виводить на екран дугу поточного кольору. *X*, *Y* - координати центра, *a*, *b* - початковий і кінцевий кут (в градусах), *R* - радіус.

PiesLice(X,Y,a,b,R) - виводить на екран сектор зафарбований поточним стилем

Ellipse(X,Y,a,b,Rx,Ry) - виводить на екран еліпсну дугу (еліпс), *Rx*, *Ry* - горизонтальний і вертикальний радіуси.

FillPoly(N,координати) - виводить на екран замкнутий багатокутник зафарбований поточним стилем.

DrawPoly(N,координати) - виводить на екран ламану лінію. *N*- кількість точок злому(вершин), координати - набір координат точок злому.

SetColor(C) - встановлює колір ліній та символів.

SetBKColor(C) - встановлює колір фону.

0 - чорний	4 - червоний	8 - темно-сірий	12 - рожевий
1 - синій	5 - фіолетовий	9 - яскраво-синій	13 - малиновий
2 - зелений	6 - коричневий	10 - яскраво-зелений	14 - жовтий
3 - блакитний	7 - світло-сірий	11 - яскр.-блакитний	15 - білий

SetFillStyle(код, C) - встановлює стиль і колір зафарбовування від-повідно до коду

- C (колір):

1 – суцільне 2 – лініями 7 – символом '+' 11 – крапками

FloodFill(X,Y, колір межі) - зафарбовує замкнену область поточним стилем. X,Y - координати точки всередині замкненої області.

OuttextXY(X,Y,'text') - виводить заданий текст починаючи з точки з координатами X,Y.

ClearDevice - очищає екран.

Repeat until keypressed - затримка зображення на екрані.

Наприклад:

```

program aa;
uses Graph, Crt;
var gd,gm : integer;
begin
gd:=3;gm:=2;
InitGraph ( gd, gm, 'c:\tp\bgi');
setfillstyle (3,11);
bar(50,350,590,400);
Line(250,375,390,375);
Line(150,300,490,300);
Line(150,300,250,375);
Line(490,300,390,375);
Line(320,75,320,125);
Line(320,200,320,300);
Line(265,125,390,125);
Line(265,200,390,200);
arc (390,162,270,90,37);
arc (265,162,270,90,37);
setfillstyle (1,15);
floodfill(310,175,15);
setfillstyle (11,7);
floodfill(295,340,15);
CloseGraph;
Repeat until keypressed;
end.
```

Програми, в яких передбачено створення рухомих графічних зображень, називають *анімаційними*. Найпростішим способом переміщення певного графічного фрагменту є послідовне відтворення його вздовж певної траєкторії та “замальовування” кольором фону попереднього фрагменту.

Наприклад:

```

program K1;
uses Graph;
var gd,gm,i,j,x,k,y : integer;
begin
gd:=3;gm:=2;
InitGraph ( gd, gm, 'c:\tp\bgi');
```



```

setfillstyle (3,11);    {побудова зображення }
bar(50,350,590,400);
Line(250,375,390,375);
Line(150,300,490,300);
Line(150,300,250,375);
Line(490,300,390,375);
Line(320,75,320,125);
Line(320,200,320,300);
Line(265,125,390,125);
Line(265,200,390,200);
arc (390,162,270,90,37);
arc (265,162,270,90,37);
setfillstyle (1,15);
floodfill(310,175,15);
setfillstyle (11,7);
floodfill(295,340,15);
y:=140; k:=-1; {створення рухомого зображення }
for x:=26 to 614 do
begin
  setcolor(15);
  circle(x*2,y,15);
  for i:=1 to 9000 do;
  if y<26 then k:=1;
  setcolor(0);
  circle(x*2,y,15);
  y:=y+k;
end;
CloseGraph;
end.

```

Створювати рухомі зображення можна також методом “покадрової прокрутки”, розмістивши окремі кадри на різних відеосторінках та виводячи їх по чергово на екран.

