

ВІННИЦЬКИЙ ДЕРЖАВНИЙ СІЛЬСЬКОГОСПОДАРСЬКИЙ ІНСТИТУТ

**“Комп’ютерна техніка та
програмування”,
“Інформатика та комп’ютерна
техніка”**

*Методичні вказівки до виконання
лабораторних робіт*

Вінниця - 1999

Міністерство агропромислового комплексу України
Вінницький державний сільськогосподарський інститут

МЕТОДИЧНІ ВКАЗІВКИ

(перероблені та доповнені)

до виконання лабораторного практикуму з дисциплін
"Комп'ютерна техніка та програмування" та
"Інформатика та комп'ютерна техніка"

Лабораторний практикум з дисциплін *"Комп'ютерна техніка та програмування" та "Інформатика та комп'ютерна техніка"*
для студентів спеціальностей 70.91.901, 70.50.106 Вінниця, ВДСГІ, 1999 р.

Методичні вказівки (перероблені та доповнені) до виконання лабораторного практикуму з дисципліни "Комп'ютерна техніка та програмування" та "Інформатика та комп'ютерна техніка" для студентів спеціальностей 70.91.901, 70.50.106, 70.50.202. /Укладачі О.В.Бісікало, М.П.Пільніцький, Є.А.Паламарчук, С.Ф.Роботько, Є.Е.Сироватка, Т.В.Січко - Вінниця, ВДСГІ, 1999 р., с.48.

В лабораторному практикумі з дисциплін "Комп'ютерна техніка та програмування" "Інформатика та комп'ютерна техніка" наведено методичні вказівки до виконання дванадцяти лабораторних робіт загальним обсягом 52 годин. Завдання лабораторного практикуму спрямовані на вивчення студентами економічних спеціальностей та спеціальностей механізації сільського господарства алгоритмічної мови програмування Turbo Pascal та програми для роботи з файлами Norton Commander. Особливістю методичних вказівок є можливість наскрізного використання індивідуальних варіантів завдань агропромислового спрямування для всіх лабораторних робіт практикуму.

Розглянуто і схвалено на кафедрі інформатики та АПУ Вінницького Державного сільськогосподарського інституту, протокол N від 1999 р.

Вінниця, 1999 р.

ЗМІСТ

1	Вимоги до оформлення лабораторних робіт	3
2	Лабораторна робота N 1 : Оболонка Turbo Pascal в режимах редактора.	4
3	Лабораторна робота N 2 : Оболонка Turbo Pascal в режимах налагоджувача програм (DEBUGGER) та складання лінійних програм.	7
4	Лабораторна робота N 3 : Проектування та реалізація розгалужених алгоритмів.	13
5	Лабораторна робота N 4 : Складання циклічних програм. Табулювання функцій.	17
6	Лабораторна робота N 5 : Norton Commander та його режими.	20
7	Лабораторна робота N 6 : Обробка масивів.	24
8	Лабораторна робота N 7 : Сортування елементів в масивах.	27
9	Лабораторна робота N 8 : Моделювання облікових задач агропромислового напрямку в середовищі Turbo Pascal.	30
10	Лабораторна робота N 9 : Вивчення графічних можливостей Turbo Pascal (частина I).	37
11	Лабораторна робота N 10 : Вивчення графічних можливостей Turbo Pascal (частина II).	40
12	Лабораторна робота N 11 : Вивчення функцій роботи з рядковими змінними. Електронний довідник з рядковими масивами.	42
13	Лабораторна робота N 12 : Вивчення функцій роботи з рядковими змінними. Електронний довідник з файловою базою даних.	48
14	Лабораторна робота N 13 : Вивчення операторів роботи з текстовим екраном. Створення меню.	50
15	Питання до іститу Література	59

ВИМОГИ ДО ОФОРМЛЕННЯ ЛАБОРАТОРНИХ РОБІТ

Виконувати лабораторні роботи слід за своїм варіантом, номер якого співпадає з номером бригади або вказується викладачем. Лабораторні роботи треба виконувати акуратно, в окремому зошиті, чорнилом будь-якого кольору за винятком червоного.

Звіт про виконання лабораторних робіт обов'язково повинен містити:

- назву теми;
- мету та умови роботи;
- блок-схему алгоритма;
- означення змінних, констант з вказівкою їх діапазона;
- текст програми на алгоритмічній мові;
- інструкція до виконання програми;
- результати виконання (копія екрану);
- висновки по роботі.

Оформлені таким чином роботи повинні бути захищені у визначені викладачем строки. При співбесіді студент повинен знати відповідний теоретичний матеріал і вміти розв'язувати задачі.

ЛАБОРАТОРНА РОБОТА N 1

ТЕМА : **ОБОЛОНКА TURBO PASCAL В РЕЖИМАХ РЕДАКТОРА.**

МЕТА РОБОТИ: Навчитись працювати з текстами в середовищі
TURBO PASCAL.

ЗАВДАННЯ:

1. Навчитись користуватись головним та локальним меню **TURBO PASCAL**.
2. Навчитись набирати та редагувати текст в оболонці **TURBO PASCAL**.
3. Навчитись записувати та завантажувати файли.
4. Навчитись працювати з блоками тексту.
5. Навчитись працювати з вікнами.

ВМІСТ ЗВІТУ:

1. Звіт повинен мати відповіді на всі запитання завдань до лабораторної роботи, а також детальне описання дій по виконанню тієї чи іншої операції.
2. В звіті мають бути наведені висновки за отриманими результатами.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Запустити з меню **NORTON COMMANDER** програму **TURBO PASCAL** (клавіші **F2** та **P**).
2. Увімкнути режим головного меню (клавіші **F10** та **ENTER**).
3. За допомогою клавиш **→** та **←** продивитись зміст головного меню. За допомогою лекційного матеріалу визначити призначення режимів меню **TURBO PASCAL**. Звернути увагу на "гарячі клавіші", а також на додаткові підказки для основних режимів головного меню, що розташовані під рамкою вікна. Занотуйте, які режими можна запускати через локальне меню (**LOCAL MENU** або **ALT_F10**).
4. Увімкнути режим редагування (меню, **FILE, NEW**). Занотуйте в звіт назву файлу, що встановив **TURBO PASCAL** (див. назву в верхній частині рамки робочого вікна).
5. Набрати у вікні розклад дзвінків. При наборі тексту зверніть увагу на роботу клавиш керування курсором. Зверніть увагу на те, що курсор може просуватись вниз тільки до певної позиції. Це свідчить, що далі немає пустих рядків. Спробуйте їх створити, натискаючи клавішу **ENTER**.
6. Спробуйте змінювати окремі символи на екрані монітору за допомогою клавиш **BS** та **DEL**. Занотуйте різницю їх роботи.
7. Натисніть клавішу **INS** декілька разів. Як змінився курсор ? Встановіть курсор в середину будь-якого рядка та спробуйте змінити будь-які

символи коли курсор тонкий (режим авторозсування включений) та коли курсор великий (режим авторозсування вимкнений).

