

ЧЕРНІГІВСЬКИЙ НАЦІОНАЛЬНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ імені
Т.Г. ШЕВЧЕНКА

ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ

Кафедра технологічної освіти та інформатики

Г.М.НІТЧЕНКО

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ДО ЛАБОРАТОРНИХ РОБІТ З КУРСУ
ОСНОВИ ПРОГРАМУВАННЯ
ДЛЯ СТУДЕНТІВ НАПРЯМУ ПІДГОТОВКИ
6.010103 ТЕХНОЛОГІЧНА ОСВІТА**

Чернігів 2012

ББК 397р 30 – 252.43
УДК 378.147.091.33-027.22:044(075.8)
Н 69

Рецензенти

Дорошенко М.В. кандидат педагогічних наук, доцент кафедри інформатики і обчислювальної математики Дрогобицького державного педагогічного університету ім. І.Франка

Цибко Г.Ю. кандидат педагогічних наук, доцент, завідувач кафедрою інформатики і обчислювальної техніки Чернігівського національного педагогічного університету імені Т.Г.Шевченка

Нітченко Г.М.

Н 69 **Методичні рекомендації до лабораторних робіт з курсу "Основи програмування" для студентів напрямку підготовки 6.010103 «Технологічна освіта» / Г.М. Нітченко. – Чернігів: ЧНПУ імені Т.Г.Шевченка, 2012. – 64 с.**

ББК 397р 30 – 252.43
УДК 378.147.091.33-027.22:044(075.8)

Рекомендовано до друку Вченою радою технологічного факультету
Чернігівського національного педагогічного університету імені Т.Г. Шевченка
(протокол №2 від 21 вересня 2012 р.)

ЛАБОРАТОРНА РОБОТА №1
МОВА ПРОГРАМУВАННЯ ТУРБО ПАСКАЛЬ.
ЛІНІЙНІ ПРОГРАМИ ТА ПРОГРАМИ З РОЗГАЛУЖЕННЯМИ

МЕТА: вивчити основні стандартні математичні процедури і функції TP. Закріпити вивчений матеріал при створенні власних програм з розгалуженнями.

ЗНАТИ: форму запису (повну та скорочену) умовного оператора, принцип виконання умовного оператора.

ВМІТИ: застосовувати умовний оператор при написанні власних програм.

ОБЛАДНАННЯ: технічне забезпечення: ПЕОМ, програмне забезпечення: система програмування Turbo Pascal.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

При необхідності змінити порядок виконання операторів програми в залежності від виконання певних умов, тобто здійснити розгалуження процесу обчислень, використовують умовний оператор і оператор варіанту. При програмуванні розгалуження на дві гілки доцільно використовувати умовний оператор (його також називають оператором розгалуження), на більшу кількість гілок – оператор варіанту.

Умовний оператор застосовується в двох формах – повній і скороченій.

Формат скороченої форми оператора:

if <логічний вираз> *then* <оператор1>;

if (якщо), *then* (то) – службові слова, *оператор1* – довільний оператор мови TP.

Виконання оператора: обчислюється значення *логічного виразу*. Якщо воно дорівнює *TRUE*, виконується *оператор1*, якщо *FALSE* – відбувається перехід до виконання наступного оператора програми. *Оператор1* може бути як простим, так і складеним. В останньому випадку він береться в операторні дужки.

Формат повної форми оператора:

if *логічний вираз* *then* *оператор1* *else* *оператор2*;

Виконання оператора: обчислюється значення *логічного виразу*. Якщо воно дорівнює *TRUE*, виконується *оператор1*, якщо *FALSE* – виконується *оператор2*, після чого відбувається перехід до виконання наступного оператора програми. *Оператор1* і *оператор2* можуть бути як простими, так і складеними.

Перед службовим словом else крапка з комою не ставиться.

Приклад. Визначити, яке число ввів користувач – парне чи не парне.

var n: integer;

begin

write ('Введіть ціле число: ');

readln (n);

if n mod 2 = 0 then

write ('Воно парне.')

else

write ('Воно непарне.');

readln

end.

Прості та складені умови. Висловлювання, яке може бути істинним (правильним) або хибним (неправильним) називається умовою.

Проста умова — це висловлювання, в якому два вирази з'єднані знаком операції відношення.

Складена умова — це висловлювання, в якому дві або більше простих умов з'єднані знаками логічних операцій.

У мові програмування Паскаль реалізовані операції відношення:

> — «більше»;

< — «менше»;

= — «дорівнює»;

<> — «не дорівнює»;

>= — «не менше»;

<= — «не більше»;

логічні операції: not — «ні»; and — «і»; or — «або».

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера.) .
2. По літературі до лабораторної роботи з'ясувати контрольні теоретичні запитання.
3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Ввімкнути комп'ютер.
2. Запустити Turbo Pascal.
3. Виконати відповідні практичні завдання з варіантів (варіант вибирається згідно номеру комп'ютера) для самостійного виконання.

Варіант 1

1. Написати програму знаходження периметру прямокутника.
2. Нехай R – четверта координатна чверть. Знайти $f(x,y)$, якщо

$$f(x, y) = \begin{cases} \frac{x^2 + y^5 \cdot \cos x}{\sqrt{xy}}, & \text{якщо } (x, y) \text{ належать } R; \\ \frac{3x}{\ln((x+y)^4)}, & \text{у протилежному випадку.} \end{cases}$$

Варіант 2

1. Написати програму знаходження значення виразу: $y = |x| + x^4$.
2. Нехай R – четверта координатна чверть. Знайти $f(x,y)$, якщо

$$f(x, y) = \begin{cases} \cos^2 x + 5 \sin x^2 - x^4, & \text{якщо } (x, y) \text{ належать } R; \\ \frac{x \cdot |y| - y^2}{x^2 + y^2}, & \text{у протилежному випадку.} \end{cases}$$

Варіант 3

1. Написати програму знаходження площі прямокутника.
2. Нехай R – третя координатна чверть. Знайти $f(x, y)$, якщо

$$f(x, y) = \begin{cases} \frac{\sqrt{\cos x + \sin y + 15} + |x - y|}{y^2 + x^4}, & \text{якщо } (x, y) \text{ належать } R; \\ \frac{\sqrt{|x \cdot y|} - 9y^2}{x^2 + y^2}, & \text{у протилежному випадку.} \end{cases}$$

Варіант 4

1. Написати програму знаходження значення виразу: $y = |x| + 4x^2 - 7$.
2. Нехай R – перша координатна чверть. Знайти $f(x, y)$, якщо

$$f(x, y) = \begin{cases} \frac{\sin x + \sqrt{\cos^2 x + 10|y|}}{x^3 y^2 + e^{2x}}, & \text{якщо } (x, y) \text{ належать } R; \\ \ln|x| + \frac{14xy}{17(x+y)^2}, & \text{у протилежному випадку.} \end{cases}$$

Варіант 5

1. написати програму знаходження довжини кола, якщо відомий його радіус.
2. Нехай R – четверта координатна чверть. Знайти $f(x, y)$, якщо

$$f(x, y) = \begin{cases} \frac{\sqrt{|\cos x| |\sin y|} - (x - y)^2}{y^2 + e^{-x^4}}, & \text{якщо } (x, y) \text{ належать } R; \\ \cos x - \sin^2(y^3 - 5x), & \text{у протилежному випадку.} \end{cases}$$

Варіант 6

1. Написати програму знаходження значення виразу: $y = 2(3x + 4x^2) - 7x$.
2. Нехай R – третя координатна чверть. Знайти $f(x, y)$, якщо

$$f(x, y) = \begin{cases} \cos^2(x + 5) - \frac{|y^3| - xy^2}{\sqrt{|x \cdot y|} + e^{(x+y)}}, & \text{якщо } (x, y) \text{ належать } R; \\ \frac{|y| - y^2}{\sqrt{|x|} + e^{x^2}}, & \text{у протилежному випадку.} \end{cases}$$

Варіант 7

1. Написати програму знаходження площі круга, якщо відомий його радіус.
2. Нехай R – четверта координатна чверть. Знайти $f(x, y)$, якщо

$$f(x, y) = \begin{cases} \frac{\sqrt{|\cos|x||15 + |x|}}{y^2 + e^{4x}} - 3x^2, & \text{якщо } (x, y) \text{ належать } \mathbb{R}; \\ xy^2 + 8x, & \text{у протилежному випадку.} \end{cases}$$

Варіант 8

1. Написати програму для обчислення площі трапеції, якщо відомі довжини її основ та висоти. $\left(S = h \frac{a+b}{2} \right)$

2. Нехай \mathbb{R} – друга координатна чверть. Знайти $f(x, y)$, якщо

$$f(x, y) = \begin{cases} \cos^3 x - \sin |y| + \frac{2x^3}{5x^2 - 7\sqrt{|y|}}, & \text{якщо } (x, y) \text{ належать } \mathbb{R}; \\ \frac{x^3 - \sqrt{|y|} - y^3}{\sqrt{x^2} - \cos(y)}, & \text{у протилежному випадку.} \end{cases}$$

Варіант 9

1. Написати програму для обчислення об'єму піраміди, якщо відомі її висота та площа основи. $\left(V = \frac{Sh}{3} \right)$

2. Нехай \mathbb{R} – третя координатна чверть. Знайти $f(x, y)$, якщо

$$f(x, y) = \begin{cases} \cos x^3 - \frac{|y| - y^2}{\sqrt{|x|} + e^{x^2}}, & \text{якщо } (x, y) \text{ належать } \mathbb{R}; \\ \frac{\sqrt{|\cos x \cdot \sin y|} + |x - y|}{\sqrt{y^2 + x^4}}, & \text{у протилежному випадку.} \end{cases}$$

Варіант 10

1. Написати програму обчислення об'єму паралелепіпеду.

2. Нехай \mathbb{R} – третя координатна чверть. Знайти функцію $f(x, y)$, якщо

$$f(x, y) = \begin{cases} \frac{\sqrt{|\cos x + \sin y + 15 + |x^2 - y|}}{y^2 + x^4}, & \text{якщо } (x, y) \text{ належать } \mathbb{R}; \\ \frac{x^2 \cdot |y| - y^2}{\sqrt{x^2 + y^2}}, & \text{в протилежному випадку.} \end{cases}$$

4. Показати викладачу виконані завдання на будь-якому носії інформації. Оформити лабораторну роботу згідно звіту до роботи.

5. Захистити роботу.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Вкажіть формат повної та скороченої форм умовного оператора.
2. Як виконується умовний оператор? Наведіть приклади.
3. Як можна впорядкувати три числа за зростанням?
4. Знайти помилку у запису умовного оператора:

- 1). **IF** <умова>
 THEN <серія1>;
 ELSE <серія2>;
 END
- 2). **IF** <умова1>
 THEN IF <умова2>
 BEGIN THEN <серія1> **END**
 BEGIN ELSE <серія2> **END**
- 3). **IF** <умова1> <серія1>;
 THEN <серія1>
 ELSE IF <умова2> <серія2>
- 4). **IF** <умова1>;
 IF <умова2>
 BEGIN IF <умова3>
 THEN <серія1>
 ELSE <серія2>
 END
- 5). **IF** <умова1>
 THEN <умова2>
 BEGIN THEN <серія1>
 ELSE <серія2>
 END
- 6). **IF** <умова1>
 THEN <серія1>
 ELSE IF <умова2>
 <серія2>;

5. Формат запису оператора варіанту.

6. Принцип виконання оператора варіанту. Навести приклади.

7. Знайти помилки у запису фрагментів програм:

- 1). **VAR** A:BOOLEAN;
 CASE A **OF**
 1,3,5: WRITE ('непарне');
 2,4,6: WRITE ('парне');
- 2). **VAR** C:CHAR;
 CASE C **OF**
 A, E, Y: WRITE ('');
 B, D, F, M: WRITE ('');
 ELSE WRITE ('');
 END
- 3). **VAR** A:BOOLEAN; B:REAL;
 CASE SIN(PI/3)+COS(PI/2)-B **OF**
 $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$: A:=TRUE;
 0,1: A:=FALSE
 END

1. Абрамов С.А. и др. Задачи по программированию. – М.: Наука, 1988. – 224 с.
2. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык паскаль. – М.: Наука, 1988. – 320 с.
3. Грогно П. Программирование на языке Паскаль. – М.: Наука, 1982.
4. Йенсен К., Вирт Н. Паскаль: руководство для пользователя и описание языка. – М.: Финансы и статистика, 1982.
5. Рамський Ю.С., Цибко Г.Ю. Основи програмування (мовою Паскаль). Курс лекцій. Лабораторний практикум. – К.: НПУ імені М.П. Драгоманова, – 2004.
6. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989. – 160 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на питання для самостійної підготовки.
2. Текст програми, отриманий після виконання практичного завдання, записати у зошит (завдання виконується згідно свого варіанту).
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №2 ЦИКЛ ІЗ ЛІЧИЛЬНИКОМ. ЦИКЛИ З ПЕРЕДУМОВОЮ ТА ПІСЛЯУМОВОЮ

МЕТА: Ознайомитись з операторами циклу з передумовою, післяумовою та циклом з параметром. Вивчити особливості використання даних операторів.

ЗНАТИ: формат запису циклічних операторів, принцип роботи в програмі.

ВМІТИ: використовувати циклічні оператори при написанні програм, застосовувати циклічні оператори при табулюванні функцій.

ОБЛАДНАННЯ: технічне забезпечення: ПЕОМ, програмне забезпечення: система програмування Turbo Pascal 6.0.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Циклічними програмами називають програми, в яких реалізовано команди циклу.

У Паскалі передбачено три різновиди операторів циклу: цикл із передумовою, цикл з післяумовою, цикл із лічильником (з параметром). Також реалізована робота із вкладеними циклами. Вкладені цикли — циклічні процеси, що допускають укладеність одних циклів в інші.

Цикл із лічильником (з параметром) — це цикл, у якому тіло циклу виконується заздалегідь відому кількість разів.

За умови, коли відома кількість повторень тіла циклу, зручно використовувати оператор циклу з параметром.

Формат циклу:

for <параметр_циклу> := <вираз1> to <вираз2> do <оператор>;

for (для), *to* (до), *do* (виконувати) – службові слова, параметр_циклу (або лічильник циклу) - ім'я змінної дискретного типу, вираз1 і вираз2 – вирази, тип яких співпадає з типом параметра циклу. Оператор – довільний оператор мови ПР. Складений оператор береться в операторні дужки.

Виконання оператора: обчислюється значення *виразу1*, це значення присвоюється параметру циклу. Потім обчислюється значення *виразу2*. Якщо значення параметра не перевищує значення *виразу2*, виконується тіло циклу, до параметра застосовується функція SUCC, отримане значення знову порівнюється зі значенням *виразу2* і т.д. Після того як значення параметра перевищить значення *виразу2*, виконання оператора циклу закінчується.