Наприклад:

```

program Kl1;
  uses Graph;
  var gd,gm,i,j : integer;
  p: pointer;
  size: word;
begin
  gd:=3;gm:=2;
  InitGraph ( gd, gm, 'c:\tp\bgi');
  setcolor(15);    {створення зображення корабля}
  line(100,150,400,150);
  line(50,100,450,100);
  line(50,100,200,250);
  line(450,100,300,250);
  line(200,250,200,270);
  line(300,250,300,270);
  line(200,270,150,300);
  line(300,270,350,300);

```



```

line(150,300,350,300);
line(150,150,250,250);
line(250,150,250,250);
line(300,150,250,250);
line(350,150,250,250);
line(200,150,250,250);
bar(150,100,200,150);
Size:=imagesize(145,95,205,155); {створення рух. зображ.}
getmem(p,size);
getimage(145,95,205,155,p^);
for j:=1 to 9000 do
for i:=1 to 150 do
begin
putimage(145+i,95,p^,normalput);
end;
for i:=1 to 150 do
begin
putimage(295-i,95,p^,normalput);
end;
readln;
CloseGraph;
end.

```

Задачи

1. Скласти програму, яка виводить на екран зображення червоного кола.

<pre> Program Graph_1; uses Graph; var gd,gm : integer; begin gd:=3;gm:=2; InitGraph (gd, gm, 'c:\tp\bgi'); setcolor(4); circle(320,240,100); readln; closegraph; end. </pre>	<p>Ім'я програми Підключення графічного модуля Розділ оголошення змінних Змінні: gd,gm – цілого типу</p> <p>Початок програми Задання параметрів графічного режиму Ініціалізація графічного режиму Встановлення червоного кольору ліній Побудова кола Затримка зображення Закриття графічного режиму</p> <p>Кінець програми</p>
---	---

2. Скласти програму, яка виводить на екран зображення зеленого прямокутника.

<pre> Program Graph_2; uses Graph; var gd,gm : integer; begin gd:=9;gm:=2; InitGraph (gd, gm, 'c:\tp\bgi'); setcolor(2); rectangle(320,240,100,400); readln; closegraph; end. </pre>	<p>Ім'я програми Підключення графічного модуля Розділ оголошення змінних Змінні: gd,gm – цілого типу</p> <p>Початок програми Задання параметрів графічного режиму Ініціалізація графічного режиму Встановлення зеленого кольору ліній Побудова прямокутника(координати довільні) Затримка зображення Закриття графічного режиму</p> <p>Кінець програми</p>
--	---



3. Скласти програму, яка виводить на екран зображення концентричних кілець різного кольору.

```

Program Graph_3;
  uses Graph;
  var
    gd,gm,r,c : integer;
  begin
    gd:=9;gm:=2;
    InitGraph ( gd, gm, 'c:\tp\bgi');
    r:=5;
    while r<100 do
      begin
        c:=random(15);
        setcolor(c);
        circle(320,240,r);
        r:=r+5;
      end;
    readln;
    closegraph;
  end.

```

Ім'я програми
 Підключення графічного модуля
 Розділ оголошення змінних
 Змінні: gd,gm,r,c – цілого типу

Початок програми
 Задання параметрів графічного режиму
 Ініціалізація графічного режиму
 Встановлення початкового радіусу
 Поки R<100 виконувати
 Початок циклу
 Присвоїти змінній C довільне число від 0 до 15
 Встановлення відповідного кольору ліній
 Побудова кола радіуса – R
 Зм
 Затримка зображення
 Закриття графічного режиму

Кінець програми

4. Скласти програму, яка виводить на екран зображення блакитного кільця.

```

Program Graph_4;
  uses Graph;
  var
    gd,gm : integer;
  begin
    gd:=9;gm:=2;
    InitGraph ( gd, gm, 'c:\tp\bgi');
    setcolor(11);
    setfillstyle(1,11);
    circle(320,240,100);
    circle(320,240,200);
    floodfill(320,350,11);

    readln;
    closegraph;
  end.