8. Визначте, яку роль грають цифри внизу рамки вікна. Зверніть увагу на зірочку, що розташована біля цифр. Вона свідчить про те, що в файлі були зроблені якісь зміни, тобто його редагували.

9. Запишіть у файл під назвою LESSONS розклад дзвінків, що Ви набрали. Для цього увімкніть режим SAVE ("зберегти") за допомогою меню або "гарячої" (якої ?) клавіші. У вікні, що з'явилося, наберіть LESSONS та натисніть **ENTER**. Занотуйте, що змінилось в назві вікна та що трапилось з зірочкою. Спробуйте записати цей файл вдруге. Чи потрібно тепер вводити ім'я файлу?

10. Закрийте вікно (меню -> **WINDOW** -> **CLOSE** або **ALT_F3**).

11. Завантажте знову файл LESSONS. Для цього або натисніть **F3** або скористуйтесь меню -> **FILE** -> **OPEN**. За допомогою клавіші **TAB** встановіть вказівник на поле Files, клавішами вертикального керування курсором знайдіть файл LESSONS.PAS (зверніть увагу, що при записі **TURBO PASCAL** автоматично додав до назви LESSONS розширення .PAS) та натисніть клавішу **ENTER**.

12. Виконайте п.4. Зверніть увагу на номер вікна та назву файлу. Запишіть його під назвою BIOGRAPH (аналогічно п.9) та наберіть своє прізвище, ім'я та по-батькові, а потім зробіть декілька пустих рядків (див. п.5). Перейдіть в перше вікно, натиснувши **ALT_1**. Поверніться знову в друге вікно (**ALT_3**). Спробуйте робити перехід між вікнами за допомогою меню вікон - (**ALT_0**).

13. Перейдіть в друге вікно. Встановіть курсор на ім'я. Утримуючи клавішу **Shift**, спробуйте рухати курсор в різні боки. Забарвлений текст - це виділений блок. Попрактикуйтесь в позначенні різних частин (слів, рядків) тексту.

14. Утримуючи клавішу **CTRL** послідовно натисніть **К Н** (лат.букви). Повторіть цю дію декілька разів. Що при цьому відбувається?

15. Позначте в блок весь рядок з ІПБ. Запам'ятайте його копію в "кишеню", натиснувши **CTRL_Ins**. При цьому на екрані ніяких змін не відбудеться. Встановіть курсор на будь-який наступний рядок та натисніть **SHIFT_Ins**. В місті розташування курсора має з'явитись той рядок, що ви запам'ятали в "кишені". Спробуйте знову натиснути **SHIFT_Ins**. Що відбулось?

Занотуйте спостереження в звіт. Запам'ятайте, як можна копіювати (розмножувати) блок.

16. Позначте як блок ті рядки, що ви щойно витягли з кишені. Натисніть **SHIFT_Del**. Блок має зникнути, але він в "кишені"! Спробуйте його влаштувати в будь-якому місті вашого тексту, натиснувши **SHIFT_Ins**. Запам'ятайте, як можна пересувати блок.

17. Напишіть, наприклад, свій рік народження, позначте його як блок та натисніть **CTRL_Del**. Спробуйте витягти блок з кишені. Його там не буде, але з'явиться блок, що використовувався в передостанній раз. Отже режим **CTRL_Del** використовується тільки для знищення блоків.

18. Почніть послідовно натискати сполучення **ALT_BS** (режим **UnDo** - "відкат") та уважно прослідкуйте, як буде змінюватись інформація на екрані монітора. Занотуйте, як можна його використовувати.

19. Аналогічно з'ясуйте, як працює режим **ReDo** (меню -> **Edit** -> **ReDo**).

20. Зробіть спробу запам'ятати в блок своє ПІБ та перенесіть його під розклад дзвінків в перше вікно. Скористайтесь методами п.п.12 та 14.

21. Скористайтесь режимом головного меню та з'ясуйте, як розташовуються вікна методом **TILE** (меню -> **Window** -> **Tile**) та черепичним (меню -> **Window** -> **Cascade**) методами. Зробіть спробу переходити між вікнами (**ALT_1**, **ALT_2**, **ALT_3** тощо).

22. Увімкніть режим **CASCADE** та перейдіть на перше вікно. Натисніть **CTRL_F5**. Рамка вікна має змінити колір, що буде свідчити, що тепер можна змінювати розміри вікна та його розташування на екрані. Утримуючи клавішу **SHIFT** за допомогою клавішей керування курсором спробуйте змінювати вертикальні та горизонтальні розміри вікна.

23. Встановивши невеликі розміри вікна, відпустіть клавішу **SHIFT** та спробуйте пересувати вікно по екрану. Натисніть клавішу **ENTER**. Занотуйте, що відбулось.

24. Закінчіть роботу з **TURBO PASCAL**, виконавши дії (меню -> **File** -> **Exit**) або натиснувши **ALT_X**. Якщо з'явиться попередження, що в файл були внесені зміни і **PASCAL** пропонує їх записати, то натисніть клавішу **Y** (**Yes**).

ПИТАННЯ САМОКОНТРОЛЮ:

1. Якими двома способами можна включати режими роботи **TURBO PASCAL**?
2. Як завантажити та записати файл на магнітні носії?
3. Які режими переключає клавіша **Ins**? Як при цьому змінюється курсор?
4. Які надписи є на рамці вікон та для чого вони призначені?
5. Як закрити вікно?
6. Як позначити блок та які дії можна робити з ним?
7. Як можна вмикнути\вимкнути підсвітку блоку?
8. За допомогою яких дій можна змінювати розміри та розташування вікон?

ЛАБОРАТОРНА РОБОТА N 2

ТЕМА: ОБОЛОНКА TURBO PASCAL В РЕЖИМАХ НАЛАГОДЖУВАЧА ПРОГРАМ (DEBUGGER) ТА СКЛАДАННЯ ЛІНІЙНИХ ПРОГРАМ.

МЕТА РОБОТИ: Навчитись складати лінійні програми та працювати в режимах налагоджувача програм.

ЗАВДАННЯ:

1. Дослідити роботу налагоджувача TURBO PASCAL (Debugger) на прикладі програми N1 з додатку лабораторної роботи .
2. Скласти алгоритм та лінійну програму N2 - обчислення арифметичного виразу за варіантом з таблиці 1. Початкові дані для розв'язання завдання наведені в таблиці 2.
3. Дослідити роботу програми N2 в режимі налагоджувача (Debugger) та в автоматичному режимі виконання (CTRL_F9).

ВМІСТ ЗВІТУ:

1. Описати дії, які треба виконати при роботі з програмою N1 в режимі налагоджувача програм (DEBUGGER).
2. Навести блок-схему розробленої програми N2.
3. Навести текст програми N2 та результати виконання.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Запустити TURBO PASCAL.
2. Відкрити файл з назвою MYPROG1.PAS (F3 -> MYPROG1.PAS -> Enter)
3. Набрати текст програми N1 або зкопіювати її текст з файла лабораторної роботи 2 (каталог CTP) та записати файл на диск (кл.F2).

Дослідження виконання програм в режимі наладки.