$Кількість\ повторень\ тіла\ циклу = значення_виразу2 - значення_виразу1 + 1$

Якщо значення *виразу1* більше значення *виразу2*, то тіло циклу не виконується жодного разу.

Різниця між двома сусідніми значеннями параметра циклу називається *кроком циклу*.

Крок циклу з параметром завжди постійний і залежить від типу параметра.

Якщо параметр циклу спадає, використовується модифікований формат:

for <параметр_циклу> := <вираз2> downto <вираз1> do <оператор>;

На відміну від попереднього формату циклу, при виконанні оператора до параметра циклу застосовується функція PRED. Значення *виразу2* повинно бути більше значення *виразу1*.

Приклад. Обчислити суму перших 50 натуральних чисел методом послідовного додавання.

а) $S := 0;$
for $i := 1$ to 50 *do* $S := S + i;$
б) $S := 0;$
for $i := 50$ *downto* 1 *do* $S := S + i;$

Цикл із передумовою (або цикл-«поки») — це цикл, у якому тіло циклу виконується тільки у разі виконання умови, заданої перед тілом циклу. Якщо умова стає невірною, то робота циклу припиняється і керування передається оператору, наступному за оператором циклу.

Формат циклу:

while <логічний вираз> *do* <оператор>;

while (поки), *do* (виконувати) – службові слова, оператор – довільний оператор мови TP, який називається тілом циклу. Складений оператор береться в операторні дужки.

Виконання оператора: обчислюється значення логічного виразу, якщо воно дорівнює *TRUE*, то виконується оператор і відбувається повернення до умови. Процес продовжується доти, поки значення логічного виразу не стане рівним *FALSE*. Після цього виконання циклу закінчується.

Якщо при першому обчисленні значення логічного виразу його значення дорівнює *FALSE*, тіло циклу не виконується жодного разу.

Зауваження. 1. Щоб цикл коли-небудь закінчив роботу, в його тілі обов'язково повинен бути оператор, що впливає на значення логічного виразу. Краще, якщо цей оператор останній у тілі циклу.

2. Логічний вираз має бути коректним, тобто його значення повинно бути визначеним ще до першого виконання тіла циклу.

Приклад. Обчислити суму перших 50 натуральних чисел методом послідовного додавання.

$i := 1; S := 0;$
WHILE $i <= 50$ *DO begin*
 $S := S + i;$
 $i := i + 1;$
end;

Цикл із післяумовою (або цикл-«до») — це цикл, у якому тіло циклу виконується доти, поки умова, задана після тіла циклу, не стане правильною. Якщо умова стає правильною, то робота циклу припиняється й управління передається оператору, наступному за оператором циклу.

Формат циклу:

repeat <оператор> *until* <логічний вираз>;

repeat (повторювати), *until* (до) – службові слова; оператор – довільний оператор мови TP, в тому числі порожній. Складений оператор в операторні дужки не береться.

Виконання оператора: виконується тіло циклу, потім обчислюється значення логічного виразу. Якщо воно дорівнює *FALSE*, то відбувається повернення до виконання тіла циклу, якщо *TRUE* – виконання циклу закінчується.

На відміну від циклу з передумовою, тіло циклу з післяумовою завжди виконується принаймні один раз. Це треба враховувати при виборі оператора для виконання циклічних обчислень.

Для циклу з післяумовою також справедливе зауваження 1.

Приклад. Обчислити суму перших 50 натуральних чисел методом послідовного додавання.

$i := 0; S := 0;$

$REPEAT i := i + 1; S := S + i UNTIL i \geq 50;$

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).

2. По літературі до лабораторної роботи з'ясувати контрольні теоретичні запитання.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Ввімкнути комп'ютер.
2. Запустити Turbo Pascal.
3. Виконати відповідні практичні завдання з варіантів (варіант вибирається згідно номеру комп'ютера) для самостійного виконання.

Варіант №1

1. Обчислити суму перших n членів послідовності:

$$y = 1/2 + 1/4 + 1/16 + 1/64 + \dots$$

2. Написати програму, яка виводить на екран таблицю для визначення потужності для однофазного змінного струму. ω змінюється від -2π до 2π з кроком $\pi/6$.

ω	$P = UI \cos \omega$

Варіант №2

1. Обчислити добуток перших n ($3 < n < 10$) членів послідовності:

$$y = \frac{1}{1^2} \cdot \frac{1}{2^2} \cdot \frac{1}{3^2} \cdot \dots \cdot \frac{1}{n^2}$$

2. Написати програму, яка виводить на екран таблицю для визначення електромагнітної моменту синхронного двигуна. ϕ змінюється від -3π до π з кроком $\pi/6$. $k=5,34$; $\Phi=23,2$.

Φ	$M_{em} = k\Phi I \cos \phi$

Варіант №3

1. Обчислити добуток перших n ($n > 20$) членів послідовності:

$$y = \frac{1}{1+1} + \frac{2}{1+2} + \frac{3}{1+3} + \dots + \frac{n}{1+n}$$

2. Написати програму, яка виводить на екран таблицю для визначення повного опору Z кола з послідовним з'єднанням активного опору, індуктивності та ємності. ω змінюється від -2π до 2π з кроком $\pi/6$.

ω	$Z = \sqrt{R^2 + (\omega L - \frac{1}{\omega C})^2}$

Варіант №4

1. Обчислити суму перших n членів послідовності:
 $y=1+2/4+3/9+4/16+\dots$
2. Написати програму, яка виводить на екран таблицю для визначення повного опору Z для кола з послідовним з'єднанням активного опору та індуктивності. ω змінюється від $-\pi$ до π з кроком $\pi/5$.

ω	$Z = \sqrt{R^2 + \omega^2 L^2}$

Варіант №5

1. Обчислити добуток перших n членів послідовності:
 $y=2/3*3/4*4/5*5/6*6/7*7/8*\dots$
2. Написати програму, яка виводить на екран таблицю для визначення миттєвого значення струму i . ω змінюється від $-\pi$ до π з кроком $\pi/5$.

ω	$i=I_m \sin \omega t$

Варіант №6

1. Обчислити суму перших n ($n>4$) членів послідовності:
 $y=1+1/2^3+1/3^3+\dots+1/n^3$.
2. Написати програму, яка виводить на екран таблицю для визначення миттєвого значення напруги u . ω змінюється від $-2\pi/3$ до $2\pi/3$ з кроком $\pi/6$.

ω	$u=U_m \sin \omega t$

Варіант №7

1. Обчислити суму перших n ($n>100$) членів послідовності:
 $y=\sin 100+\sin 101+\dots+\sin n$.
2. Написати програму, яка виводить на екран таблицю для визначення електрорушійної сили. ω змінюється від $-2\pi/3$ до $2\pi/3$ з кроком $\pi/6$.

ω	$e=E \sin \omega t$

Варіант №8

1. Обчислити суму перших n ($n>1$) членів послідовності:
 $y=\cos 1+\cos 2+\dots+\cos n$.
2. Написати програму, яка виводить на екран таблицю для визначення потужності для однофазного змінного струму. $\omega = \pi/6$. I змінюється від 4,5 до 7,5 з кроком 1/3.

I	$P = UI \cos \omega$

Варіант №9

1. Обчислити суму перших n ($n > 3$) членів послідовності:

$$y = 2/2 + 3/5 + \dots + \frac{n+1}{n^2+1}$$

2. Написати програму, яка виводить на екран таблицю для визначення повного опору Z кола з послідовним з'єднанням активного опору, індуктивності та ємності. $\omega = \pi/3$. R змінюється від 4 до 13 з кроком 1,5.

R	$Z = \sqrt{R^2 + (\omega L - \frac{1}{\omega C})^2}$

Варіант №10

1. Обчислити добуток перших n ($n > 20$) членів послідовності:

$$y = \frac{1}{1+1^2} + \frac{1}{1+2^2} + \frac{1}{1+3^2} + \dots + \frac{1}{1+n^2}$$

2. Написати програму, яка виводить на екран таблицю для визначення електромагнітної моменту синхронного двигуна. $\varphi = \pi/6$. $k=5,34$; $\Phi=23,2$. I змінюється від 1 до 9 з кроком 0,5.

I	$M_{em} = k\Phi I \cos\varphi$

4. Показати викладачу виконані завдання на будь-якому носії інформації.
Оформити лабораторну роботу в зошиті згідно звіту до роботи.
5. Захистити роботу.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

- Який загальний вигляд має оператор циклу з передумовою?
- Що таке табулювання функції? Наведіть приклади.
- Визначити значення змінних S та I після виконання таких операторів:
 - $S:=0$; $I:=0$;
WHILE $I < 5$ **DO** $I:=I+1$; $S:=S+1/I$;
 - $S:=0$; $I:=1$;
WHILE $I > 1$ **DO BEGIN** $S:=S+1/I$; $I:=I-1$ **END**;
 - $S:=0$; $I:=0$;
WHILE $I < 5$ **DO BEGIN** $I:=I+1$; $S:=S+I$; **END**;
 - $S:=4$; $I:=1$;
WHILE $S >= 0$ **DO BEGIN** $S:=S-I$; $I:=I+1$ **END**;
- Замінити в слідуючих фрагментах програм вказівку **while** на вказівку повторення з параметром **for**
 - $I:=0$; $N:=5$;
WHILE $I < N$ **DO BEGIN** $I:=I+1$; <серія> **END**
 - $I:=5$; $P:=15$;
WHILE $I <= P$ **DO BEGIN** <серія> $I:=I+1$ **END**;
 - $I:=1$; $J:=1$; $N:=10$; $K:=10$;
WHILE ($J <= N$) **DO BEGIN** $I:=1$;
WHILE $I <= K$ **DO**
BEGIN <серія> $I:=I+1$
END;

J:=J+1 **END**;

5. Який загальний вигляд має оператор циклу з післяумовою?
6. Чим відрізняються цикли While і Repeat?
7. Знайти помилку в запису циклу Repeat:
 - a). I:=10
BEGIN REPEAT <серія> I:=I+1 **UNTIL** I<10
 - б). I:=10;
REPEAT I=0 <серія> I:=I-1 **UNTIL**
 - в). I:=10
BEGIN REPEAT <серія> I:=I-1 **UNTIL** I>10
8. Визначити значення змінних X та I після виконання таких операторів:
 - a) X:=0; I:=6;
REPEAT X:=X+1; I:=I+1 **UNTIL** I>5;
 - б) X:=0; I:=4;
REPEAT X:=X+1; I:=I+1 **UNTIL** I>5;
 - в) X:=0; I:=1;
REPEAT X:=X+1/I; I:=I-1 **UNTIL** I<=1;
9. Замінити (якщо це можливо) в слідуючих фрагментах програм вказівку **while** на вказівку повторення **repeat**
 - a) I:=0; N:=5;
WHILE I<N **DO BEGIN** I:=I+1; <серія> **END**
 - б) I:=2; P:=8
WHILE I<=P **DO BEGIN** <серія> I:=I+1 **END**;
 - в) I:=1; N:=10;
WHILE I<>N **DO BEGIN** <серія> I:=I+2 **END**
- 10.Що називається циклом у програмуванні? Вкажіть формат оператора циклу з параметром. Що називається кроком циклу?
- 11.Які існують модифікації оператора циклу з параметром. Як вони виконуються?
- 12.Знайти помилку в запису вказівки повторення з параметром FOR:
 - 1) **BEGIN** X:=0 **TO** 200
BEGIN
 <серія>
END
 - 2) **FOR** X=-4 **TO** 100
 <серія>
END
 - 3) **BEGIN**
 FOR X:=0 **TO** 50
 <серія>
 END
 - 4) **FOR** X=20 **TO** 0 **DO**
 BEGIN
 <серія>
 END
 - 5) **FOR** I:=N **TO** 1
 BEGIN

<серія>

END

13. Знайти помилку в запису вказівки повторення з параметром **FOR**:

- 1) **VAR X : REAL;**
FOR X:=0 TO 200 DO
BEGIN
 <серія>
END
- 2) **VAR X : [1..100];**
FOR X:=-4 TO 100 DO
BEGIN
 <серія>
END
- 3) **VAR S : STRING;**
FOR S:='A' TO 'Z' DO
BEGIN
 <серія>
END
- 4) **VAR S : ['A'..'Z'];**
FOR X:=Z DOWNTO A DO
BEGIN
 <серія>
END
- 5) **VAR X : INTEGER;**
FOR X:='A' TO 'Z' DO
BEGIN
 <серія>
END

ЛІТЕРАТУРА

7. Абрамов С.А. и др. Задачи по программированию. – М.: Наука, 1988. – 224 с.
8. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык паскаль. – М.: Наука, 1988. – 320 с.
9. Грогно П. Программирование на языке Паскаль. – М.: Наука, 1982.
10. Йенсен К., Вирт Н. Паскаль: руководство для пользователя и описание языка. – М.: Финансы и статистика, 1982.
11. Рамський Ю.С., Цибко Г.Ю. Основи програмування (мовою Паскаль). Курс лекцій. Лабораторний практикум. – К.: НПУ імені М.П. Драгоманова, – 2004.
12. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989. – 160 с.

ЗВІТ ДО РОБОТИ

4. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на питання для самостійної підготовки.
5. Текст програми, отриманий після виконання практичного завдання,

- записати у зошит (завдання виконується згідно свого варіанту).
6. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №3 МАСИВИ В МОВІ П

МЕТА: Ознайомитись з масивами. Вивчити особливості використання масивів.
Навчитися складати програми для обробки лінійних таблиць.

ЗНАТИ: способи опису масивів, операції, які можна виконувати над ними.

ВМІТИ: використовувати масиви при написанні програм, використовувати цикл з параметром при розв'язуванні задач на опрацювання матриць.

ОБЛАДНАННЯ: технічне забезпечення: ПЕОМ, програмне забезпечення: система програмування Turbo Pascal 6.0.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Структури даних. Масиви

У різних розділах математики та інших наук дані, що мають вигляд інформації, заданої як послідовність рядків і стовпчиків, називають порізно: матриці - у вищій алгебрі, таблиці - у розрахункових задачах, масиви - у програмуванні.

Масив — це структурована сукупність фіксованої кількості елементів одного типу, доступ до яких здійснюється за допомогою індексів.

Індексом називається порядковий номер елемента масиву.

Тип елементів масиву називається *базовим типом масиву*.

Формат опису масиву у розділі опису типів:

1. *Type* <ім'я_масиву> = *Array*[список індексів] *of* <базовий_тип>;

Var A:<ім'я_масиву>;

2. *Var* A: *Array*[список індексів] *of* <базовий_тип>;

Array (масив), *of* (з) – службові слова, *ім'я_масиву* – ідентифікатор, тип індексів – діапазон (підмножина значень простого дискретного типу).

Приклад.

Type Dim3 = *Array*[1..3] *of* *Real*;

Var W, V: Dim3;

Змінна V є структурою з трьох дійсних чисел: V[1], V[2], V[3]. Числа 1,2,3 – індекси.