```

Ім'я програми
 Підключення графічного модуля
 Розділ оголошення змінних
 Змінні: gd,gm – цілого типу

Початок програми
 Задання параметрів графічного режиму
 Ініціалізація графічного режиму
 Встановлення блакитного кольору ліній
 Встановлення суцільного, блакитного зафарбування
 Побудова внутрішнього кола
 Побудова зовнішнього кола
 Зафарбування кільця поточним стилем – суцільний, блакитний
 Затримка зображення
 Закриття графічного режиму

Кінець програми

5. Скласти програму, яка виводить на екран зображення еліпса (горизонтальний).

```

Program Graph_5;
  uses Graph;
  var
    gd,gm : integer;
  begin
    gd:=9;gm:=2;
    InitGraph ( gd, gm, 'c:\tp\bgi');
    setcolor(11);
    setfillstyle(1,11);
    ellipse(320,240,0,360,200,100);
    floodfill(320,245,11);

    readln;
  end.

```

Ім'я програми
 Підключення графічного модуля
 Розділ оголошення змінних
 Змінні: gd,gm – цілого типу

Початок програми
 Задання параметрів графічного режиму
 Ініціалізація графічного режиму
 Встановлення блакитного кольору ліній
 Встановлення суцільного, блакитного зафарбування
 Побудова еліпса
 Зафарбування еліпса поточним стилем – суцільний, блакитний
 Затримка зображення

```
closegraph;
end.
```

Закриття графічного режиму
Кінець програми

6. Скласти програму, яка виводить на екран графік функції $y=\sin(x)$, зміна параметра x - довільна.
7. Скласти програму, яка виводить на екран графік функції $y=x^2$, зміна параметра x - довільна.
8. Скласти програму, яка виводить на екран графік функції $y=3*x+25$, зміна параметра x - довільна.
9. Скласти програму, яка виводить на екран графік функції $y=1/x^{1/2}$, зміна параметра x - довільна.
10. Скласти програму, яка виводить на екран графік функції $y=\cos(x)^2$, зміна параметра x - довільна.

Практична робота

"Створення і реалізація програм з використанням графічних вказівок"

Мета роботи:

1. Отримання навиків використання графічних вказівок.
2. Розвиток практичних навиків роботи в середовищі програмування.

Завдання:

Скласти програму розв'язку задачі згідно варіанту. В зошити записати текст програми та результат виконання.

Задача 1. Графіка.

Моя емблема. У заданій частині графічного екрана намалювати фігуру 1, у середині фігури 1 - фігуру 2, а у середині фігури 2 - текст: своє прізвище та ініціали. Усі елементи рисунка виконати різними кольорами. Замкнені області залити кольорами.

Частини екрана, типи фігур визначають згідно з варіантом i :

№	Частина екрана	Фігура 1	Фігура 2
1.	Верхня половина	Коло	Квадрат
2.	Нижня половина	Прямокутник	Коло
3.	Ліва половина	Трикутник	Еліпс
4.	Права половина	Еліпс	Прямокутник
5.	Верхня права чверть	Квадрат	Трикутник
6.	Нижня ліва чверть	Коло	Еліпс
7.	Нижня права чверть	Прямокутник	Коло
8.	Верхня ліва чверть	Трикутник	Прямокутник
9.	Весь екран	Еліпс	Трикутник
10.	Права половина	Прямокутник	Коло
11.	Верхня права чверть	Трикутник	Прямокутник
12.	Нижня ліва чверть	Еліпс	Трикутник
13.	Нижня права чверть	Коло	Еліпс
14.	Верхня ліва чверть	Прямокутник	Квадрат
15.	Весь екран	Трикутник	Коло

Задача 2. Графік функції.