5. Натиснути F9 (компіляція - Build). Якщо з'явиться повідомлення "Compile successful", а це буде свідчити про відсутність синтаксичних помилок в програмі, то виконувати наступний пункт завдання. Якщо будуть помилки, то вказівник TURBO PASCAL зупиниться на першій із них. Виправте їх та повторіть компіляцію.
6. Встановіть курсор на будь-яку букву змінної Summ та натисніть CTRL_F7 ("Add watch"), а потім ENTER. Змінна Summ з'явиться в вікні наладки "Watches", що призначено для спостереження за станом змінних програми. Аналогічно додайте в це вікно змінні X1 та X2.
7. Натисніть клавішу F7 (виконання програми по рядках). Зверніть увагу на те, що спеціальний вказівник став на початок програми (слово BEGIN) та змінні в вікні "Watches" набули різних (хаотичних) значень.

8. Почніть поспідовно натискати клавішу **F7** та уважно спостерігайте за станом змінних у вікні **Watches**. Перевірте правильність виконання дій програмою. Виконайте оператори до рядку `ch := 'W';`. Спостереження запишіть в звіт.
9. Перейдіть у вікно **Watches**, встановіть вказівник на змінну **Summ** та натисніть клавішу **Del**. Змінна **Summ** зникне. Аналогічно вилучте змінні **X1** та **X2**, а потім перейдіть у вікно з текстом вашої програми.
10. Аналогічно п.5 додайте до вікна наладки "**Watches**" змінні **ch** та **MyString**. Продовжуючи натискати кл. **F7** виконайте програму до рядку `"ClrScr;"`. Уважно спостерігайте за роботою символічних та рядкових змінних **ch** та **Summ**. Занесіть в звіт результати Ваших спостережень. Чи співпадають вони з теоретичним матеріалом ?
11. Натисніть **ALT_F5**. Ви побачите екран, з яким працює Ваша програма. Це так званий "робочий екран". Натискання будь-якої клавіші поверне Вас назад до тексту програми (в налащик).
12. Продовжуйте поспідовно по кроках (**F7**) виконувати програму та після кожного кроку спостерігайте за робочим екраном (**ALT_F5**). Визначте, як працюють оператори **WRITELN**. Зверніть увагу на те, як вивели своє значення змінні **X1** та **X2** та на їх типи.
13. Завершіть виконання програми (продовжуйте її виконувати, поки не зникне підсвічений рядок-вказівник налащика). Після цього встановіть вказівник на рядок `ch := 'Z';` та натисніть **F4** ("Go to cursor" - режим "виконувати програму до рядку ..."). Зверніть увагу на те, що попередні рядки програми автоматично виконались і на рядку `ch := 'Z';` вмикається режим наладки. Спробуйте встановити вказівник на рядок `"ClrScr;"` та знову натисніть **F4**. Що відбулось ? Чи можна знову виконувати програму по кроках (**F7**) ?
14. Не виходячи з режиму наладки, спробуйте його вимкнути за допомогою **CTRL_F2**. Що трапилось з рядком-вказівником та змінними в вікні "**Watches**" ?
15. Зробіть дослідження роботи клавіші **F8**. (Для програм, що не мають процедур, її робота аналогічна кл. **F7**).
16. В процесі роботи з налащиком встановіть курсор на змінну **X2** (рядок-вказівник налащика вже має бути нижче за неї) та натисніть **CTRL_F4**. Порівняйте результат, що з'явився в вікні "**Evaluate and Modify**" з значенням цієї змінної в вікні "**Watches**".
17. Змініть назву змінної в рядку "**Expression**" на **X1** та натисніть **Enter**. Що спостерігається ? За допомогою клавіші **TAB** встановіть курсор на рядок "**New value**" та введіть будь-яке нове значення і натисніть **Enter**.

Таким способом можна примусово змінювати значення будь-яких змінних в програмі в процесі її наладки.

18. Спробуйте в рядку "Expression" написати будь-який арифметичний вираз, наприклад $7*(134.77-67)/1000$, та натисніть **Enter**. В рядку "Result" з'явиться відповідь.
19. Запустіть програму на виконання (без наладчика за допомогою **CTRL_F9** (RUN)). Цей режим можна використовувати, коли Вам просто треба запустити програму, навіть якщо вона не зкомпільована. Тут також формуються повідомлення про помилки, якщо вони існують в програмі.

Складання лінійних програм.

20. Закрити файл з програмою MYPROG1.PAS. Користуючись діями, аналогічними п.2, відкрити файл MYPROG2.PAS.
21. Скласти програму, що реалізує розв'язування завдання за варіантом (табл.1,2), що визначає викладач. Значення змінних присвоїти за допомогою оператора присвоєння безпосередньо в програмі. Результати розрахунків вивести на екран за допомогою оператора **WRITELN**. Зробіть так, щоб програма виводила на екран повідомлення про те, що це за програма та хто її розробив, наприклад,

Програма для розрахунку за формулою $T*(Y-\cos(r))$
Розробив студент гр.1-Оа Олекса Якимчук.

22. Виконати складену програму в наладчику та, вивівши змінні у вікно **Watches**, користуючись навичками, що одержані в першій частині цієї лабораторної роботи.
23. Виконати складену програму в режимі виконання (**CTRL_F9**).
24. Записати програму під ім'ям MYPROG2.PAS (**F10** -> File -> Save as...).
25. Замість операторів присвоєння початкових значень змінних, використати оператори введення **READLN**.
26. Виконати дії аналогічні п.п.17,18.

ПИТАННЯ САМОКОНТРОЛЮ:

1. Як запустити режим наладчика в **TURBO PASCAL**?
2. Як розташувати змінні програми в вікні наладчика "Watches"?
3. Як можна вилучати змінні з вікна "Watches" ?
4. Коли починають діяти змінні у вікні "Watches" ?
5. Чим відрізняється режим **F7** (Trace into) від **F8** (Step over) ?
6. Як в процесі наладки можна подивитись на робочий екран програми?
7. Як "запустити" програму на виконання?
8. Що треба робити, якщо з'являється повідомлення про помилку?
9. Що можна одержати в режимі Evaluate\Modify (**CTRL_F4**) ?
10. Яким чином в **TURBO PASCAL** можна швидко порахувати арифметичні вирази?

Приклад: програма - N1

Додаток

USES CRT;

VAR Summ, X1 : real;
X2 : integer;
ch : char;
MyString : string;

BEGIN

Summ := 0; X1 := 0; X2 := 0;

X1 := 3.141592;

X2 := 2;

Summ := X1*2;

Summ := Summ + X2;

ch := 'W';

MyString := '';

ch := 'Z';

ch := '.';

MyString := 'Олекса Петренко - ';

MyString := MyString + 'народився в м.Тульчині'+ch;

ClrScr;

Writeln('Змінна Summ після обчислень дорівнює : ', Summ);

Writeln('а аргументи X1 та X2 були : X1=', X1, ' X2=', X2);

Writeln('В рядковій змінній MyString записано : ', MyString);

Readln; {цей оператор очікує натискання клавіші Enter}

END.