Якщо базовим типом масиву є інший масив, утворюється структура, яка називається багатовимірним масивом.

Приклад.

Type Vector = *Array* [1..4] *of* *Integer*;

Matrix = *Array* [1..4] *of* *Vector*;

Var Matr: *Matrix*;

Таку ж структуру можна отримати, використовуючи іншу форму запису:

Type Matrix = *Array* [1..4,1..4] *of* *Integer*;

Var Matr: *Matrix*;

або

Var Matr: *Array* [1..4,1..4] *of* *Integer*;

Останній запис показує, що тип можна задавати безпосередньо при описі

змінних.

Елементи масиву Matr: Matr[1,1], Matr[2,3], Matr[4,4] (всього 16 елементів)

Для звернення до окремого елемента масиву треба вказати ім'я масиву і в квадратних дужках індекси елемента. Елемент одновимірного масиву має один індекс, багатовимірного – стільки індексів, яка вимірність масиву.

Двовимірні масиви широко використовуються для подання матриць.

В загальному випадку матриця записується так:

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

У TP така матриця є масивом типу *Array[1..m,1..n] of Real*;

Якщо кількість рядків матриці дорівнює кількості стовпців, матриця називається квадратною. Головна діагональ квадратної матриці проходить з верхнього лівого кута до правого нижнього. Побічна діагональ – з нижнього лівого кута до правого верхнього.

Приклад. Дано масив з 25 дійсних чисел. Визначити максимальний та мінімальний елементи масиву та поміняти їх місцями. Отриманий результат вивести на екран.

```
PROGRAM MAS;
```

```
VAR
```

```
  M: ARRAY [1.. 25] OF REAL;
```

```
  MAX, MIN, MN: REAL;
```

```
  I, K, N: INTEGER;
```

```
BEGIN
```

```
  {Введення елементів масиву}
```

```
  FOR I: = 1 TO 25 DO READ ( M [I] );
```

```
  {Змінним MAX і MIN надаємо значення першого елемента масиву}
```

```
  MAX: = M [1]; K: = 1;
```

```
  MIN: = M [1]; N: = 1;
```

```
  {Визначаємо максимальний та мінімальний елементи масиву та їх індекси}
```

```
  FOR I:= 2 TO 25 DO
```

```
    BEGIN
```

```
      IF M [I] > MAX THEN BEGIN MAX: = M [I]; K: = I; END;
```

```
      IF M [I] < MIN THEN BEGIN MIN: = M [I]; N: = I; END;
```

```
    END;
```

```
  {Міняємо місцями максимальний та мінімальний елементи}
```

$MN := M [K];$
 $M [K] := M [N];$
 $M [N] := MN;$

{Виведення отриманого масиву на екран}

FOR I := 1 TO 25 DO WRITELN (M[I]);
READLN;
END.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. З'ясувати які існують способи впорядкування масивів і пояснити як вони працюють (метод прямого вибору, вставленням, методом «Бульбашки», методом Шелла).
2. З'ясувати контрольні-теоретичні питання до лабораторної роботи.
3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Ввімкнути комп'ютер.
2. Запустити Turbo Pascal.
3. Виконати відповідні практичні завдання з варіантів (варіант вибирається згідно номеру комп'ютера) для самостійного виконання.

Варіант №1

1. Задано лінійну таблицю $A[1:15]$, елементи якої рівні нулю чи одиниці. Скласти програму для заміни елементів, рівних нулю – одиницею, а елементів, рівних одиниці – нулем.
2. Написати програму, яка знаходить максимальний та мінімальний за абсолютною величиною (модулем) елементи у лінійній таблиці $A[1:15]$ та переставляє їх місцями.

Варіант №2

1. Задано лінійну таблицю $A [M:N]$, яка складається з дійсних чисел. Скласти програму знаходження суми елементів, що стоять на парних місцях таблиці і більші за число 9.
2. Відомі результати екзамену 22 студентів однієї з груп по програмуванню. Визначити, скільки студентів склали екзамен на «добре» та «відмінно».

Варіант №3

1. Задано лінійну таблицю $A[1:15]$, яка складається з цілих чисел. Скласти програму знаходження і виведення на екран всіх парних чисел в даній таблиці.
2. Написати програму, яка знаходить максимальний за абсолютною величиною (модулем) елемент у лінійній таблиці $A[1:15]$.

Варіант №4

1. Написати програму знаходження середнього арифметичного всіх від'ємних елементів масиву.
2. Задано лінійну таблицю $A[1:15]$, яка складається з додатних цілих чисел.

Скласти програму для запису до лінійної таблиці $X[1:k]$ тільки тих чисел, при діленні яких на 5 і на 6 отримується одна і та ж ціла частина.

Варіант №5

1. В одновимірному масиві, який складається з n дійсних елементів, знайти номер максимального за модулем елемента.
2. Задано лінійну таблицю $A[1:15]$, яка складається з дійсних чисел. Знайти суму абсолютних значень (модулей) всіх від'ємних елементів.

Варіант №6

1. Задано лінійну таблицю $A[1:15]$ дійсних чисел. Скласти програму знаходження середнього арифметичного всіх додатних елементів.
2. Дано дійсні числа $a_{1990}, a_{1991}, \dots, a_{2004}$ – кількість опадів (у мм), що випали у місті за останні 15 років. Обчислити середню кількість опадів за цей період і щорічне відхилення від середнього значення.

Варіант №7

1. Скласти програму заповнення і виведення на екран лінійної таблиці $A[1:15]$ таким чином: елементи з непарними номерами рівні числу 2, а елементи з парними номерами рівні числу 5.
2. Написати програму, яка знаходить найбільший з елементів масиву, які мають непарні індекси.

Варіант №8

1. Задано лінійні таблиці $A[1:M]$ і $B[1:N]$. Скласти програму заповнення таблиці C , до якої спочатку заносяться всі елементи таблиці A , потім всі елементи таблиці B .
2. Написати програму визначення кількості елементів лінійної таблиці $A[1:15]$, абсолютні значення яких належать до інтервалу $(3;14)$.

Варіант №9

1. Скласти програму для перерозташування елементів лінійної таблиці $A[1:N]$ в зворотному до заданого порядку. Елементи таблиці задано з клавіатури.
2. Написати програму, яка знаходить найбільший з елементів масиву, які мають парні індекси.

Варіант №10

1. Написати програму знаходження добутку від'ємних елементів лінійного масиву $A[1..10]$.
2. Задано лінійну таблицю $A[1:15]$, яка складається з цілих чисел. Скласти програму визначення, скільки разів число п'ять зустрічається серед елементів, що стоять на непарних місцях таблиці.

3. Показати викладачу виконані завдання на будь-якому носії інформації. Оформити роботу згідно звіту до роботи.

4. Захистити роботу.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Як відбувається впорядкування масиву методом прямого вибору. Навести приклади.
2. Як відбувається впорядкування масиву методом «бульбашки». Навести приклади.

3. Як відбувається впорядкування масиву методом Шелла. Навести приклади.
4. Що називається масивом?
5. Що таке базовий тип масиву, індекс масиву?
6. Як описуються масиви у програмі на ПР?
7. Що таке вимірність масиву? Як описуються багатовимірні масиви?
8. Як можна організувати введення елементів масиву?
9. Як виводяться елементи масиву?
10. Як впорядкувати елементи таблиці методом „бульбашки”?
11. За правилами мови Pascal описати лінійні таблиці, що містять наступну інформацію:
 - 1) річні оцінки учнів з математики, якщо в класі не більше 40 учнів;
 - 2) кількість виготовлених за день деталей на кожному з 30 верстатів;
 - 3) назви столиць країн Європи;
 - 4) назви підручників, які Ви використовуєте.
12. Знайдіть логічну помилку в запису вказівок про виконання операції над табличними величинами:
 - 1) I:=1; N:=10;
WHILE I<=N **DO** A[I]:=A[I]+1;
 - 2) I:=1; J:=1;
WHILE I<=100 **DO**
 WHILE J<=200 **DO**
 A[I,J]:=2*A[I,J];
 J:=J-1;
 - 3) N:=-5; I:=N;
WHILE I<>1 **DO**
 BEGIN
 A[I]:=0;
 I:=I+1
 END;
 - 4) I:=1; N:=-5;
WHILE I>N **DO**
 IF B[I,4]>0 **THEN** B-I,4]:=1 **ELSE** I:=I+1;
 - 5) I:=K; Y:=FALSE;
WHILE (I<K) **AND** Y **DO**
 BEGIN
 IF A1[2,I]='8' **THEN** Y:=TRUE **ELSE** I:=I+1
 END;
 - 6) **WHILE** I<1 **DO BEGIN IF** A[K]<0 **THEN** A[K]:=ABS(A[K])
END;
13. Записати за правилами мови Pascal фрагменти програм, за якими виконуються наступні дії:
 - 1) перебираються всі елементи таблиці, опис якої за правилами мови Pascal має вигляд:
VAR LETER:ARRAY [1..100] **OF** CHAR
і кожному, який співпадає з літерною константою ‘A’, надається значення ‘B’.
 - 2) перебираються всі елементи таблиці, опис якої за правилами мови

Pascal має вигляд:

```
CONST K=100; P:=10;  
VAR C:ARRAY [1..P,1..K] OF REAL
```

і підраховується кількість додатних елементів.

3) перебираються всі елементи таблиці, опис якої за правилами мови

Pascal має вигляд:

```
CONST N=100; VAR C:ARRAY [1..N] OF CHAR
```

доти, поки не зустрінеться елемент, значення якого співпадає з ' ', при цьому запам'ятовується номер цього елемента.

ЛІТЕРАТУРА

1. Абрамов С.А. и др. Задачи по программированию. – М.: Наука, 1988. – 224 с.
2. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык паскаль. – М.: Наука, 1988. – 320 с.
3. Грогоно П. Программирование на языке Паскаль. – М.: Наука, 1982.
4. Йенсен К., Вирт Н. Паскаль: руководство для пользователя и описание языка. – М.: Финансы и статистика, 1982.
5. Рамський Ю.С., Цибко Г.Ю. Основи програмування (мовою Паскаль). Курс лекцій. Лабораторний практикум. – К.: НПУ імені М.П. Драгоманова, – 2004.
6. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989. – 160 с.
7. Морзе Н.В. Практичні роботи з основ алгоритмізації та програмування мовами BASIC та PASCAL.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на питання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту (варіант вибирається згідно номера свого комп'ютера).
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №4 ПРОЦЕДУРИ ТА ФУНКЦІЇ В МОВІ TP

МЕТА: Ознайомитись з поняттями «процедура» і «функція» в мові програмування TP Pascal, закріпити практичні навички роботи з системою TP Pascal на прикладі реалізації алгоритмів за допомогою процедур і функцій.

ЗНАТИ: поняття «процедура» та «функція», формати їх опису та виклику, класифікацію параметрів підпрограм, способи передавання параметрів, правила локалізації, поняття рекурсії та побічного ефекту.

ВМІТИ: розв'язувати задачі з максимальним використанням підпрограм.

ОБЛАДНАННЯ: технічне забезпечення: ПЕОМ, програмне забезпечення: система програмування Turbo Pascal 6.0.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Підпрограмою називається поіменована логічно завершена група операторів мови, яку можна викликати для виконання за іменем довільну кількість разів у різних місцях програми.

Програма, яка використовує підпрограми, по відношенню до них називається *головною* або *основною*.

У мові Паскаль існує два види підпрограм: процедура (procedure) і функція (function). Всі процедури і функції мови Паскаль поділяються на дві групи: вбудовані і визначені користувачем.

Вбудовані (стандартні) процедури і функції є частиною мови і можуть викликатися за іменем без попереднього опису (наприклад, Read, Write, Sin, Cos, Pred, Succ тощо).

Процедури і функції користувача обов'язково описуються в програмі у розділі визначення процедур і функцій, який слідує за розділом опису змінних.

При введенні у програму підпрограм виникає поділ даних і їх типів на глобальні і локальні.

Глобальні змінні (константи, типи) – ті, які описані в головній програмі. Ними можна користуватися і в головній програмі, і в підпрограмі.

Локальні змінні (константи, типи) – ті, які описані всередині підпрограми: або у списку параметрів (тільки змінні), або у розділах Const, Type, Var тіла підпрограми. За межами підпрограми локальні об'єкти недоступні.

Процедура – це іменована група операторів, яка реалізує певну частину загальної задачі і викликається за іменем з будь-якої позиції у головній програмі.

Процедури використовуються у випадках, коли в підпрограмі необхідно отримати декілька результатів.

Параметри використовуються для обміну інформацією між процедурою і основною програмою.

Параметри, які використовуються при описі процедури, називаються *формальними*.

Параметри, які використовуються при виклику процедури, називаються *фактичними*.

Кількість, тип і порядок слідування фактичних параметрів повинні бути

такими, як у формальних параметрів. Параметри можуть мати будь-який тип, але визначений заздалегідь. Процедура може взагалі не мати параметрів.

Формат опису процедури:

Procedure ім'я_процедури [(список формальних параметрів)]; – заголовок процедури

розділи описів;
Begin
оператори
End; } *тіло процедури*

Procedure (процедура) – службове слово, *ім'я_процедури* – ідентифікатор, *список формальних параметрів* = *форм_пар1: Тип1, форм_пар2: Тип2, ...*
розділ описів – опис локальних об'єктів процедури (за синтаксисом співпадає з розділом описів програми на TP).

Тіло процедури за структурою аналогічне програмі на TP, *ім'я_процедури* – ідентифікатор, унікальний у межах програми.

Для звернення до процедури в головній програмі використовується оператор виклику процедури.

Формат оператора виклику процедури:

Ім'я_процедури[(список фактичних параметрів)];

список фактичних параметрів = *факт_пар1, факт_пар2, ...*

У мові Паскаль є два способи передавання параметрів: за значенням і за посиланням.

Параметри, які передаються за значенням, називаються *параметрами-значеннями*, параметри, які передаються за посиланням, називаються *параметрами-змінними*.

Параметри-значення передаються з головної програми в процедуру, але не повертаються з неї. Фактичний параметр, відповідний формальному параметру-значенню, може бути будь-яким виразом того ж типу. Характерна риса параметрів-значень – зміна формальних параметрів не викликає зміни фактичних параметрів. Тому параметри-значення використовуються в процедурах для подання їх аргументів (вхідних даних).

Параметри-змінні і передаються в процедуру, і повертаються з неї в основну програму. Фактичний параметр, відповідний параметру-змінній, може бути тільки іменем змінної. Характерна риса параметрів-змінних – будь-яка зміна формального параметра означає зміну фактичного параметра. Параметри-змінні використовуються в процедурах для подання їх результатів, для того щоб результати могли бути передані і використані в основній програмі. У списку формальних параметрів перед іменами параметрів-змінних ставиться службове слово *Var*.

Правильно написана підпрограма спілкується з основною програмою тільки через параметри.