Скласти програму рисування графіка функції $y=tg(x)$ на деякому проміжку (проміжок задати самостійно). У результаті експериментів розташувати його правильно на екрані.



Робота з файлами

Теоретичний матеріал

Файл - це об'єкт, який складається з сукупності даних одного типу.

Файл описується в розділі опису типів даних TYPE та в розділі опису змінних - VAR.

Наприклад:

1)Type

tr = file of real;
ch = file of char;

2)Type

c = file of integer;

Var

k1 : c;

Процедури і функції для роботи з файлами.

Assign (ім'я файла, зовнішнє ім'я) - зв'яже ім'я файла, який використовується в програмі з іменем файла, який знаходиться на диску.

Наприклад: Assign(text, t.dat)

Reset (ім'я файла) - відкриває уже існуючий файл для зчитування даних.

Наприклад: Reset (text)

Rewrite (ім'я файла) - створює і відкриває новий файл для запису даних. Якщо вказаний файл уже існує на диску то його вміст знищується а файл відкривається.

Close (ім'я файла) - закриває відкритий файл.

Запис і зчитування даних з файла

Write (ім'я файла, ім'я змінної) - дозволяє записувати дані в файл на диску.

Read (ім'я файла, ім'я змінної) - дозволяє читати дані з файла на диску.

Приклади розв'язку задач:

1. В файлі z.dat задано сторону квадрату. Скласти програму для знаходження його периметру: $P = 4 \cdot a$. Результат обчислень вивести на екран.

Program File_1;

```
var
  a:real;
  p:real;
  f: text;
begin
  assign (f,'z.dat');
  reset(f);
  readln(f,a);
  close(f);
  p:=4*a;
  writeln('Периметр квадрату =',p:0:2);
end.
```

Ім'я програми

Розділ оголошення змінних

Змінна: a – дійсного типу

Змінна: p – дійсного типу

Змінна: f – типу text;

Початок програми

Пов'язуємо файлову змінну F з файлом Z.dat

Відкриваємо файл Z.dat для зчитування

Читаємо з файлу значення змінної – A

Закриваємо файл Z.dat

Обчислюємо периметр квадрату

Виведення на екран отриманого результату

Кінець програми

2. В файлі z.dat задано в одному рядку сторони прямокутника – a, b. Скласти програму для знаходження його площі: $S = a \cdot b$, та периметру: $P = 2 \cdot (a + b)$. Результат обчислень вивести на екран.

Program File_2;

```
var
  a, b: real;
  s, p: real;
  f: text;
begin
  assign (f, 'z.dat');
  reset(f);
  read (f, a, b);
  close(f);
  s := a * b;
  p := 2 * (a + b);
  writeln('Площа прямокутника =', p:0:2);
  writeln('Периметр прямокутника =', p:0:2);
end.
```

Ім'я програми

Розділ оголошення змінних
Змінні: a, b – дійсного типу
Змінні: s, p – дійсного типу
Змінна: f – типу text;

Початок програми

Пов'язуємо файлову змінну F з файлом Z.dat
Відкриваємо файл Z.dat для зчитування
Читаємо з файлу значення змінних – A та B
Закриваємо файл Z.dat
Обчислюємо площу прямокутника
Обчислюємо периметр прямокутника
Виведення на екран площі прямокутника
Виведення на екран периметру прямокутника

Кінець програми

3. В файлі z.dat задано в трьох рядках сторони трикутника. Скласти програму для знаходження його периметру. результат обчислень вивести на екран.