ВАРІАНТИ ЗАВДАНЬ

Таблиця 1

Номер варіанту	З а в д а н н я
1	$y = \frac{\lg^2(ax+b)}{\sqrt[3]{ax+b+tg^2(x^2)}} \sin\left(\frac{bx}{a}\right)$
2	$y = \frac{a - \sin(2x) + b \cos(cx)}{\sqrt{2x+cx^2} - \lg^2\left(\frac{3x}{b}\right)}$
3	$y = \frac{\cos(x^2+3) + tg(3x-b)^{1.2}}{\sqrt{x^2+3} + \lg(3x-a) - e^{0.3x} }$
4	$y = ax + \frac{x - b \cos(x) \sqrt{ 3x } + 2}{\cos(x^2/(x+3)) + c^2 \lg(x^2)}$
5	$y = \frac{x^4+1}{4} + \frac{\cos(ax/b) e^{c \sin(x)}}{atg(x^2/2) + e^{-0.2x}}$
6	$y = \frac{ \sin^2(x) - x^2/2 - x^{1.2} ^{1.2}}{\cos x \sin ax + tg^2 bx + c}$
7	$y = \sqrt[3]{b + \ln ax \cos((x^2+c)/(x+b))} \sin x+1 $

8	$y = \frac{3 \cos(x) \left \operatorname{tg}((x^2 + a)/b) \right + 1}{\sqrt[3]{c + x^2 + e^{-0.13x}}}$
9	$y = \frac{\operatorname{tg}(x+a)e^{\cos x + \ln x}}{\sqrt[3]{\ln 3x + b + (x+c)^2}}$
10	$y = \frac{ac^{bx+1} + \sin^2 x^2}{ \cos^2 x \ln x + e^{\cos x + 1} + c}$
11	$y = \frac{\sqrt[3]{\sin x^2 + a} \operatorname{ctg}(bx)}{\operatorname{tg}(cx^2/2) + \cos^2(ax/2)}$
12	$y = \frac{\cos^2(x^2 + a) + x}{\sqrt[3]{e^{-x} + \operatorname{ctg}(x^2 + b)}} + \frac{1}{x^3 + \sin cx }$
13	$y = \frac{(\sin((x^2 + a)/(x + b)) + 3\operatorname{tg}(x^2/(c + \cos x)))^{1.2}}{\sqrt[5]{x^2 + 1} + e^{-0.3x}}$
14	$y = \frac{(\sin^2(x/(x+a)) + \sqrt[3]{3 \cos x \operatorname{tg}(bx/c)})^a}{\sqrt[5]{x^2 + a^2} + e^{x+a}}$
15	$y = \frac{(\sqrt{ \sin^2 dx + \cos x^2 } + e^{\cos x})^2}{b \operatorname{tg} x + c \operatorname{ctg} x(x/2)}$
16	$y = \frac{e^{-x} \sqrt{a^2 x + b^2 \cos(cx^2)}}{\sin x + \cos x + e^{x-a}}$
17	$y = \frac{\sqrt{ \sin^2 x + \cos x^2 }}{\sin x^2 + \cos x}$
18	$y = \frac{x^2 + 1 + 5x + \cos x}{a^{x+1} + b^{\cos x + c}}$
19	$y = \frac{\sqrt[3]{ \sin x^2 + \cos^2 x } + \operatorname{ctg}^3 x + c^x}{b^{-0.3x} \sqrt{ \sin x } + \sin(\cos^2 ax)}$
20	$y = \frac{a^{0.2x} + b^2 \cos^2 x + \sqrt[3]{x^3 + cx^2}}{\cos x + \operatorname{tg}((x^2 + a)/(c + \sin x))}$

Значення параметрів А, В, С, Х

Таблиця 2

Вар.	А	В	С	Х
1	2.1	3.4	-	Pi/6+0.37
2	-1.0	2.4	4.3	Pi/2+4.1
3	1.2	0.3	-	Pi/2
4	-4.2	-0.1	1.3	-pi/4
5	1.4	2.0	-0.2	Pi/3

6	0.8	0.32	2.3	$\pi/6$
7	1.21	2.43	-4.7	$\pi/8$
8	3.3	1.7	2.16	$\pi/12$
9	0.44	1.67	-0.73	$\pi/5$
10	2.33	-1.1	1.83	$-\pi/8-0.4$
11	1.7	0.79	0.62	$\pi/9$
12	0.73	-0.34	0.49	$2\pi/5$
13	1.33	0.56	1.4	$\pi/5$
14	1.1	-1.86	3.14	$\pi/3$
15	1.2	1.69	0.62	$2\pi/13$
16	0.82	6.73	0.57	$-3\pi/25$
17	2.41	-0.29	1.48	$-4\pi/33$
18	1.36	0.87	1.54	$-2\pi/17$
19	0.81	1.28	0.68	$2\pi/21$
20	1.4	-0.65	1.99	$\pi/6$

ПОЯСНЕННЯ до деяких режимів оболонки TURBO-PASCAL:

Compile successful	успішна компіляція (помилки немає).
Debug	наладчик програми
Evaluate/Modify	дивитись або коректувати значення змінної або виразу
Expression	Вираз
New value	нова величина
Result	Результат
Watches	Спостереження

ЛАБОРАТОРНА РОБОТА N 3

ТЕМА: ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ РОЗГАЛУЖЕНИХ АЛГОРИТМІВ

МЕТА РОБОТИ: Придбати навички проектування та програмної реалізації алгоритмів з використанням операторів умовного переходу (розгалуження).

ЗАВДАННЯ:

1. Скласти алгоритм програми за варіантом з табл.1, номер якого визначає викладач.
2. Скласти програму відповідно до розробленого алгоритму.
3. Набрати програму в TURBO PASCAL і та відпрацювати її.

ВМІСТ ЗВІТУ:

1. Навести блок-схему алгоритма та її описання.
2. Навести текст розробленої програми.
3. Навести результати виконання програми.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Розробити блок-схему програми відповідно до завдання. Програма обов'язково має виводити на екран свою назву та прізвище студента, що її розробив. Всі початкові, до розрахунків, дані мають вводиться з клавіатури комп'ютера. Перед кожним введенням програма має повідомляти, які саме дані потрібно ввести з клавіатури.
2. Розробити текст програми на папері згідно з алгоритмом.
3. Запустити TURBO PASCAL.
4. Відкрити файл з назвою IF_THEN.PAS (F3 -> IF_THEN.PAS -> Enter).
5. Набрати текст розробленої програми та записати файл з результатами (кл.F2).
6. Відпрацювати програму (знайти та виправити помилки), результати показати викладачеві.