Приклад 1. Процедура без параметрів, яка виводить на екран рядок з 60 зірочок.

```
procedure pr;  
var i : integer ;  
begin
```



```

for i :=1 to 60 do write (' * ');
writeln;
end;

```

Приклад 2. Процедура с параметрами. Дано три різних масиви цілих чисел (розмір кожного не перевищує 15). В кожному масиві знайти суму елементів та їх середнє арифметичне значення.

```

Program proc;
var i , n , sum: integer;
    sr : real;
procedure work (r:integer; var s:integer; var s1:real);      {процедура work}
    var mas : array [1..15] of integer ;                  { опис масиву mas}
        j : integer;
begin
    s:=0;
    for j:=1 to r do begin                                {введення елементів масиву
mas}
        write(' Vvedit element - ', j, ': ');
        read (mas[j]);
        s:=s+mas [j];
    end;
    s1:=s/r;
end;
begin                                                    { головна програма}
    for i:=1 to 3 do begin
        write ('Vvedit rozmir ',i, ' masivu: ');
        readln(n);
        work (n, sum, sr);                               {виклик процедури work}
        writeln ('Suma elementiv = ',sum);
        writeln ('Seredne arifmetichne = ',sr:4:1);
    end;
    readln;
end.

```

Функція – це іменована група операторів, яка реалізує певну частину загальної задачі і має своїм результатом одне значення простого типу. Ім'я функції використовується у виразах як операнд.

Формат опису функції:

Function Ім'я_функції [(список форм. пар-рів)]: тип_результату; –
заголово
к
функції

розділи описів; }
Begin } тіло функції
оператори }
End;

Function (функція) – службове слово, *тип_результату* – ім'я простого типу,

тіло функції за структурою аналогічне тілу процедури і програми на ПР. Ім'я_функції – ідентифікатор, унікальний у межах програми.

В розділі операторів повинен знаходитись принаймні один оператор присвоювання, в лівій частині якого стоїть ім'я функції, а в правій – певний вираз. Значення цього виразу функція повертає в основну програму. Якщо таких присвоювань декілька, результатом функції є значення останнього оператора присвоювання.

Для звернення до функції у головній програмі використовується ім'я функції з указаним в дужках списком фактичних параметрів:

Ім'я_функції [(список факт. пар-рів)];

Співвідношення між формальними і фактичними параметрами у процедур і функцій аналогічне. У заголовку функції можуть використовуватись як параметри-значення, так і параметри-змінні.

Приклад. Обчислити значення виразу
 $f(x)=x^2+x+3+e^{x^2+x+3}+\sin^2(x^2+x+3) \quad x=1,2,\dots,10.$

```
Program Viraz;  
Var x: Byte; f: Real;  
Function y(x: Real): Real;  
Begin  
  y:=x*x+x+3;  
End;  
Begin  
  for x:=1 to 10 do  
    Begin  
      f:=y(x)+exp(y(x))+sqr(sin(y(x)));  
      Writeln(f)  
    End;  
  End.  
End.
```

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера.) .
2. По літературі до лабораторної роботи вивчити контрольні-теоретичні питання до лабораторної роботи.
3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Ввімкнути комп'ютер.
2. Запустити Turbo Pascal.
3. Виконати відповідні практичні завдання з варіантів (варіант вибирається згідно номеру комп'ютера) для самостійного виконання.

Варіант №1

1. Дано дійсні числа s, t . Отримати $h(s,t) + \max(h(s-t, st), h(s-t, s+t)) + h(1,1)$, де $h(a,b) = a/(1+b^2) + b/(1+a^2)$
2. Дано три дійсні лінійні матриці A, B та C розмірністю $[1 \times n]$. Визначити, в якій з них найбільший елемент зустрічається раніше. (Скласти підпрограми для введення матриці, знаходження найбільшого елемента)

Варіант №2

1. Дано дійсні числа s, t . Отримати $f(t, -2s, 1.17) + f(2.2, t, s-t)$,

$$f(a, b, c) = \frac{2a - b - \sin(c)}{5 + c}$$

2. Дано дві дійсні лінійні матриці A та B розмірністю $[1 \times m]$. Визначити, в якій з них сума елементів найбільша. (Скласти підпрограми для введення матриці, знаходження суми елементів)

Варіант №3

1. Дано дійсні числа s, t . Отримати $g(1.2, s) + g(t, s) - g(2s-1, st)$, де $\frac{a^2 + b^2}{a^2 + 3ab}$.
2. Дано три лінійні матриці A, B, C розмірністю 10. Створити підпрограму введення елементів цих таблиць та визначення максимальних елементів у кожній з них. (Скласти підпрограми для введення матриці, визначення максимальних елементів)

Варіант №4

1. Дано дійсне число h . Отримати $\frac{2.8(t(h)+2) - t(h+2)}{6 - t(h^2+3)}$, де $t(y) = \sum_{k=1}^{20} \frac{y}{k^2}$.
2. Дано дві дійсні лінійні матриці A і B розмірністю 10. Визначити, в якій з них мінімальний елемент зустрічається раніше. (Скласти підпрограми для введення матриці, знаходження мінімальних елементів)

Варіант №5

1. Дано дійсне число y . Отримати $\frac{1.7t(0.25) + 2t(1+y)}{6 - t(y^2-1)}$, де $t(y) = \sum_{k=1}^{10} \frac{y}{2k}$.
2. Дано три дійсні лінійні таблиці A, B та C розмірністю 10. Визначити, в якій з них кількість максимальних елементів найбільша. (Скласти підпрограми для введення матриці, знаходження максимальних елементів, підрахунок кількості максимальних елементів)

Варіант №6

1. Дано дійсні числа a, b, c . Отримати $\frac{\min(c, a+b) + \min(a, b+c)}{1 + \min(a+bc, 15)}$.
2. Дано три лінійні матриці A, B, C розмірністю 10. Створити підпрограму введення елементів цих таблиць та визначення кількості парних елементів у кожній матриці. (Скласти підпрограми для введення матриці, знаходження кількості парних чисел)

Варіант №7

1. Дано дійсні числа k, p . Отримати $f(1-k, 2p^2, 3.58) + f(k+p, p^3, k-2p)$, де

$$f(a, b, c) = \frac{2a - b - \sin(c)}{5 + c}$$

2. Дано три лінійні матриці A, B, C розмірністю 10. Створити підпрограму введення елементів цих таблиць та визначення мінімальних елементів у кожній з них. (Скласти підпрограми для введення матриці, визначення

мінімальних елементів)

Варіант №8

1. Дано дійсні числа r, h . Отримати $g(3.5, r^2) + g(r^2+h, 2h) - g(3h-8, rh+1)$,

$$\text{де } g(a, b) = \frac{a^3 + 3b^2}{b^2 + 5b - a}$$

2. Дано три лінійні матриці A, B, C розмірністю 10. Створити підпрограму введення елементів цих таблиць та визначення середньої арифметичної суми для кожної матриці. (Скласти підпрограми для введення матриці, визначення середнього арифметичного)

Варіант №9

1. Дано дійсне число y . Отримати $\frac{1.7f(0.25) + 2f(1+y)}{6 - f(y^2 - 1)}$, де $f(x) = x + x^3 + x^5 + x^7$.

2. Дано три лінійні матриці A, B розмірністю 10. Створити підпрограму введення елементів цих таблиць та знаходження суми всіх елементів. (Скласти підпрограми для введення матриці, визначення суми елементів матриці)

Варіант №10

1. Дано дійсні числа q, h . Отримати $g(5.4, q^3) - g(h^2+q, 3hq) - g(0.3h+9, qh+1)$,

$$\text{де } g(a, b) = \frac{a^2 + 8b}{a^2 + 5a - b}$$

2. Дано три лінійні матриці A, B розмірністю 10. Створити підпрограму введення елементів цих таблиць та знаходження добутку всіх елементів. (Скласти підпрограми для введення матриці, визначення добутку елементів матриці)

4. Показати викладачу виконані завдання на будь-якому носії інформації. Оформити лабораторну роботу згідно звіту до роботи.

5. Захистити роботу.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Для чого потрібні у програмі процедури та функції?
2. Що таке функція? Як вона описується?
2. Що таке процедура? Як вона описується?
3. Яка різниця між процедурою та функцією?
4. Що називається формальними та фактичними параметрами?
5. Що таке глобальні та локальні змінні?
6. Що називається параметрами-значення, параметрами змінними?
7. Вкажіть правила локалізації TP.
8. Що таке рекурсія? Наведіть приклади використання рекурсії.
9. Що таке побічний ефект? Наведіть приклади.
10. Чому при звертанні до процедури аргумент, який передається параметру-змінній, може бути тільки змінною і ніколи константою або виразом?

ЛІТЕРАТУРА

13. Абрамов С.А. и др. Задачи по программированию. – М.: Наука, 1988. –

224 с.

- 14.Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык паскаль. – М.: Наука, 1988. – 320 с.
- 15.Грогоно П. Программирование на языке Паскаль. – М.: Наука, 1982.
- 16.Йенсен К., Вирт Н. Паскаль: руководство для пользователя и описание языка. – М.: Финансы и статистика, 1982.
- 17.Рамський Ю.С., Цибко Г.Ю. Основи програмування (мовою Паскаль).Курс лекцій. Лабораторний практикум. – К.: НПУ імені М.П. Драгоманова, – 2004.
- 18.Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989. – 160 с.

ЗВІТ ДО РОБОТИ

7. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на питання для самостійної підготовки.
8. Виконати завдання згідно свого варіанту (варіант вибирається згідно номера свого комп'ютера).
9. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №5 РЯДКОВІ ВЕЛИЧИНИ В МОВІ ТР

МЕТА: Ознайомитись з можливостями мови Турбо Паскаль (ТР) в опрацюванні символічних і рядкових величин.

ЗНАТИ: стандартні процедури і функції опрацювання символів і рядків у ТР.

ВМІТИ: застосовувати вивчений матеріал при створенні власних програм опрацювання текстової інформації.

ОБЛАДНАННЯ: технічне забезпечення: ПЕОМ, програмне забезпечення: система програмування Turbo Pascal 6.0.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Мова ТР підтримує стандартний символічний тип Char, і, крім того, рядки, які описуються типом String або String[n].

Символ – це один знак. Будь-який – буква, цифра, арифметичний знак або пробіл, розділовий знак або підкреслення, а також спеціальні символи.

У практиці часто доводиться обробляти окремі символи. Змінним символічного типу Char можна привласнювати певні значення. Символічні величини використовуються для представлення як рядків символів, так і окремих символів.

Дані символічних величин представляються двійковими кодами.

Оскільки для комп'ютера символ – число, то символічні величини можна порівнювати між собою. При цьому більшим буде той символ, число якого в таблиці символів більше. Всі наявні символи пронумеровані від 0 до 255.

Операції над символами

Символи можна лише присвоювати і порівнювати один з одним. При порівнянні символів вони вважаються рівними, якщо рівні між собою їх ASCII-коди; і один символ більше за інший, якщо має більший ASCII-код. Операції порівняння записуються традиційним способом: <, >, <=, >=, =, <>.

При роботі із символічними змінними часто використовують функції Chr і Ord.

Chr(X:Byte):Char – приймає як параметр число, і повертає символ, що відповідає цьому числу в таблиці ANSI-II.

Приклад.

Chr(97) = 'a' (символ 'a')

Chr(0) = (нульовий символ)

Chr(32) = ' ' (пропуск)

Символи з кодами від 1 до 31 (управляючі), можуть бути подані їх “клавіатурними” позначеннями – значком “^” і літерою алфавіту з тим самим номером (для діапазону кодів 1..26) або службовим знаком (для діапазону 27..31):

Приклад.

^A = #1 = Chr(1) – код 1,

^B = #2 = Chr(2) – код 2,

^ = #26 = Chr(26) – код 26,

^[= #27 = Chr(27) – код 27,

^_ = #31 = Chr(31) – код 31.

Ord(C:Char):Byte – приймає як параметр символ, і повертає число, під яким цей символ зберігається в таблиці ANSI-II.

Приклад.

$Ord('a') = 97$

$Ord('') = 0$

$Ord(' ') = 32$

$UpCase(C:Char):Char$ – переводить символи 'a'..'z' у верхній регістр 'A'..'Z'

$Pred(C:Char):Char$ – повертає символ, що передує C.

$Succ(C:Char):Char$ – повертає символ, наступний за C.

Величиною рядкового типу (*strings*) називається послідовність символів, укладена в одинарні лапки. Цей тип даних є стандартним для мови Паскаль, хоча він є структурованим. Величини можуть бути як константами, так і змінними. При завданні змінних символного типу можна вказати кількість символів у цій величині.

Для опису даних рядкового типу використовується ідентифікатор *String*, за яким необов'язково слідує взяті в квадратні дужки значення максимально допустимої довжини рядка даного типу. Кількість символів у рядку (довжина рядка) може динамічно змінюватися від 0 до 255. Змінну рядкового типу можна визначити через опис типу в розділі опису типів або безпосередньо в розділі опису змінних. Рядкові дані можуть використовуватись у програмі також як константи.

Формат опису:

Type

$\langle \text{ім'я типу}1 \rangle = \text{String}[\text{максимальна довжина рядка}];$

$\langle \text{ім'я типу}2 \rangle = \text{String};$

Var

$\langle \text{ім'я змінної}1 \rangle : \langle \text{ім'я типу}1 \rangle;$

$\langle \text{ім'я змінної}2 \rangle : \text{String} [\text{максимальна довжина рядка}];$

$\langle \text{ім'я змінної}3 \rangle : \text{String};$

Приклад.

Const

$\text{Address} = \text{'вул.Короленка, 5'};$

Type

$\text{Str20} = \text{String}[20];$

$\text{Strmax} = \text{String}[255];$

$\text{St} = \text{String};$

Var

$\text{A: Strmax; B: St20; C: St;}$

$\text{D: String}[30];$

Рядки, що мають різні довжини, сумісні між собою в операторах присвоювання і порівняння, але при передаванні в процедури і функції треба стежити за узгодженістю типів параметрів.

Рядки можна розглядати як масиви символів. Будь-який символ рядка можна отримати за його номером, починаючи з 1.

Приклад.

```

Var
  i : Byte;
  S : String[15];
Begin
  S := 'Масив символів.';
  for i:=1 to 15 do Writeln(S[i]);           {друкування рядка S посимвольно}
      ReadLn                                 {пауза до натиснення Enter}
End.

```

Окремий символ сумісний за типом зі значенням типу Char.

На початку роботи програми рядкові змінні містять "сміття", тому завжди рекомендується перед використанням рядкових змінних надавати їм початкові значення, наприклад, порожній рядок ''. Для введення значень рядкових змінних використовується оператор ReadLn, а не Read. Одним оператором можна ввести тільки один рядок.

Операції над рядками.

Рядки можна порівнювати, зливати і присвоювати. Злиття рядків записується як їх сума.