Program File_3;

```
var
  a, b, c: real;
  p: real;
  f: text;
begin
  assign (f, 'z.dat');
  reset(f);
  readln(f, a);
  readln(f, b);
  readln(f, c);
  close(f);
  p := a + b + c;
  writeln('Периметр трикутника =', p:0:2);
end.
```

Ім'я програми

Розділ оголошення змінних
Змінні: a, b, c – дійсного типу
Змінна: p – дійсного типу
Змінна: f – типу text;

Початок програми

Пов'язуємо файлову змінну F з файлом Z.dat
Відкриваємо файл Z.dat для зчитування
Читаємо з файлу значення змінної – A
Читаємо з файлу значення змінної – B
Читаємо з файлу значення змінної – C
Закриваємо файл Z.dat
Обчислюємо периметр трикутника
Виведення на екран отриманого результату

Кінець програми

4. В файлі z.dat задано радіус кола – R. Скласти програму для знаходження його площі. Результат обчислень вивести у файл z.sol.

Program File_4;

```
var
  r: real;
  s: real;
  f, v: text;
begin
  assign (f, 'z.dat');
  reset(f);
  readln(f, r);
```

Ім'я програми

Розділ оголошення змінних
Змінна: r – дійсного типу
Змінна: s – дійсного типу
Змінні: f, v – типу text;

Початок програми

Пов'язуємо файлову змінну F з файлом Z.dat
Відкриваємо файл Z.dat для зчитування
Читаємо з файлу значення змінної – R



```
close(f);
s:=3.14*sqr(r);
assign (v,'z.sol');
rewrite(v);
writeln(v,'Площа круга =',s:0:2);
close(v);
end.
```

Закриваємо файл Z.dat
Обчислюємо площу круга радіуса – R
Пов'язуємо файлову змінну V з файлом Z.sol
Створюємо файл Z.sol для запису інформації
Виводимо отриманий результат у файл Z.sol
Закриваємо файл Z.sol
Кінець програми

5. Обчислити відстань між двома точками з координатами X1,Y1; X2,Y2.
Результат обчислень вивести у файл z.dat та на екран.

Program File_5;

```
var
x1,y1,x2,y2: real;
s: real;
f: text;
begin
assign (f,'z.dat');
rewrite(f);
readln(x1,y1);
readln(x2,y2);
s:=sqrt(sqr(x1- x2)+sqr(y1-y2));
writeln('S =',s:0:2);
writeln(f,'S =',s:0:2);
close(f);
end.
```

Ім'я програми

Розділ оголошення змінних
Змінні: x1,y1,x2,y2 – дійсного типу
Змінна: s – дійсного типу
Змінна: f – типу text;

Початок програми

Пов'язуємо файлову змінну F з файлом Z.dat
Створюємо файл Z.dat для запису інформації
Вводимо координати першої точки
Вводимо координати другої точки
Обчислюємо довжину відрізка
Виводимо отриманий результат на екран
Виводимо отриманий результат у файл Z.dat
Закриваємо файл Z.dat

Кінець програми

Література

1. Т.П.Караванова. Основи алгоритмізації та програмування. – Київ: Форум, 2003
2. Епанешников А. Программирование в среде Turbo Pascal 7.0. – М.: Диалог, 1993
3. Прайс Д. Программирование на языке Паскаль. – М.: Мир, 1987

Confidential



ЗМІСТ

МОВА ПРОГРАМУВАННЯ. СЕРЕДОВИЩЕ ПРОГРАМУВАННЯ.	3
ВВЕДЕННЯ, ВИВЕДЕННЯ ДАНИХ. ВКАЗІВКА ПРИСВОЄННЯ.	5
ВКАЗІВКА РОЗГАЛУЖЕННЯ	18
ВКАЗІВКА ПОВТОРЕННЯ	28
МАСИВИ.	37
ПРОЦЕДУРИ ТА ФУНКЦІЇ.	50
РЯДКОВІ ВЕЛИЧИНИ.	57
ГРАФІЧНІ ВКАЗІВКИ. ПОБУДОВА ГРАФІКІВ ФУНКЦІЙ.	62
РОБОТА З ФАЙЛАМИ	68
ЛІТЕРАТУРА	71

Confidential



Lined writing area consisting of multiple horizontal lines for text entry.

Confidential



A series of horizontal lines for writing, starting from the top of the page and extending downwards.

Confidential

Confidential