ВАРІАНТИ ЗАВДАНЬ

Таблиця 1

Номер варіанту	З а в д а н н я
1.	$Y = \begin{cases} \cos^2 x^2 / \sqrt{x^2 + 4,2} & \text{якщо } X \leq 2; \\ -0,5 * \lg x^2, & \text{якщо } X > 5; \end{cases}$ $x = \sqrt{((ab+b)/\ln b)}; a=12,3; b=4,1$
2.	$Y = \begin{cases} 4,73 * \lg(x^2+a) / \sqrt{x^2+3,7x^4}, & \text{якщо } X \leq -2,6; \\ x^2 * \sin 2x, & \text{якщо } X > 0,2; \end{cases}$ $x = 9,3 * \sin(a+b/2a); a=1,3; b=3,4$
3.	$Y = \begin{cases} 0,34 * \cos(x^2/2), & \text{якщо } X \leq 2,3; \\ \lg^2(\sin x + 0,2), & \text{якщо } X > 3,7; \end{cases}$ $x = 12,3 * \cos(a/(\sin b + a^2)); a=3,9; b=1,23$
4.	$Y = \begin{cases} \sin 2x / (1 + (x^2+1)/(1+e^{-0,32x})), & \text{якщо } X \leq 1,3; \\ -0,29x + \cos^2 2,7x^2, & \text{якщо } X > 2,1; \end{cases}$ $x = 8,23 * \lg a^2 - b^2 ; a=6,2; b=7,4$
5.	$Y = \begin{cases} (\cos^3 \lg(x +2,13)), & \text{якщо } X < 0,97; \\ 2,08x^2 + \ln^3 2x / (1+x^2), & \text{якщо } X > 1,72; \end{cases}$ $x = a^2 + 4b * \sin(\cos(a/(2b))); a=1,6; b=1,9$
6.	$Y = \begin{cases} \sqrt{(2x^2+1)/e^{-0,62x} + 7,2} & \text{якщо } X \leq 2,32; \\ \sin^2 x * \ln 0,34x, & \text{якщо } X > 4,09; \end{cases}$ $x = \ln a / (\lg b + \ln^2 a); a=4,1; b=2,7$
7.	$Y = \begin{cases} \ln^2(\cos(2x+1,3) + 0,2), & \text{якщо } X < 1,8; \\ 2,3 * \cos x^2 / (x+3,7x^2), & \text{якщо } X \geq 2,7; \end{cases}$ $x = -0,19a * \ln b; a=4,1; b=6,3$

8.
$$Y = \begin{cases} \cos(\ln(1,3 + |2,72x|)), & \text{якщо } X \leq -2,1; \\ \sqrt{(e^{0,29x} + 5,1x^2)/(\cos^2 x + 4,1)}, & \text{якщо } X > 1,3; \end{cases}$$

 $x = 4,3 \cdot \sin 3a \cdot \cos(b/(a+b)); a = 1,97; b = 2,61$
-
9.
$$Y = \begin{cases} 4,7 \cdot \cos 0,2x / (\sin 2x + \ln(x^2 + 1)), & \text{якщо } X \leq 1,2; \\ 8,3 \cdot \cos x^2 + 3x, & \text{якщо } X > 5,6; \end{cases}$$

 $x = 7,3 \cdot \ln |a/b| e^{-0,3}; a = 0,8; b = -0,45$
-
10.
$$Y = \begin{cases} 9,74 \cdot x^2 + \cos 6,4x, & \text{якщо } X < -3,1; \\ \sqrt{(4,3x^2 + 2)/(1 + \ln(x + 0,3))}, & \text{якщо } X > -1,37; \end{cases}$$

 $x = 5,3 \cdot \sin(b/(\cos a + 2)); a = 1,6; b = 4,9$
-
11.
$$Y = \begin{cases} \sqrt{|x| + 0,3 \cdot \ln |x|}, & \text{якщо } X \leq -5,4; \\ \cos x^2 / -0,1x + 3,7x^2, & \text{якщо } X > -0,9; \end{cases}$$

 $x = a/(a-b) \cdot \ln^2(ab) \cdot \cos(a/b); a = 7,3; b = 2,7$
-
12.
$$Y = \begin{cases} 5,13 \cdot \ln x^2 / 2,1x^2 + 3,7, & \text{якщо } X \leq -1,31; \\ x^3 \cdot \cos 0,7x, & \text{якщо } X > 0,8; \end{cases}$$

 $x = -6,3 \cdot \cos(3 \cdot \sin a / (1,3 + \ln b)); a = 1,7; b = 1,3$
-
13.
$$Y = \begin{cases} 0,72 \cdot \sin(x^3/3), & \text{якщо } X < 1,7; \\ \cos^2(\sin x + 0,6), & \text{якщо } X > 4,1; \end{cases}$$

 $x = (a^2 - 3b)/(a+b); a = 5,7; b = 2,9$
-
14.
$$Y = \begin{cases} \operatorname{tg} 3x + e^{-0,23x}, & \text{якщо } X < -0,35; \\ \lg 2x / (1,7 + 0,73x), & \text{якщо } X > 1,92; \end{cases}$$

 $x = \sqrt{\sin^2 a + 8,7 \cos b}; a = 1,9; b = 3,7$
-
15.
$$Y = \begin{cases} 6,3x + \cos^2 2x / (1 + x^2), & \text{якщо } X \leq 0,37; \\ \cos^2 \lg(x^2 + 0,29), & \text{якщо } X > 0,62; \end{cases}$$

 $x = -3,7 \cdot \cos(a^2 + b^2/(a+b)); a = 1,3; b = 0,9$

$$16. \quad Y = \begin{cases} \sqrt{2x^2 + \cos x^3}, & \text{якщо } X < 0, \\ \ln^2 0,62x / (1 + x^2 \sin^2 x), & \text{якщо } X > 3,1; \end{cases}$$

$x = 12,3 \cdot \cos((3a + b^2) / (2a^2 + 3b)); a = 1,79; b = 2,33$

$$17. \quad Y = \begin{cases} \cos(\ln|x| / (2x + 1,3)), & \text{якщо } X < -1,6; \\ 2,3 \cdot \cos^2 x / 3,7x^3 + x, & \text{якщо } X > 3,9; \end{cases}$$

$x = -4,7 \cdot \cos|(a-b) / (a+b)|; a = 1,34; b = 6,06$

$$18. \quad Y = \begin{cases} \lg^2|3,77x|, & \text{якщо } X < -1,7; \\ \sqrt{\cos x^2 + 5,9}, & \text{якщо } X \geq -0,3; \end{cases}$$

$x = \ln|(a^2 - ab) / (2b - a)|; a = 2,3; b = 3,6$

$$19. \quad Y = \begin{cases} 8,3x \cdot \cos x + 3, & \text{якщо } X < 0,1; \\ \sin 2x + \lg(x^2 + 1), & \text{якщо } X \geq 1,3; \end{cases}$$

$x = (a + \sin^2 a) / (|\cos b| + \ln(a/b)); a = 7,9; b = 1,36$

$$20. \quad Y = \begin{cases} \sqrt{2,3x^2 + |\cos x|} / (1 + \ln^2 x^2), & \text{якщо } X \leq -1,9; \\ 6,63 - \operatorname{tg}^3(x/3), & \text{якщо } X > 0; \end{cases}$$

$x = 3,1 \cdot \sin(b^2 - a / (ab)); a = 2,8; b = 3,2$

ПИТАННЯ САМОКОНТРОЛЮ:

1. Який оператор використовується для безумовного переходу?
2. Які правила використання міток в програмі?
3. Який оператор використовується для умовного переходу?
4. Що треба робити, якщо існує складна умова? Як в цьому випадку робиться запис складної умови?
5. Що робити, якщо при дотриманні умови треба виконувати декілька операторів?
6. Для чого використовується службове слово **ELSE**? Що треба робити, якщо після його спрацювання треба виконувати декілька операторів?
7. Які особливості формату операторів **WRITE** або **WRITELN**?
8. Як працює оператор **READ**?