Приклад

```

Var
  S1, S2, S3 :String;
Begin
  S1 := 'Всім ';
  S2 := 'привіт';
  S3 := S1 + S2;           {S3 = 'Всім привіт'}
  S3 := S3 + '!';        {S3 = 'Всім привіт!'}
End.

```

Порівняння рядків відбувається посимвольно, починаючи з першого символу в рядку. Рядки вважаються рівними, якщо вони мають однакову довжину і посимвольно еквівалентні. Якщо при посимвольному порівнянні виявиться, що один символ більше за інший (його код більший), то рядок, що його містить, також вважається більше. Решти рядків і їх довжини не грають ролі. Будь-який рядок більше "порожнього місця".

Приклад.

```

'abcd' = 'abcd'   (TRUE),
'abcd' <> 'abcde' (TRUE),
'abcd' > 'abcD'   (м.я. 'd' > 'D'),
'abcd' > 'abc'    (м.я. 'd' > ''),
'aBcd' < 'ab'     (м.я. 'B' < 'b'),
'' > ''          (м.я. #32 > '').

```

Процедури та функції для роботи з рядками

Length(S : String) : Byte – видає довжину рядка S.

Concat(S1, S2, ..., Sn) : String – повертає конкатенацію або злиття рядків S1,S2,...,Sn. Аналогічно діє операція "+":

```

S3 :=Concat(S1,S2);           {аналогічне до S3:=S1 +
S2}

```

Copy(S : String; Start, Len : Integer) : String – повертає підрядок довжиною

Len, який починається з позиції Start рядка S. Якщо Start більше всієї довжини рядка S, то функція повертає порожній рядок. Якщо Len більше, ніж кількість символів від Start до кінця рядка S, то повертається залишок рядка S від Start до кінця.

а) $SCopy := Copy('ABC***123', 4, 3);$ $\{SCopy = '***'\}$
 б) $SCopy := Copy('ABC', 4, 3);$ $\{SCopy = ''\}$
 в) $SCopy := Copy('ABC***123', 4, 11);$ $\{SCopy = '***123'\}$

$Delete(Var S : String; Start, Len : Integer)$ – вилучає з S підрядок довжиною Len, який починається з позиції Start рядка S. Після вилучення підрядка його залишки склеюються. Якщо Start=0 або перевищує довжину рядка S, або Len=0, то рядок не змінюється. Якщо Len більше, ніж залишок рядка, вилучається підрядок від Start і до кінця S.

$S := 'РЯДОК'; Delete(S, 2, 2)$ $\{S = 'ПОК'\}$

$Insert(Var S: String; SubS: String; Start: Integer)$ – вставляє в S підрядок SubS, починаючи з позиції Start. Якщо змінений рядок S виявляється надто довгим, то він автоматично скорочується з правого боку до оголошеної довжини S.

$S := 'Ранок-вечір'; Insert('день-', S, 7);$ $\{менер S = 'Ранок-день-вечір'\}$

$Pos(SubS, S : String) : Byte$ – Шукає входження підрядка SubS в рядок S і повертає номер першого символу SubS в S або 0, якщо S не містить SubS. Недоліком даної функції є те, що вона повертає найближчу стартову позицію SubS в S від початку рядка, ігноруючи наступні, якщо вони є. Так, виклик

$P := Pos('noo', 'Boonoonoonooos');$ $\{завершує роботу, повернувши значення 4, хоча є ще 7 і 10\}$

$Str(X [: Width [: dec]]; Var S : String)$ – перетворює числове значення X в рядкове S. X може бути змінною або значенням цілого або дійсного типу. Можна задавати поля формату, вказавши ширину поля для числа і кількість десяткових знаків. Якщо число має менше знаків, ніж дано в полі, воно вирівнюється по правому краю, порожнє місце заповнюється пропусками. Якщо задати поле Width від'ємним, вирівнювання здійснюватиметься по лівому краю, а залишки стираються. Якщо формат з крапкою обмежує кількість знаків у дробовій частині, то вона буде округлена при перетворенні. Значення самого числа при цьому не зміниться.

$Str(6.66:8:2, S);$ $\{S = ' 6.66'\}$
 $Str(6.66:-8:2, S);$ $\{S = '6.66'\}$
 $Str(6.66:8:0, S);$ $\{S = ' 7'\}$

$Val(S : String; Var X; var ErrCode : Integer)$ – перетворює рядкове значення S (рядок цифр) в значення числової змінної X. Якщо перетворення можливе, то змінна ErrCode=0, інакше вона містить номер символу в S, на якому відбулася помилка. Тип V повинен відповідати вмісту рядка S.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).
2. По літературі до лабораторної роботи з'ясувати контрольні теоретичні

запитання.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Ввімкнути комп'ютер.
2. Запустити Turbo Pascal.
3. Виконати відповідні практичні завдання з варіантів (варіант вибирається згідно номеру комп'ютера) для самостійного виконання.

Варіант 1

1. Дано текст. Визначити, які символи зустрічаються в тексті частіше: “а” чи “о”. Якщо якийсь із символів відсутній – повідомити про це.
2. Надрукувати всі слова тексту, що містять рівно три літери 'о'.

Варіант 2

1. Вилучити з тексту всі слова, що складаються з однієї літери, підрахувавши їх кількість.
2. З кожного слова непарної довжини вилучити середню літеру.

Варіант 3

1. Дано деякий текст, у якому є хоча б одна кома. Визначити порядковий номер: першої коми в тексті; останньої коми в тексті.
2. Надрукувати всі симетричні слова тексту.

Варіант 4

1. Дано деякий текст. Відредагувати його таким чином, щоб після кожної коми або крапки стояв хоча б один пропуск.
2. Порівняти кількість слів парної довжини і слів, що містять літеру 'к'.

Варіант 5

1. Скласти програму, що вилучає із заданого слова всі літери “а” (наприклад, “застава” – “зств”).
2. Надрукувати слова, які входять в текст не менше двох разів.

Варіант 6

1. Підрахувати кількість входжень в текст першого слова.
2. Визначити номер слова, що містить максимальну кількість букв 'о'.

Варіант 7

1. Дано деякий текст. Відредагувати його таким чином, щоб після кожної крапки (після крапки є хоча б один пропуск) наступне слово починалось з великої літери.
2. Надрукувати слова з парними номерами, які містять непарну кількість символів.

Варіант 8

1. У даному тексті всі послідовності крапок замінити на одну крапку.
2. Нехай дано формулу. Визначити коректність формули щодо кількості відкритих та закритих дужок. Вважається, що закриті дужки не стоять перед відкритими. Якщо дужки у формулі відсутні – повідомити про це.

Варіант 9

1. Скласти програму, яка кожному літеру “а”, що зустрічається в тексті, замінює на групу символів “ку” (наприклад, “ади” – “куди”).
2. Надрукувати слова, довжина яких максимальна.

Варіант 10

1. Скласти алгоритм-шифрувальник, який замінює кожний символ тексту його ASCII-кодом.
2. Порівняти кількість слів, що містять літеру 'к' і слів, що складаються з чотирьох літер.
4. Показати викладачу виконані завдання на будь-якому носії інформації. Оформити лабораторну роботу згідно звіту до роботи.
5. Захистити роботу.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Для чого використовуються Pred, Succ, Chr, Ord? Опишіть їх дію на прикладах.
2. Вкажіть формат процедури Insert. Наведіть приклади її використання.
3. Вкажіть формат процедури Delete. Наведіть приклади її використання.
4. Вкажіть формат функції Copy. Наведіть приклади її використання.
5. Вкажіть формат функції Pos. Наведіть приклади її використання.
6. Вкажіть формат процедури Val. Наведіть приклади її використання.
7. Вкажіть формат процедури Str. Наведіть приклади її використання.
8. Якими способами можна з'єднати кілька рядків?
9. Вкажіть формат функції Length. Наведіть приклади її використання.

ЛІТЕРАТУРА

1. Абрамов С.А. и др. Задачи по программированию. – М.: Наука, 1988. – 224 с.
2. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык паскаль. – М.: Наука, 1988. – 320 с.
3. Грогоно П. Программирование на языке Паскаль. – М.: Наука, 1982.
4. Йенсен К., Вирт Н. Паскаль: руководство для пользователя и описание языка. – М.: Финансы и статистика, 1982.
5. Рамський Ю.С., Цибко Г.Ю. Основи програмування (мовою Паскаль). Курс лекцій. Лабораторний практикум. – К.: НПУ імені М.П. Драгоманова, – 2004.
6. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989. – 160 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на питання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту (варіант вибирається згідно номера свого комп'ютера).
3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №6 МНОЖИНИ У МОВІ ТР

МЕТА: Ознайомитись з можливостями мови Турбо Паскаль (ТР) в опрацюванні множин. Закріпити вивчений матеріал при створенні власних нескладних програм опрацювання множин.

ЗНАТИ: стандартні процедури і функції опрацювання множин у ТР;

ВМІТИ: створювати власні програми для опрацювання множин;

ОБЛАДНАННЯ: технічне забезпечення: ПЕОМ, програмне забезпечення: система програмування Turbo Pascal 6.0.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Множина (Set) – це структурований тип даних, який подає набір взаємопов'язаних за певною ознакою або групою ознак об'єктів, які можна розглядати як єдине ціле. Кожний об'єкт у множині називається елементом множини. Всі елементи множини повинні належати одному зі скалярних типів, крім дійсного. Цей тип називається базовим типом множини. Базовий тип задається діапазоном або переліком. Область значень типу множина – набір можливих підмножин, складених з елементів базового типу.

Для опису множинного типу використовується словосполучення Set of (множина з).

Формат опису множини:

Type

<ім'я типу> = Set of <базовий скалярний тип>;

Var

<ім'я змінної> : <ім'я типу>;

Var

<ім'я змінної> : Set of <базовий скалярний тип>;

Приклад.

Type

SetChar = Set of Char; {множина з символів}

SetDigit = Set of 0..9; {множина з чисел від 0 до 9}

SetDChar = Set of '0'..'9'; {множина з символів '0', '1', ..., '9'}

SetSpring = Set of (March, April, May); {множина з весняних місяців}

В Турбо Паскалі дозволяється визначати множини, які складаються не більше, ніж з 256 елементів. Стільки ж елементів містять типи Byte (0..255) і Char, і це ж число є обмеженням кількості елементів у будь-якому іншому перелічувальному базовому типі множини, який задає користувач. Кожному елементу перелічувального типу приписується певний номер. Нумерація здійснюється від 0 до 255.

Змінна типу множина підлягає певному синтаксису. Елементи множини повинні братися у квадратні дужки.

Приклад.

SByte := [1,2,3,4,10,20,30,40];

SChar := ['a', 'b', 'c'];

SSpring := [April];

SDiap := [1..4]; *{те ж саме, що і [1,2,3,4]}*

SComp := [1..4, 5,7,10..20];

Empty := [];

Порожня множина записується як [].

Порядок слідування елементів всередині дужок не має значення, не має значення і кількість повторення елементів. Повторні входження елементів ігноруються.

Так, записи ['a','b','b','a'] і ['a','b'] еквівалентні.

Операції над множинами

= *перевірка на рівність* – результатом є логічне значення, рівне TRUE, якщо S1 і S2 складаються з однакових елементів незалежно від порядку слідування, і FALSE у протилежному випадку.

<> *перевірка на нерівність* – результатом є логічне значення, рівне TRUE, якщо S1 і S2 відрізняються хоча б одним елементом, і FALSE у протилежному випадку.

<= *перевірка на підмножину* – результатом є логічне значення, рівне TRUE, якщо всі елементи S1 містяться і в S2 незалежно від їх порядку слідування, і FALSE у протилежному випадку.

>= *перевірка на надмножину* – результатом є логічне значення, рівне TRUE, якщо всі елементи S2 містяться в S1, і FALSE у протилежному випадку.

in *перевірка входження елемента у множину* – результатом є логічне значення, рівне TRUE, якщо значення E належить базовому типу множини і входить у множину [...] (S1). Якщо множина не містить у собі значення K, то результатом є FALSE.

+ *об'єднання множин* – результатом об'єднання є множина, отримана злиттям елементів цих множин і виключенням елементів, що повторюються.

- *різниця множин* – результатом операції отримання різниці S1 – S2 є множина, складена з елементів, які входять в S1, але не входять в S2.

*** *перетин множин* – результатом перетину є множина, що складається лише з тих елементів S1 і S2, які містяться одночасно і у S1, і у S2.

Основним недоліком є те, що Турбо Паскаль не дозволяє виводити множини на екран і отримувати окремі елементи з множин. Введення множин можливе лише по елементам.

Приклад.

В порядку спадання вивести всі цілі числа з діапазону 1..4900, які можна подати у вигляді $n^2 + 2k^2$, але не можна подати у вигляді $7ij + j + 3$ ($n, k, i, j \geq 0$).

Uses Crt;

Type S = 1..4900;

SS = Set of S;

Var p: S;

S1, S2: SS;

n, k: integer;

Begin

S1 := []; S2 := []; {Ініціалізація множин}

for n:= 0 to 70 do

```

for k:= n to 70 do
  begin
    S1 := S1 +[n*n +2*k*k];
    S2 := S2 + [7*n*k + k + 3]
  end;
for p := 4900 downto 1 do
  if (p in S1) and (not (p in S2))
  then writeln(p)
End.

```

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера).
2. По літературі до лабораторної роботи з'ясувати контрольні теоретичні запитання.
3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Ввімкнути комп'ютер.
2. Запустити Turbo Pascal.
3. Виконати відповідні практичні завдання з варіантів (варіант вибирається згідно номеру комп'ютера) для самостійного виконання.

Варіант 1

1. Ввести послідовність, що складається з цифр. Кожну третю цифру записати до множини S1, інші цифри – до множини S2. Знайти перетин множин S1 і S2. Ті цифри, які не увійшли до перетину множин, записати до нової множини.

Варіант 2

1. Ввести послідовність великих і маленьких літер латинського алфавіту. Сформувані дві множини: S1 – множина з великих літер, S2 – з маленьких. Надрукувати елементи отриманих множин в алфавітному порядку.

Варіант 3

1. Ввести дві послідовності маленьких літер латинського алфавіту. Сформувані дві множини: множину S1 з приголосних літер першої послідовності і множину S2 з приголосних літер другої послідовності. Якщо множина S1 є підмножиною S2, то знайти різницю множин S2 – S1.

Варіант 4

1. Ввести послідовність з великих і маленьких літер російського алфавіту. Сформувані три множини: множину з великих голосних літер, множину з великих приголосних літер і множину з маленьких приголосних літер. Вивести елементи всіх створених множин у алфавітному порядку.

Варіант 5

1. Ввести послідовність, що складається з літер російського та латинського

алфавітів, цифр та знаків пунктуації (крапка, кома, тире тощо). Заносити символи до множин: S1 – літери російського алфавіту, S2 – літери латинського алфавіту, S3 – цифри, S4 – знаки пунктуації. Знайти кількість літер у кожній множині.

Варіант 6

1. Ввести послідовність, що складається з літер латинського алфавіту і цифр. Переписати введені літери в алфавітному порядку, а числа в порядку спадання.

Варіант 7

1. Ввести послідовність символів російського алфавіту і записати їх до множини S1. До множини S2 переписати всі глухі приголосні з S1 та вивести її на екран.

Варіант 8

1. Ввести послідовність з цілих чисел. Сформувати множину з чисел, кратних п'яти, та множини з парних і непарних чисел. Знайти перетин та різницю двох перших множин та надрукувати елементи останньої в алфавітному порядку.

Варіант 9

1. Ввести послідовність літер латинського алфавіту. Кожну другу літеру записати до множини S1, інші – до множини S2. Знайти об'єднання, перетин і різницю отриманих множин.

Варіант 10

1. Ввести послідовність з цілих чисел. Сформувати множину з парних чисел та множину з чисел, кратних 3. Знайти перетин цих множин.

4. Показати викладачу виконані завдання на будь-якому носії інформації. Оформити лабораторну роботу згідно звіту до роботи.

5. Захистити роботу.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Для чого призначений тип даних Set? Які його переваги і недоліки?
2. Як описується множинний тип засобами TR?
3. Які типи даних можуть використовуватись як базовий тип множини?
4. Скільки елементів може містити множина в TR? Скільки різних множин можна утворити на основі заданого базового типу?
5. Вкажіть операції над множинами у TR, результатом яких є логічне значення. Наведіть приклади їх використання.
6. Вкажіть операції над множинами у TR, результатом яких є множина. Наведіть приклади їх використання.
7. Як здійснюється доступ до окремого елемента множини?
8. Чи має значення порядок слідування елементів у множині?

ЛІТЕРАТУРА

1. Абрамов С.А. и др. Задачи по программированию. – М.: Наука, 1988. – 224 с.
2. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык паскаль. – М.: Наука, 1988. – 320 с.
3. Грогно П. Программирование на языке Паскаль. – М.: Наука, 1982.
4. Йенсен К., Вирт Н. Паскаль: руководство для пользователя и описание языка. – М.: Финансы и статистика, 1982.

5. Рамський Ю.С., Цибко Г.Ю. Основи програмування (мовою Паскаль). Курс лекцій. Лабораторний практикум. – К.: НПУ імені М.П. Драгоманова, – 2004.

6. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989. – 160 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на питання для самостійної підготовки.

2. Виконати завдання згідно свого варіанту (варіант вибирається згідно номера свого комп'ютера).

3. Зробити висновок.

ЛАБОРАТОРНА РОБОТА №7 ОПРАЦЮВАННЯ ГРАФІКИ У ТР

МЕТА: ознайомитись з можливостями мови Турбо Паскаль (ТР) в опрацюванні графіки. Закріпити вивчений матеріал при створенні власних нескладних програм опрацювання графіки у ТР.

ЗНАТИ: особливості створення графічних об'єктів у ТР, вивчити стандартні процедури і функції створення графічних примітивів та роботи з ними.

ВМІТИ: застосовувати вивчений матеріал при написанні власних програм для створення графічних зображень.

ОБЛАДНАННЯ: технічне забезпечення: ПЕОМ, програмне забезпечення: система програмування Turbo Pascal 7.0.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

В текстовому режимі екран розбивається на 25 рядків по 80 позицій у кожному. В кожен позицію (знакомісце) може бути виведений певний символ з кодової таблиці.

В графічному режимі екран складається з окремих точок, які називаються *пікселями*. Кожен піксель характеризується своїми координатами і кольором.

Робота з графікою в ТР відбувається у три етапи:

1. ініціалізація графічного режиму;
2. виконання графічних побудов;
3. закриття графічного режиму.

Для ініціалізації графічного режиму призначена процедура:

Procedure InitGraph(Var GraphDriver: Integer; Var GraphMode: Integer; DriverPath: String);

GraphDriver – номер графічного драйвера,

GraphMode – номер графічного режиму для даного драйвера,

DriverPath – шлях до графічного драйвера.

Про успішність ініціалізації може повідомляти функція:

Function GraphResult: Integer;

Якщо результатом функції є константа *GrOk* (за замовчуванням 0), то ініціалізація пройшла успішно. В іншому випадку повертається код помилки.

Для закінчення роботи з графікою призначена процедура

Procedure CloseGraph;

Вона звільняє пам'ять, зайняту під графічний драйвер, і переводить екран у текстовий режим.

Будь-яка програма має вигляд:

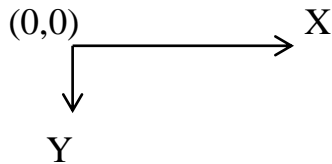
```
Program <ім'я програми>;                                     {не обов'язково}
uses Graph; {обов'язково для роботи в графічному режимі}
var  grDriver, grMode, errCode: Integer; {ці змінні використовуються
процедурою InitGraph}
...
begin                                                         { тіло програми }
...
grDriver := Detect;                                           { автоматичне визначення номера драйвера }
InitGraph(grDriver,grMode,'шлях до драйвера'); {шлях до драйвера,
```

```

errCode := GraphResult;
if errCode = grOK then
  begin
    ...
    CloseGraph;           {закриває режим графіки }
  end
else
  begin                   { режим не удалось открыть}
    writeln('GraphErrorMsg(errCode)'); {на екран виводиться відповідне
    текстове повідомлення про помилку,
    яка відповідає коду errCode}
  end;
end.

```

Система координат екрана має вигляд:



Для визначення максимальних координат екрана:

Function GetMaxX: Integer;

Function GetMaxY: Integer;

Для визначення поточного положення графічного курсору (ГК) на екрані використовуються функції:

Function GetX: Integer;

Function GetY: Integer;

Procedure MoveTo(x,y: Integer); – переміщує ГК в точку з координатами (x,y)

Procedure MoveRel(dx,dy: Integer); – переміщує ГК на dx по горизонталі і dy по вертикалі від поточної позиції

Procedure ClearDevice; – очищує екран і переводить графічний курсор в позицію (0,0);

Procedure GraphDefaults; – очищує екран і перевстановлює всі параметри як за замовчуванням.

На графічному екрані може бути визначене графічне вікно – прямокутна частина екрану, у верхній лівий кут якої переноситься початок системи координат:

Procedure SetViewPort(x1, y1, x2, y2: Integer; ClipMode: Boolean);

(x1, y1)-(x2, y2) – діагональ вікна;

Параметр *ClipMode* – може мати два значення: *ClipOn(TRUE)* – зображення не виводиться за межі графічного вікна (обрізається) і *ClipOff(FALSE)* – зображення “не зважає” на межі вікна.

Procedure ClearViewPort; – очищення робочого простору графічного вікна (фактично, відмови від нього).

Побудова основних графічних примітивів

Procedure PutPixel(x,y: Integer; Color: Word); – зафарбовує піксель (x,y) у

колір Color;

Function GetPixel(x,y: Integer):Word; – повертає значення кольора точки (x,y);

Procedure Line(x1, y1, x2, y2: Integer); – будує відрізок, кінці якого мають координати (x1,y1), (x2,y2);

Procedure LineTo(x,y: Integer); – будує відрізок з поточної точки до точки (x,y);

Procedure LineRel(dx, dy: Integer); – будує відрізок відносно поточної позиції ГК.

Лінії будуються за попередньо встановленими стилем і кольором.

Procedure SetLineStyle(LineStyle, Pattern, Thickness: Word); – встановлення стилю ліній

параметр *LineStyle* – тип ліній: 0 – звичайна лінія, 1 – точечна, 2 – штрихпунктирна, 3 – пунктирна, 4 – тип ліній явно задається шаблоном;

параметр *Pattern*, якщо *LineStyle* \neq 4, дорівнює 0, інакше необхідно задати шаблон з 16 пікселів (1 - світиться, 0 – ні: 1100110011001100), перевівши у десяткову або шістнадцяткову систему числення;

параметр *Thickness* – товщина ліній. 1 – нормальна товщина (1 піксель), 3 – потовщена лінія.

Для встановлення кольору фону і кольору пера (побудов) використовуються процедури:

Procedure SetColor(Color: Word); - колір пера;

Procedure SetBkColor(Color: Word); - колір фону.

Отримання поточних значень кольорів:

Function GetColor: Word; - повертає поточний колір пера;

Function GetBkColor: Word; - повертає поточний колір фону.

Для попередньо встановлюється шаблон і колір заливки:

Procedure SetFillStyle(Pattern: Word; Color: Word); – зафарбовування замкнених областей

Procedure Rectangle(x1,y1,x2,y2: Integer); малює прямокутник з діагоналлю (x1,y1) – (x2,y2);

Procedure Circle(x,y: Integer; Radius: Word); – зображення кола.

Procedure Arc(x, y: Integer; StartAngle, EndAngle, Radius: Word); – дуга радіуса Radius з центра (x, y) від кута StartAngle до EndAngle;

Procedure Ellipse(x, y: Integer; StartAngle, EndAngle, XRadius, YRadius: Word); – еліптична дуга, XRadius і YRadius – розміри горизонтальної і вертикальної полуосей. Для зображення еліпса слід задати StartAngle=0, EndAngle=360.

Procedure Bar(x1, y1, x2, y2: Integer); – прямокутник з діагоналлю (x1, y1)-(x2, y2), внутрішня область якого залита за поточним шаблоном;

Procedure Sector(x, y: Integer; StartAngle, EndAngle, XRadius, YRadius: Word); – сектор еліпса, залитий за заданим шаблоном;

Procedure FillEllipse(x, y: Integer; XRadius, YRadius: Word); – еліпс поточного кольору, залитий за поточним шаблоном;

Procedure FloodFill(x,y: Integer; Border: Word); – універсальна процедура, яка заливає всю область навколо точки (x,y), обмежену лініями кольору Border.

Procedure OutText(TextString: String); – виводить на екран рядок TextString від позиції графічного курсора;

Procedure OutTextXY(x, y: Integer; TextString: String); – виводить на екран рядок TextString від позиції (x,y);

Можна задати власні параметри тексту:

Procedure SetTextStyle(Font, Direction: Word; CharSize: Word);

Font визначає номер шрифту: 0 – матричний шрифт 8x8 (за замовчуванням), 1 – напівжирний шрифт, 2 – потончений, 3 – рублений, 4 – готичний;

Direction – розташування тексту: 0 - горизонтальне (за замовчуванням), 1 – вертикальне знизу вгору;

CharSize – розмір символів: 0 – стандартний розмір, 1-10 – стандартний розмір з відповідним коефіцієнтом.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1. Записати в зошит тему, мету, обладнання, практичні завдання свого варіанту (вибирається згідно номеру комп'ютера.) .

2. По літературі до лабораторної роботи з'ясувати контрольні теоретичні запитання.

3. По інструкції до лабораторної роботи ознайомитись з порядком виконання роботи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Ввімкнути комп'ютер.
2. Запустити Turbo Pascal.
3. Виконати відповідні практичні завдання з варіантів (варіант вибирається згідно номеру комп'ютера) для самостійного виконання.

Варіант №1

1. Створити програму, яка виводить на екран малюнок 1 (Табл.1).

Варіант №2

1. Створити програму, яка виводить на екран малюнок 2 (Табл.1).

Варіант №3

1. Створити програму, яка виводить на екран малюнок 3 (Табл.1).

Варіант №4

1. Створити програму, яка виводить на екран малюнок 4 (Табл.1).

Варіант №5

1. Створити програму, яка виводить на екран вигляд мобільного телефону.

Варіант №6

1. Створити програму, яка виводить на екран зображення вантажної машини.

Варіант №7

1. Створити програму, яка виводить на екран малюнок 5 (Табл.1).

Варіант №8

1. Створити програму, яка виводить на екран малюнок 6 (Табл.1).

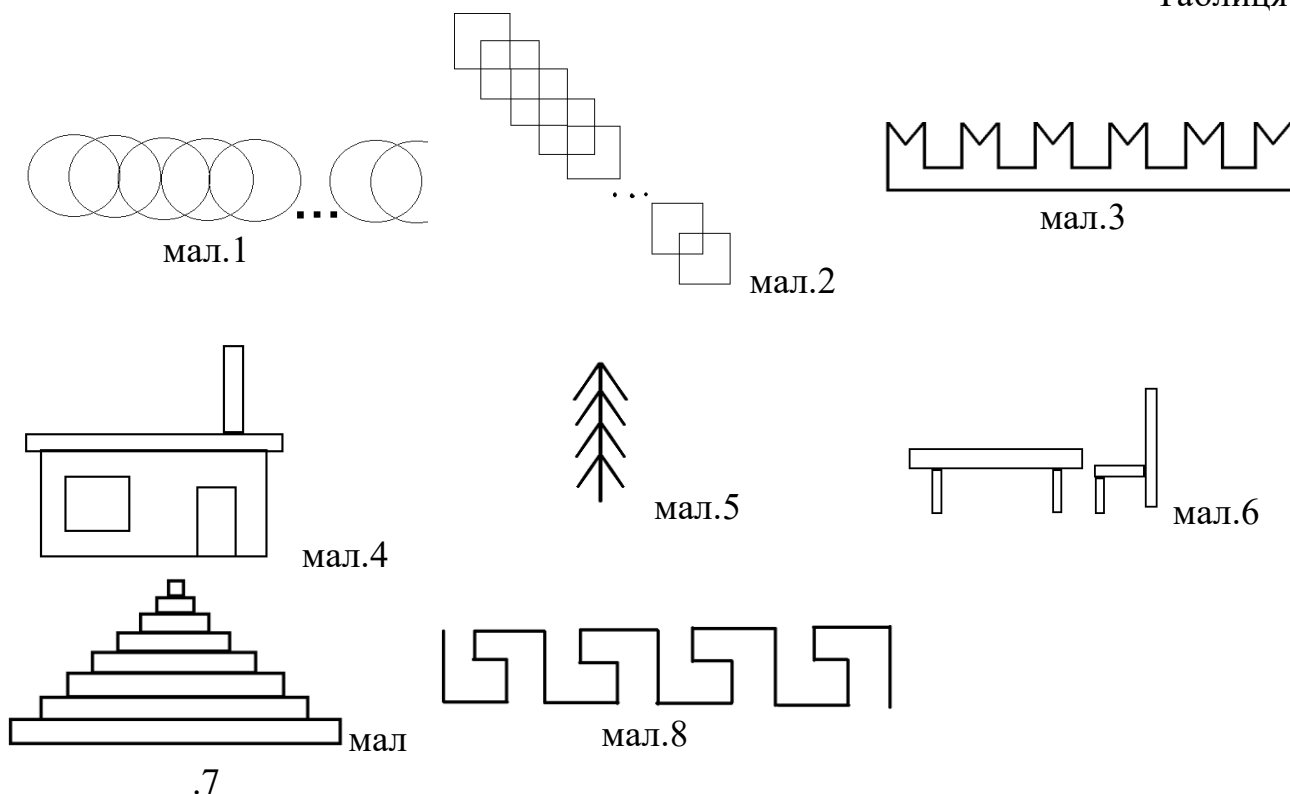
Варіант №9

1. Створити програму, яка виводить на екран малюнок 7 (Табл.1).