ЛАБОРАТОРНА РОБОТА N 4

ТЕМА : СКЛАДАННЯ ЦИКЛІЧНИХ ПРОГРАМ. ТАБУЛЮВАННЯ ФУНКЦІЙ.

МЕТА РОБОТИ: Навчитись розробляти алгоритми та програми, за допомогою яких можна проводити табуляцію математичних функцій.

ЗАВДАННЯ :

1. Розробити алгоритм програми згідно з одержаним варіантом. (Номер варіанту визначається за номером студента в журналі викладача).
2. Розробити програму згідно алгоритму, створивши цикл за допомогою операторів **REPEAT-UNTIL** (для непарних варіантів) та **WHILE-DO** (для парних варіантів). Кількість точок табуляції - 20 шт.
3. Набрати програму та назвати файл з нею TABLE.
4. Виконати програму спочатку за допомогою налагодчика (DEBUGGER'a), а потім в режимі прогону. Результати показати викладачеві.
5. Побудувати графік функції (по точках) в звіті.

ВМІСТ ЗВІТУ:

1. Навести розроблені блок-схему та програму згідно з завданням до лабораторної роботи (таблиця 1).
2. За результатами табулювання функції побудувати графік.
3. Навести висновки за результатами роботи.

ВАРІАНТИ ЗАВДАНЬ :

Таблиця 1

№ - варіанту	Функція	кордон-1	кордон-2	кордон-3
1	$y = \log_2(x^2 + 2x + 3)$	-3; 0	0; 3	-3; 3
2	$y = 2x \cdot e^{-3x}$	-2; 2.2	0; 3	-2; 3
3	$y = 4 \cos x / (1 + x^2)$	$-2\pi; 2\pi$	$-\pi/2; \pi/2$	0; 2π
4	$y = \cos x / \sqrt{x}$	0; 2π	$\pi; 2\pi$	0.5; π
5	$y = ((2x - 1)/(x^2 - x + 1))^3$	-5; 0	0; 5	-5; 5
6	$y = 4 / (x^2 - x + 1)$	-4; 0.5	0.5; 4	-4; 4
7	$y = (x - 3) / (x - 4) $	-4; 3	3; 3.95	4.1; 8
8	$y = (1 - x^2)(2 + x)$	-3; 1	-2; 0	-1; 1
9	$y = \arccos((1 - x^2)/(1 + x^2))$	-3; 1	-2; 0	-1; 1
10	$y = \cos(\log_2 x)$	0.1; 1	1; 4	0.1; 4
11	$y = \cos 2^x$	-4; 0	0; 4	-4; 4

№ - варіанту	Функція	кордон-1	кордон-2	кордон-3
12	$Y = 2x \sin(\pi x/4)$	$-3\pi; \pi$	$-\pi; \pi$	$\pi; 2\pi$
13	$y = \sqrt[3]{x^2} - x$	$-3; 0$	$0; 3$	$-3; 3$
14	$y = 1/2(\sin x - \sin x)$	$-2\pi; 2\pi$	$-\pi/2; \pi/2$	$-\pi; \pi$
15	$y = (2x+3)/(1+x+x^2)$	$-3; 3$	$-3; 0$	$0; 3$
16	$y = e^{-x} \cdot \sin x$	$-\pi; \pi/2$	$0; 2\pi$	$-2\pi; 2\pi$
17	$y = 2x \cdot e^{-3x}$	$-2; 2,5$	$0; 3$	$-2; 3$
18	$y = \arctg x + 2^x$	$-2\pi; 2\pi$	$0; 3$	$-6; 0$
19	$y = x^2 - x^4$	$-3; 3$	$-5; 0$	$0; 5$
20	$Y = \arcsin(\sin x)$	$-\pi; \pi/2$	$\pi/2; 2\pi$	$-2\pi; 2\pi$

ІТАННЯ САМОКОНТРОЛЮ:

Що таке табулювання функції ?

Що таке крок змінення аргументу та як його обрахувати?

Як працюють оператори циклів ?

Які оператори циклів Ви знаєте ?

За допомогою якого оператора можна призупинити виконання програми ?

Який оператор дозволяє подати звуковий сигнал ?

Як працює оператор **WRITE** та чим він відрізняється від оператору **WRITELN** ?

Для чого в операторах **WRITE** використовується символ $< : >$?

Як можна було б трохи спростити Вашу програму ?

одаток

**ПРИКЛАД НАПИСАННЯ ПРОГРАМ ДЛЯ ТАБУЛЯЦІЇ ФУНКЦІЇ З
ВИКОРИСТАННЯМ ОПЕРАТОРІВ ЦИКЛУ REPEAT-UNTIL:**

```
Uses Crt;
```

```
var X,Y,step : real;
```

```
BEGIN
```

```
  ClrScr; {почистимо робочий екран програми та виведемо  
  CopyRight}
```

```
  writeln('Програма для обрахування функції ...');
```

```
  writeln('Виконав студент гр. 10А Грищенко М.Р.');
```

```
  writeln('=====');
```

```
  step := (6-1)/20; {порахуємо крок зміни аргументу}
```

```
  x := 1; {даємо аргументові початкове значення}
```



```

repeat
  y := Cos(12.89-X)/ABS(22.89-X*X)/(X*X*X*X); {обрахуємо
функцію}
  writeln('X=', X:3:3, ' Y=', Y:5:3); {та виведемо її
значення}
  x := x + step; {надамо аргументові наступне значення}
until x >= 6; {закінчимо цикл коли аргумент вийде за
встановлену межу }
write('^G'); {програма закінчилась, увімкнемо звуковий
сигнал}
readln; {цей оператор буде очікувати, поки не буде
натиснута клавіша ENTER}

```

END.

ПРИКЛАД НАПИСАННЯ ПРОГРАМИ ДЛЯ ТАБУЛЯЦІЇ ФУНКЦІЇ З ВИКОРИСТАННЯМ ОПЕРАТОРІВ ЦИКЛУ WHILE-DO:

Uses Crt;

var X,Y,step : real;

BEGIN

ClrScr; {почистимо робочий екран програми та виведемо
CopyRight}

writeln('Програма для обрахування функції ...');

writeln('Виконав студент гр. 10А Грищенко М.Р.');

writeln('=====');

step := (6-1)/20; {порахуємо крок зміни аргументу}

x := 1; {даємо аргументові початкове значення}

While x < 6 do begin {закінчимо цикл коли аргумент вийде за
встановлену межу }

y := Cos(2.89-X*X)/Sin(2.89-X*X)/(X*X*X*X); {обрахуємо функцію}

writeln('X=', X:3:3, ' Y=', Y:5:3); {та виведемо її значення}

x := x + step; {надамо аргументові наступне значення}

end;

write('^G'); {програма закінчилась, увімкнемо звуковий сигнал}

readln; {цей оператор буде очікувати, поки не буде
натиснута клавіша ENTER}

END.