Варіант №10

1. Створити програму, яка виводить на екран малюнок 8 (Табл.1).

Таблиця 1.



4. Показати викладачу виконані завдання на будь-якому носії інформації. Оформити лабораторну роботу згідно звіту до роботи.
5. Захистити роботу.

ПИТАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Чим розрізняється робота в текстовому і графічному режимах роботи комп'ютера?
2. Що таке роздільна здатність?
3. Як реалізується графічний режим у програмах на ТР?
4. Як відбуваються ініціалізація і закриття графічного режиму?
5. Що таке графічний курсор? Як його переміщувати по екрану?
6. Як відбувається побудова основних графічних примітивів? Наведіть приклади.
7. Як відбувається виведення тексту у графічному режимі? Наведіть приклади.
8. Як відбувається робота з фрагментами зображень у ТР?

ЛІТЕРАТУРА

1. Абрамов С.А. и др. Задачи по программированию. – М.: Наука, 1988. – 224 с.
2. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык паскаль. – М.: Наука, 1988. – 320 с.
3. Грогоно П. Программирование на языке Паскаль. – М.: Наука, 1982.

4. Йенсен К., Вирт Н. Паскаль: руководство для пользователя и описание языка. – М.: Финансы и статистика, 1982.
5. Рамський Ю.С., Цибко Г.Ю. Основи програмування (мовою Паскаль). Курс лекцій. Лабораторний практикум. – К.: НПУ імені М.П. Драгоманова, – 2004.
6. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М.: Наука, 1989. – 160 с.

ЗВІТ ДО РОБОТИ

1. Оформити лабораторну роботу в зошиті. Записати тему, мету, відповіді на питання для самостійної підготовки.
2. Виконати завдання згідно свого варіанту (варіант вибирається згідно номера свого комп'ютера).
3. Зробити висновок.

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Тема 1. Створення програм за допомогою умовного оператора та оператору варіанту

Варіант 1

1. Від річкового вокзалу відійшли одночасно у протилежних напрямках теплохід та турист. Теплохід рухався зі швидкістю V_1 км/год, а турист по стежці вздовж річки зі швидкістю V_2 км/год. Якщо через N годин турист передумає і вирішить попливти річкою назад за теплоходом зі швидкістю V_3 км/год, то чи встигне він підсісти на теплохід, який має за графіком зупинку через Y годин після початку руху і стоїть на цій зупинці Z годин? Вважати на те, що всі події відбувалися протягом однієї доби.

Вказівка. Знайти відстань, яка між теплоходом і туристом через N годин. Час, який буде у туриста для наздоганяння, $(Y-N+Z)$ годин, тому що зупинка в теплохода буде за розкладом через Y годин після початку руху, але N годин він уже плив, а Z годин теплохід буде стояти на зупинці.

2. Розробити програму виведення текстового варіанту шкільних оцінок:

- 1) 1, 2, 3 – початковий рівень;
- 2) 4, 5, 6 – середній рівень;
- 3) 7, 8, 9 – достатній рівень;
- 4) 10, 11, 12 – високий рівень.

Варіант 2

1. За рейтинговою системою оцінка визначається таким чином: якщо загальний бал учня становить не менше 92% від максимального, то виставляється оцінка 12, якщо не нижче 70%, то оцінка 8, якщо ж не нижче 50%, то оцінка 5, в інших випадках – оцінка 2. Визначте оцінку учня, якщо він набрав N балів, а максимальне значення загального балу становить S .

Вказівка. Треба знайти %-відношення балів учня до максимально можливого.

2. Введемо такі позначення для відмінків в українській мові:

- «називний» - «н» або «Н»;
- «родовий» - «р» або «Р»;
- «давальний» - «д» або «Д»;
- «знахідний» - «з» або «З»;
- «орудний» - «о» або «О»;
- «місцевий» - «м» або «М»;
- «кличний» - «к» або «К».

Розробити програму, яка за введеним позначенням відмінка видаватиме запитання, на які відповідає іменник у вказаному відмінку, наприклад: „називний” – „хто? що?”

Варіант 3

1. Дано три числа a , b , c . Якщо ці числа можуть розглядатись як сторони трикутника, то необхідно обчислити площу цього трикутника за формулою

Герона ($S = \sqrt{p(p-a)(p-b)(p-c)}$, де $p = \frac{a+b+c}{2}$). В протилежному випадку вважати $S=0$.

2. Розробити програму виведення кількості днів у місяці, якщо він задається номером від 1 до 12.

Варіант 4

1. Якщо сума трьох різних дійсних чисел x, y, z менше за одиницю, то найменше з них замінити напівсумою двох інших; в іншому випадку замінити менше з x, y напівсумою двох, що залишилися.

2. Дано ціле число n ($1 \leq n \leq 12$), яке вказує порядковий номер місяця в році. За введеним значенням n надрукувати назву відповідного місяця.

Варіант 5

1. Дано дійсні числа a, b, c, d . Якщо $a \leq b \leq c \leq d$, то кожне число замінити найбільшим з них; якщо $a > b > c > d$, то числа залишити без змін; в іншому випадку всі числа замінити їх квадратами. Результат вивести на екран.

2. Дано ціле число n ($1 \leq n \leq 4$), яке визначає порядковий номер кварталу року (січень, лютий, березень – 1 квартал і т.д.). За вказаним значенням n надрукувати перелік місяців, які належать до цього кварталу.

Варіант 6

1. Дано чотири числа b, c, d, e . Знайти $\max(b, d, e)$, $\min(b, c, e)$.

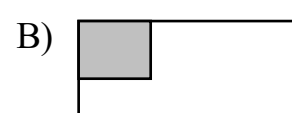
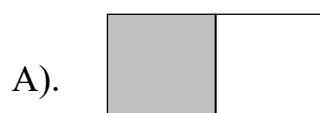
2. Розробити програму-довідник, яка за введеним значенням радіуса R та номером задачі пропонуватиме користувачу послуги у обчисленні:

- 1) 1 – довжина кола;
- 2) 2 – площі круга;
- 3) 3 – об'єму кулі.

Варіант 7

1. Дано прямокутний лист металу. Довжина A см, а ширина B см. Від листа відрізали кусок квадратної форми. Визначити, який кусок залишився: менший, більший чи такого ж розміру, а також якої форми.

Вказівка. Залишок може мати квадратну (варіант А), прямокутну (варіант Б) або іншу форми (варіант В).



2. Введемо такі позначення для відмінків в українській мові:

- «називний» - «н» або «Н»;
- «родовий» - «р» або «Р»;
- «давальный» - «д» або «Д»;
- «знахідний» - «з» або «З»;
- «орудний» - «о» або «О»;
- «місцевий» - «м» або «М»;

«кличний» - «к» або «К».

Розробити програму, яка за введеним позначенням відмінка видаватиме запитання, на яке відповідає прикметник у вказаному відмінку, наприклад: „називний” – „який?”

Варіант 8

1. Три точки задані своїми координатами $A(x_1, y_1)$, $B(x_2, y_2)$ і $C(x_3, y_3)$. Знайти ту з трьох точок, яка лежить найближче до початку координат.

2. Розробити програму видачі номера кварталу, до якого відноситься місяць, заданий числом від 1 до 12.

Варіант 9

1. Дано дійсні числа x і y . З'ясувати, в якій координатній чверті знаходиться точка з координатами (x, y) .

2. За даним цілим значенням змінної ($1 \leq k \leq 6$), яка визначає день тижня, надрукувати свій розклад занять.

Варіант ****

1. Визначити, чи пройде цеглина з ребрами a , b , c у прямокутний отвір зі сторонами x та y . Проштовхувати цеглину дозволяється лише так, щоб кожне з її ребер було паралельним чи перпендикулярним кожній зі сторін отвору.

Вказівка. Для розв'язання цієї задачі треба впорядкувати розміри отвору та розміри цеглини за зростанням, тобто досягти того, щоб було $a \leq b \leq c$ та $x \leq y$. Тоді перевірка зведеться до порівняння розмірів отвору з найменшими розмірами цеглини (адже ми можемо цеглину розвернути будь-яким боком, щоб проштовхнути її у отвір).

2. Дано натуральне число N (N не перевищує 1000). Визначити суму першої та останньої цифр даного числа.

Вказівка. Скористатися стандартними операціями цілочисельного ділення та остачі від ділення цілих чисел.

3. Дано натуральне число N від 100 до 1000000. Якщо воно менше 1000, то визначити, скільки воно має десятків, якщо від 1000 до 10000 – скільки сотень, від 10000 до 100000 – скільки тисяч, від 100000 до 1000000 – скільки десятків тисяч.

Вказівка. Використати стандартну функцію цілочисельного ділення.

4. Дано натуральне число N (N не перевищує 100), яке позначає вік людини. Додати до цього числа відповідно слова: “рік”, “роки”, “років”, наприклад: 1 рік, 12 років, але 43 роки.

Вказівка. Щоб правильно дописати відповідне слово, необхідно виділити останню цифру числа, що позначає вік людини. Тоді, якщо це цифра “1”, то дописується слово “рік”, якщо цифри “2”, “3” або “4” – дописується слово “роки”, а в усіх останніх випадках – дописується слово “років”. Виключенням являється діапазон між 10 та 20 роками: в цих випадках завжди пишеться слово “років”.

Тема 2. Цикл з лічильником. Цикли з передумовою та післяумовою

Варіант №1

1. Написати програму, яка виводить на екран таблицю з десяти рядків, кожний з яких містить десять чисел вигляду: $m.n$, де m – номер рядка, а n – номер числа у рядку.

2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність e ($e > 0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за e . Обчислити:

$$\sum_{i=1}^{\infty} \frac{i - \cos i}{\sqrt{i^2 + 2i}}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю i і якщо доданок менше, обчислення припиняються.

Варіант №2

1. Написати програму, яка виводить на екран таблицю для визначення повного опору Z , сили струму I та косинуса кута зсуву результуючої напруги відносно струму для кола з послідовним з'єднанням активного опору, індуктивності. $R=7,8, \dots, 17$.

R	$Z = \sqrt{R^2 + X_L^2}$	$I = \frac{U}{Z}$	$\cos \varphi = \frac{R}{Z}$

2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність e ($e > 0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за e . Обчислити:

$$\sum_{i=1}^{\infty} \frac{(-2)^i}{i(i+1)(i+2)}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю i і якщо доданок менше, обчислення припиняються.

Варіант №3

1. Написати програму, яка виводить на екран таблицю множення.

2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність e ($e > 0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за e . Обчислити:

$$\sum_{i=1}^{\infty} \frac{1}{i^2}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю і якщо доданок менше, обчислення припиняються.

Варіант №4

1. Написати програму, яка виводить на екран таблицю для визначення повного опору Z , сили струму I та косинуса кута зсуву результуючої напруги відносно струму для кола з послідовним з'єднанням активного опору, індуктивності та ємності. $R=1,2, \dots, 15$.

R	$Z = \sqrt{R^2 + (X_L - X_C)^2}$	$I = \frac{U}{Z}$	$\cos \varphi = \frac{R}{Z}$

2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність e ($e>0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за e . Обчислити:

$$\sum_{i=1}^{\infty} \frac{i+5}{i^2+5}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю і якщо доданок менше, обчислення припиняються.

Варіант №5

1. Написати програму, яка виводить на екран таблицю для визначення повного опору Z , сили струму I та косинуса кута зсуву результуючої напруги відносно струму кола з послідовним з'єднанням активного опору та ємності. $R=5,6, \dots, 19$. $U=20\text{В}$, $X_C=8\text{ Ом}$.

R	$Z = \sqrt{R^2 + X_C^2}$	$I = \frac{U}{Z}$	$\cos \varphi = \frac{R}{Z}$
-----	--------------------------	-------------------	------------------------------

2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність e ($e>0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за e . Обчислити:

$$\sum_{i=5}^{\infty} \frac{i - \sqrt{i}}{\sqrt{i^2 + |i|}}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю і якщо доданок менше, обчислення припиняються.

Варіант №6

1. Написати програму, яка виводить на екран таблицю для визначення миттєвих значень струму i , напруги u та потужності p , які змінюються в інтервалі часу $t=1, 2, \dots, 18$. $\omega=2/3\pi$.

T	$i=I_m \sin \omega t$	$u=U_m \sin \omega t$	$p=iu$

2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність e ($e>0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за e . Обчислити:

$$\sum_{i=1}^{\infty} \frac{i^3 - 5\sqrt{i}}{\sqrt{i^6 + |i|}}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю і якщо доданок менше, обчислення припиняються.

Варіант №7

1. Написати програму, яка виводить на екран таблицю для визначення повного опору Z кола з послідовним з'єднанням активного опору, індуктивності та ємності. R змінюється від 1 до 15. $\omega=2/3\pi$.

R	$Z = \sqrt{R^2 + (\omega L - \frac{1}{\omega C})^2}$

2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність e ($e>0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за e . Обчислити:

$$\sum_{i=1}^{\infty} \frac{i}{\sqrt{i^2 + 5 \cdot i}}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю і якщо доданок менше, обчислення припиняються.

Варіант №8

1. Написати програму для виведення на екран в два стовпчики чисел із проміжка від 1 до 30: у перший стовпчик – числа парні, у другий – непарні. Оформити виведення у вигляді таблиці.

2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність e ($e>0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за e . Обчислити:

$$\sum_{i=1}^{\infty} \frac{(-1)^i}{i(i+1)^2}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю ϵ і якщо доданок менше, обчислення припиняються.

Варіант №9

1. Створити таблицю множення на 11, 12, ..., 20.
2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність ϵ ($\epsilon > 0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за ϵ . Обчислити:

$$\sum_{i=1}^{\infty} \frac{1 + \cos i}{2^i}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю ϵ і якщо доданок менше, обчислення припиняються.

Варіант №10

1. Написати програму для виведення на екран в рядок через три пропуски чисел із проміжка від 1 до 45 які кратні 3.
2. Під час обчислення результатів деяких експериментів виникає необхідність отримання результату із заданою точністю. Нехай результатом є нескінчена сума, що задається певною формулою, і відома точність ϵ ($\epsilon > 0$) для знаходження наближеного значення результату. Будемо вважати, що необхідна точність досягнута, коли додавання наступного доданку змінює суму на величину, меншу за ϵ . Обчислити:

$$\sum_{i=1}^{\infty} \frac{i - \sqrt{i}}{\sqrt{i^2 + \sqrt{i}}}$$

Вказівка: У даній задачі перед обчисленням чергової суми ми порівнюємо наступний доданок з точністю ϵ і якщо доданок менше, обчислення припиняються.

Тема 3. Робота з масивами в мові TP

Варіант №1

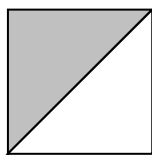
1. Скласти програму заповнення прямокутної таблиці $C[1:m, 1:n]$ числами, які рівні сумі номерів рядка і стовпчика, в яких знаходиться розглядуваний елемент таблиці.

Варіант №2

1. Скласти програму для взаємозаміни значень елементів головної і другорядної діагоналі прямокутної таблиці $A[m:n]$.

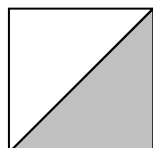
Варіант №3

1. Задано прямокутну таблицю $A[1:N,1:N]$, що містить дійсні числа. Скласти програму знаходження суми значень елементів, що розташовані в заштрихованій частині таблиці.



Варіант №4

1. Задано прямокутну таблицю $A[1:N,1:N]$, що містить дійсні числа. Скласти програму знаходження суми значень елементів, що розташовані в заштрихованій частині таблиці.



Варіант №5

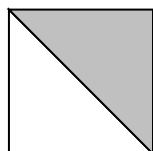
1. Скласти програму заповнення квадратної таблиці $[M:N]$ так, щоб елементам, що стоять на двох бічних діагоналях, було надано значення одиниці, а останнім – нулі. Результат вивести на екран.

Варіант №6

1. Скласти програму для перестановки місцями в лінійній цілочисельній таблиці меншого і більшого елементів.

Варіант №7

1. Задано прямокутну таблицю $A[1:N,1:N]$, що містить дійсні числа. Скласти програму знаходження суми значень елементів, що розташовані в заштрихованій частині таблиці.

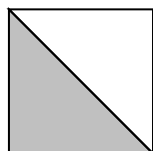


Варіант №8

1. Середню групу дитячого садочка вивели на прогулянку. Скільки дівчаток і скільки хлопчиків видно з-за паркану, якщо зріст хлопчиків задається у сантиметрах від'ємними числами, а дівчаток – додатними у вигляді цілих значень. Крім того, у всіх дівчаток на голівках зав'язані бантики заввишки 10 см, а висота паркану H см.

Варіант №9

1. Задано прямокутну таблицю $A[1:N,1:N]$, що містить дійсні числа. Скласти програму знаходження суми значень елементів, що розташовані в заштрихованій частині таблиці.



Варіант №10

1. Скласти програму обчислення кількості однакових поруч розташованих пар в таблиці A [M:N].

Варіант ****

1. Задано прямокутну таблицю дійсних чисел A[1:N,1:M]. Скласти програму для визначення в ній номера і значення елемента, який є найменшим серед найбільших значень в кожному рядку.

Тема 4. Процедури та функції в мові TP

Варіант №1

1. Три трикутники задаються координатами своїх вершин на площині. Знайти трикутник, що має найбільший периметр, та обчислити площу цього трикутника.

Варіант №2

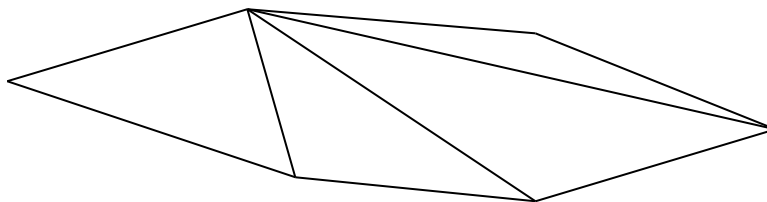
1. Три трикутники задаються координатами своїх вершин на площині. Визначити трикутник, що має найбільшу площу, та обчислити його периметр.

Варіант №3

1. Чотирикутник (довільний) задається координатами вершин на площині. Знайти мінімальну і максимальну відстань між двома вершинами чотирикутника (розглянути і несуміжні вершини).

Варіант №4

1. Шестикутник (довільний) задається координатами вершин на площині. Знайти площу шестикутника.



Варіант №5

1. Дано числа a, b, c . Обчислити

$$f = \begin{cases} \min(a, b, ab) + \min(b, ac, 1), & \text{при } \min(a, b, c) < 10 \\ \max(a^2, b^2, ab) - \max(c^2, a + b, 2a), & \text{в протилежному випадку} \end{cases}$$

Варіант №6

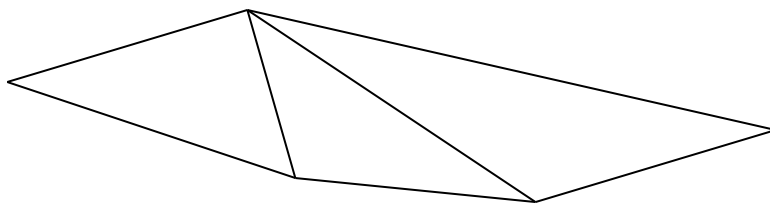
1. Три трикутники задаються координатами своїх вершин на площині. Визначити трикутник, що має найменшу площу, та обчислити його периметр.

Варіант №7

1. Чотири трикутники задаються координатами своїх вершин на площині. Визначити трикутник, що має найменший периметр, та обчислити його площу.

Варіант №8

1. П'ятикутник (довільний) задається координатами вершин на площині. Знайти площу п'ятикутника.



Варіант №9

1. Три трикутники задаються координатами своїх вершин на площині. Визначити трикутник, що має найменшу сторону, та обчислити його площу (для знаходження найбільшої сторони також використати процедуру або функцію).

Варіант №10

1. Три трикутники задаються координатами своїх вершин на площині. Визначити трикутник, що має найбільшу сторону, та обчислити його периметр.

Тема 5. Рядкові величини в мові TP

Варіант № 1

1. Надрукувати всі слова тексту, що містять рівно три літери 'o'.

Варіант №2

1. З кожного слова непарної довжини вилучити середню літеру.

Варіант №3

1. Надрукувати всі симетричні слова тексту.

Варіант №4

1. Порівняти кількість слів парної довжини і слів, що містять літеру 'к'.

Варіант №5

1. Надрукувати слова, які входять в текст не менше двох разів.

Варіант №6

1. Визначити номер слова, що містить максимальну кількість букв 'o'.

Варіант №7

1. Надрукувати слова з парними номерами, які містять непарну кількість символів.

Варіант №8

1. Нехай дано формулу. Визначити коректність формули щодо кількості відкритих та закритих дужок. Вважається, що закриті дужки не стоять перед відкритими. Якщо дужки у формулі відсутні – повідомити про це.

Варіант №9

1. Надрукувати слова, довжина яких максимальна.

Варіант №10

1. Порівняти кількість слів, що містять літеру 'к' і слів, що складаються з чотирьох літер.

Тема 7. Опрацювання записів у мові TP

Варіант №1

1. Описати комбінований тип, який містить відомості про розклад занять: назва дисципліни, викладач, вид заняття (лекція, практичне, лабораторне), номер групи, аудиторія, номер пари, день тижня. Визначити номери аудиторій, в яких будуть заняття у середу.

Варіант №2

1. Описати комбінований тип, який містить основні відомості про розклад руху літаків: місце призначення, рейс, час польоту, вартість квитка, назва літака, чи є квитки. Визначити назви рейсів і місця призначення літаків, час польоту яких менш двох годин, і на них є квитки.

Варіант №3

1. Описати комбінований тип для подання анкети школяра, яка містить в собі прізвище, ім'я, рік народження, стать, номер класу, букву класу, та оцінки з алгебри, фізики, інформатики. Визначити учнів, які не мають 3 і навчаються у 10-"Б".

Варіант №4

1. Описати комбінований тип, який містить основні відомості про розклад руху літаків: місце призначення, рейс, вартість квитка, назва літака, чи є квитки. Визначити три найдорожчі рейси, на які є квитки.

Варіант №5

1. Описати комбінований тип, який містить основні відомості про розклад руху літаків: місце призначення, рейс, вартість квитка, назва літака, чи є квитки. Визначити всі рейси на Москву, на які є квитки.

Варіант № 6

1. Описати комбінований тип, який містить основні відомості про розклад руху літаків: місце призначення, рейс, вартість квитка, назва літака, чи є квитки. Визначити найдорожчий рейс на Київ та вивести всі відомості про нього.

Варіант №7

1. Описати комбінований тип, який містить основні відомості про розклад руху літаків: місце призначення, рейс, вартість квитка, назва літака, чи є квитки. Визначити ті рейси, вартість квитка на які менша двохсот гривень і на них немає квитків.

Варіант №8

1. Описати комбінований тип, який містить відомості про розклад занять: назва дисципліни, викладач, вид заняття (лекція, практичне, лабораторне), номер групи, аудиторія, номер пари, день тижня. Визначити, чи працює певний викладач в певний день та його навантаження в цей день.

Варіант №9

1. Описати комбінований тип для подання анкети школяра, яка містить в собі прізвище, ім'я, рік народження, стать, номер класу, букву класу, та оцінки з алгебри, геометрії, інформатики. Визначити відмінників, що навчаються в 11-"А".

Варіант №10

1. Описати комбінований тип для подання анкети школяра, яка містить в собі прізвище, ім'я, рік народження, стать, номер класу, букву класу, та оцінки з алгебри, геометрії, інформатики. Вивести на екран інформацію про учнів, які мають однакове прізвище, але різні імена.

Тема 6. Опрацювання графіки в мові TP

Варіант №1

1. Створити стіну кімнати, де у вікні буде видно “зоряне небо”.

Варіант №2

1. Створити програму, яка виводить на екран вигляд комп'ютера з заставкою на моніторі “зоряне небо”.

Варіант №3

1. Створити програму, яка зображує на екрані декартову систему координат, початок якої збігається з центром екрана.

Варіант №4

1. Створити програму, яка виводить на екран зображення вантажної машини.

Варіант №5

1. Створити програму, яка на екрані по діагоналі зображує три прямокутники. В першому прямокутнику 10 різнокольорових кіл (центри кіл знаходяться у центрі прямокутника), другий і третій прямокутники заповнити різнокольоровими прямими та колами відповідно (колір, кількість та координати визначаються випадково).

Варіант №6

1. Скласти програму, яка на екран виводить зображення листа аркуша в клітинку.

Варіант №7

1. Скласти програму, яка малює огорожу з N дощечок.

Варіант №8

1. Створити програму, яка виводить на екран вигляд мобільного телефону.

Варіант №9

1. Скласти програму, що малює 21 вертикальну лінію через 30 пікселів на всю висоту екрана.

Варіант №10

1. Створити програму, яка виводить на екран зображення монітора із заставкою

ОБРОБКА ПОМИЛОК В TURBO PASCAL

Кожна програма, написана на Turbo Pascal, перед виконанням компілюється. Компілятор виявляє помилки в тексті програми і видає відповідні повідомлення про них. Якщо помилок немає, то програму можна запускати на виконання. Під час виконання теж можуть виникнути помилки. Тому помилки ділять на дві групи: помилки, які виникають під час компіляції і помилки, які виникають під час виконання. Наведемо основні з них.

ПОМИЛКИ КОМПІЛЯЦІЇ

- 1: Out of memory – вихід за границі пам'яті. Виникає у випадку, якщо компіляція виконується поза допустимими границями пам'яті. В цьому випадку необхідно розбити текст програми на два чи більше програмних модулів.
- 2: Identifier expected – не вказано ідентифікатор. Ідентифікатор повинен знаходитися у казаному місці. Можлива спроба використання зарезервованого слова.
- 3: Unkrown identifier – невідомий ідентифікатор. Цей ідентифікатор не був описаний.
- 4: Duplicate identifier – повторний ідентифікатор. Повторне описання одного і того ж ідентифікатора.
- 5: Syntax error – синтаксична помилка. В тексті знайдено неправильний знак.
- 8: String constant exceeds line – рядкова константа перевищує допустимий розмір рядка. Можливо відсутній апостроф в кінці рядкової константи.
- 10: Unexpected end of file – неправильний кінець файла. Скоріше всього, кількість begin не відповідає кількості end або коментар не закінчено знаком } або *}.
- 11: Line too long – занадто довгий рядок. Максимальна довжина рядка 126 символів
- 12: Type identifier expected – потрібен ідентифікатор типу.
- 26: Type mismatch – невідповідність типів.
- 35: Label identifier expected – потрібен індикатор мітки.
- 36: BEGIN expected – потрібен begin.
- 37: END expected – потрібен end.

- 42: Error in expresion – помилка у виразі.
- 43: Illegal assignment – неправильне присвоєння.
- 50: Do expected – потрібен оператор do.
- 54: OF expected – потрібен of.
- 55: INTERFACE expected – потрібна інтерфейсна секція.
- 57: THEN expected – потрібен then.
- 58: TO or DOWNT0 expected – потрібен to або downto.
- 62: Division bu zero – ділення на нуль.
- 73: Impiementation expected – потрібна секція реалізації.
- 84: Unit expected – потрібен Unit.
- 85: ";" expected – потрібно поставити ";".
- 86: ":" expected – потрібно поставити ":".
- 87: ", " expected – потрібно поставити ", ".
- 88: "(" expected – потрібно поставити "(".
- 89: ")" expected – потрібно поставити ")".
- 91: ":=" expected – потрібно поставити ":=".
- 92: "[" or "(" expected – потрібно поставити "[" або "(".
- 93: "]" or ")" expected – потрібно поставити "]" або ")".
- 94: ". " expected – потрібно поставити ". ".
- 95: ".. " expected – потрібно поставити ".. ".
- 112: CASE constant out of range – константа case виходить за допустимі границі.
- 113: Error in statement – помилка в операторі. Даний символ не може бути першим символом в операторі.
- 131: Header does not match previous definition – заголовок процедури або функції не відповідає попередньому опису.

133: Cannot evaluate this expression – неможливо вирахувати даний вираз.

ПОМИЛКИ ВИКОНАННЯ

2: File not found – файл не знайдено. Помилка генерується процедурами Reset, Rename, або Erase у випадку, якщо файлова зміна вказує на неіснуючий файл.

3: Path not found – шлях не знайдено. Вказаний шлях є недійсним або вказує неіснуючий каталог.

100: Disk read error – помилка читання диску.

101:Disk write error – помилка запису на диск. Генерується процедурами Close, Write, WriteLn у випадку, якщо на диску немає вільного місця.

103:File not open - файл не відкрито.

104:File not open for input – файл не відкрито для вводу.

105:File not open for output – файл не відкрито для виводу.

150:Disk is writen protected – диск захищено від запису.

200:Divizion by zero – ділення на нуль.

201:Range check error – помилка при перевірці границь.

205:Floating point overflow – переповнення при виконанні операцій з плаваючою крапкою.

206:Floating point underflow – зникнення порядку при виконанні операції з плаваючою крапкою.