ЛАБОРАТОРНА РОБОТА N 5

ТЕМА: **NORTON COMMANDER** та його режими

МЕТА РОБОТИ: Ознайомлення з базовими режимами роботи **NORTON COMMANDER**. Вивчення дій основних операцій з файлами.

ЗАВДАННЯ:

1. Навчитись користуватись підказками NC.
2. Навчитись користуватись меню.
3. Навчитись користуватись режимами перегляду (VIEW) та редагування файлів.
4. Навчитись створювати каталоги та працювати з ними.
5. Навчитись вмикати\вимикати робочі панелі, обирати робочий диск та пересуватись між панелями.
6. Навчитись копіювати, пересувати та перейменовувати файли.
7. Навчитись знищувати файли.
8. Ознайомитись з режимами робочих панелей та з методами розташування файлів на них.
9. Навчитись працювати з атрибутами файлів.
10. Навчитись запускати інші програми за допомогою NC.

ВМІСТ ЗВІТУ:

1. Звіт повинен мати відповіді на всі пункти завдання до лабораторної роботи, а також детальне описання дій по виконанню тієї чи іншої операції.
2. Завдання, що супроводжуються певними текстовими або іншими повідомленнями NC, повинні їх вміщувати.
3. В звіті мають бути наведені висновки за отриманими результатами.

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Увімкнути комп'ютер.
2. Зареєструватись в комп'ютерній мережі (Цей пункт виконується при роботі на мережевих робочих місцях).
3. Натиснути клавішу **F1** і, за допомогою клавішей пересування курсору та клавіші **ENTER**, зайти в режими підказок NC. Визначити, яким чином можна пересуватись вздовж тексту. Дослідити, що відбувається при натисканні клавіші **ESC**.
4. Вийти з підказок та увімкнути режим службового меню (клавіша **F9**). Пересуваючись вздовж нього та викликаючи підказки за допомогою клавіші **F1**, визначити особливість цього режиму підказок.
5. Визначити, які дії викликає натискання комбінацій **CTRL_F1**, **CTRL_F2**.

6. Натиснути комбінацію **ALT_F1** (**ALT_F2**) послідовно вибрати логічні диски що спостерігаються (окрім A та B) і, натиснувши після цього клавішу **ENTER**, записати в звіт результати спостережень.

7. Увімкнути обидві панелі та визначити дію клавіші **TAB**.

8. Поставити курсор на панель з назвою **Z:** і після цього встановити його на файл **TS.BAT**. Натиснути клавішу **F5** і в вікні, що з'явилося, набрати **AUTO.BAT**. Після цього натиснути **ENTER**. Встановити вказівник на файл **AUTO.BAT** і натиснути клавішу **F3**. Записати спостереження в звіт. Вийти з цього режиму (кл. **ESC** або **F10**).

9. На протилежній панелі увімкнути також диск **Z:** (див. п.6). Натиснути клавішу **F7**. В рамці, що з'явилась набрати рік свого народження та натиснути клавішу **ENTER**. Поставити вказівник на створений каталог та натиснути **ENTER**. Занотувати результати спостережень.

10. За допомогою клавіші **TAB** поставити вказівник на протилежну панель, встановити його на файл **AUTO.BAT** та натиснути кл. **F5**. Записати в звіт отримане повідомлення. Натиснути кл. **ENTER**. Записати в звіт, які Ви одержали результати.

11. Вимкнути панель, *куди* перед цим було скопійовано файл (для лівої панелі це комбінація клавіш **CTRL_F1**, а для правої - **CTRL_F2**). Встановити вказівник на файл **AUTO.BAT** та натиснути **F5**. У вікні, що з'явилося, набрати латинськими буквами своє ім'я, наприклад **VICTOR**, та натисніть **ENTER**. Запишіть спостереження в звіт.

12. Встановіть вказівник на файл **VICTOR** (або з іншим ім'ям, що Ви ввели) та натисніть клавішу **F4**. Занотуйте різницю дії клавіші **F4** від клавіші **F3**. Визначте, які зміни відбуваються при натисканні клавіш керування курсором. В будь-якому місці файлу спробуйте набрати свої прізвище, ім'я та рік народження. Натисніть клавішу **ESC** та запишіть, що Ви бачите. Натисніть **ENTER**. Порівняйте розмір, дату і час свого файлу з файлом **AUTO.BAT**. Чому це відбулось ? Перевірте результат за допомогою клавіші **F3**.

13. Увімкнути протилежну панель та натисніть клавіші **F6** а потім **ENTER**. Запишіть, що при цьому відбулось і яка різниця в одержаному результаті порівняно з п.10 завдання.

14. Встановіть вказівник на файл з вашим ім'ям (скористайтесь клавішами **TAB** та кл. пересування вказівника "курсора"), вимкніть протилежну панель та натисніть клавішу **F6**. В вікно, що з'явилося запишіть своє місце народження латинськими буквами та натисніть **ENTER**. Запишіть спостереження.

15. Встановіть вказівник на позначку « .. » (**КАТАЛОГ**) та натисніть **ENTER**. Повторно натисніть **ENTER**. Що відбулося? Занотуйте, яким чином можна входити до каталогів і як можна з них виходити ?

16. Встановіть вказівник на файл, що був створений в п.14 та натисніть клавішу **F8**. Занесіть в звіт те, що Ви побачили на екрані, та натисніть **ENTER**. Занотуйте результати.

17. Увімкніть диск **Z:** (якщо ви працюєте в мережевому комп'ютері) або диск **C:** (див. п.6) і встановіть вказівник так, щоб бачити максимум файлів.

18. Натисніть кл. **F9**, потім **ENTER**. Виберіть опцію "**Brief**" ("Коротка") та натисніть **ENTER**. Як змінилась робоча панель? Занесіть її вигляд у звіт.

19. Виконуючи дії аналогічні п.17 дослідити режими "**Full**" ("Повна"), "**Info**" ("Інформація про диск та оперативну пам'ять"), "**Tree**" ("Дерево каталогів"), а також різні режими сортування (розташування) файлів на панелі - "Ім'я" - **CTRL_F3** ("За ім'ям"), "Тип (расширение)" - **CTRL_F4** ("За розширенням"), "Время" - **CTRL_F5** ("За часом створення"), "Размер" - **CTRL_F6** ("За розміром"), "Без упорядкування" - **CTRL_F7** ("Несортовані файли"). Всі одержані результати занесіть в звіт.

20. Натисніть кл. **F9**, потім **ENTER** і за допомогою клавіші **<-** або **->**, продивіться всі меню. Результати запишіть в звіт у вигляді схеми.

21. Встановіть вказівник на перший файл та послідовно натискайте клавішу **INS** (Insert). При цьому файли помічаються, про що свідчить або змінення їх кольору або встановлення спеціальної позначки. Визначити, що відбувається при спробі повторної позначки файлів.

22. Зробіть дослідження, яку реакцію викликає натискання сірих клавіш **<+>** або **<->**, а також **<*>**, що розташовані на малій клавіатурі з підтвердженням клавішею **ENTER**.

23. Позначити декілька файлів та скопіювати їх до себе в робочий каталог, користуючись діями, аналогічними п.5. Після цього позначте декілька з них і спробуйте знищити відмічені файли за допомогою клавіші **F8**. Занотуйте результати.

24. Створіть декілька послідовно вкладених каталогів (аналогічно п.9) та запишіть в них декілька файлів (аналогічно п.10). Вийдіть із створених каталогів та спробуйте їх знищити за допомогою кл. **F8**. Занесіть в звіт ваші спостереження.

25. Встановіть вказівник на один з файлів, натисніть клавішу **F9**, після цього зайдіть в меню «**Files**» (**ENTER**) => «**Attributes**» (**ENTER**). Запишіть в звіт атрибути цього файлу. За допомогою клавіші **SPACE** ("Пропуск") увімкніть атрибут «**ReadOnly**» та підтвердіть свої дії клавішею **ENTER** стільки разів, поки не закрийся меню зміни атрибутів. Спробуйте знищити цей файл. Занотуйте реакцію Norton Commander в цьому випадку.

26. Для любого іншого файлу встановіть атрибути **System**. Занотуйте, як такі файли відтворює NC на своїй панелі?

27. Знайти файл з програмою (що має розширення **.EXE** або **.COM**) за вказівкою викладача та запустити його на виконання за допомогою клавіші **ENTER**. Вийти з програми, що була запущена та визначити стан Norton Commander.

28. Увімкніть режим пошуку файлів, натиснувши клавіші **ALT_F7**. В верхнє вікно введіть назву файлу (файлів) (за вказівкою викладача), який має відшукати NC. Запустіть пошук. Зробіть тіж самі дії, але відшукайте ряд файлів, що відповідають загальній ознаці, наприклад, шукати всі файли, що мають розширення **TXT**. Для цього у вікно пошуку занесіть ***.TXT**. (* означає "BCI"). Продивіться одержаний список. Що відбувається, якщо натиснути в списку **ENTER**? Результати занесіть в звіт.

ПИТАННЯ САМОКОНТРОЛЮ:

1. Що таке файл? Як спостерігається файл на панелі NC?
2. Що таке каталог? Як спостерігається каталог на панелі NC?
3. Для чого призначена програма NORTON COMMANDER?
4. Які основні режими роботи має NORTON COMMANDER?
5. Яку основну функцію в програмі NORTON COMMANDER виконують клавіші **ENTER** та **ESC**?
6. Які дії має робити користувач для :
 - роботи з підказками NC?
 - роботи з меню програм NC?
 - перегляду та редагування файлів?
 - знищення файлів?
 - копіювання, пересування або знищення файлів?
7. Які операції дозволяють позначити файли (або відмінити позначення) і для чого це робиться?
8. Чи можна знищити каталог? Яке попередження одержить користувач, якщо в цьому каталозі будуть знаходитись файли?
9. Як можна змінити режим роботи панелей NC?
10. Чи можна одночасно спостерігати на двох панелях одну і ту саму інформацію?
11. Які режими робочих панелей NC існують?
12. Чи можна за допомогою NC запускати інші програми?
13. Що таке атрибути файлів і як можна їх змінити?
14. Яким чином можна шукати файли за допомогою NC?

ЛАБОРАТОРНА РОБОТА N 6

ТЕМА : ОБРОБКА МАСИВІВ

МЕТА РОБОТИ: Навчитись розробляти алгоритми та програми, які виконують дії над одновимірними та двовимірними масивами.

ЗАВДАННЯ 1: Створити масив з 10 дійсних чисел (типу `real`) шляхом введення значень з клавіатури. Для всіх варіантів в отриманому масиві знайти найбільший(для парних варіантів) або найменший(для непарних варіантів) елемент та його індекс (номер) в масиві. Вивести на екран одновимірний масив у вигляді рядка.

ЗАВДАННЯ 2: Створити двовимірний масив з довільних чисел, розмірність - за варіантом завдання по таблиці 1, виконати вказане завдання. Вивести на екран двовимірний масив у вигляді матриці, а також інші результати по роботі з масивом. При виведенні на екран мають вказуватись значення та індекси елементів масиву, наприклад, $A[3,1]=-8.4567$.

ВМІСТ ЗВІТУ :

1. Навести розроблені блок-схеми та програми згідно з завданнями до лабораторної роботи (таблиця 1).
2. Навести висновки за результатами роботи.

Таблиця 1

ВАРІАНТИ ЗАВДАНЬ :

- | |
|---|
| 1. Знайти середнє арифметичне елементів масиву $A(6,4)$, відкинувши найбільший та найменший елементи. |
| 2. В масиві $A(6,6)$ всі від'ємні елементи, які знаходяться вище головної діагоналі, піднести до квадрату. |
| 3. Знайти кількість нульових і суму від'ємних елементів масиву $A(4,5)$. |
| 4. В масиві $X(6,6)$ поміняти місцями елементи, які є симетричними відносно головної діагоналі. |
| 5. Знайти найменший елемент масиву $X(7,5)$ і записати нулі в ті рядок і колонку, на перетині яких знаходиться найменший елемент. |
| 6. Обчислити суму елементів масиву $A(6,6)$, які знаходяться над головною діагоналлю. |

7. Обчислити добуток елементів масиву $H(5,5)$, які знаходяться на головній діагоналі.

8. Переписати елементи головної діагоналі масиву $A(6,6)$ в одновимірний масив B .

9. В масиві $X(6,5)$ знайти суму елементів 1-го, 3-го і 6-го рядків і розділити на суму елементів 3-ї колонки.

10. В масиві $A(5,6)$ знайти середнє арифметичне елементів для кожного рядка, а потім суму середніх арифметичних.

11. Обчислити найбільше і найменше значення в стовпцях двовимірного масиву $P(4,4)$.

12. Обчислити середнє арифметичне додатніх елементів масиву $X(4,5)$, які задовольняють умову $X_{ij} > 5$.

13. В масиві $Y(5,5)$ елементи 1-ї, 2-ї та 5-ї колонки упорядкувати по спаданню.

14. В масиві $A(5,6)$ знайти середнє арифметичне найбільшого і найменшого елементів.

15. Вивести на екран монітора всі додатні елементи головної діагоналі масива $A(6,6)$.

16. В масиві $X(6,6)$ всі додатні елементи, які знаходяться під головною діагоналлю, замінити нулями.

17. В масиві $X(6,4)$ поміняти місцями елементи 1-го і 6-го, 2-го і 5-го, 3-го і 4-го рядків.

18. Зформувати масив M , елементи якого визначають кількість додатніх елементів в кожному стовпчику масиву $A(5,7)$.

19. В масиві $X(6,4)$ елементи 1-го, 4-го і 6-го рядків, які більші числа A , піднести до квадрату (A - остання цифра в заліковій книжці).

20. В масиві $Y(4,6)$ упорядкувати по зростанню елементи 3-ї, 4-ї та 6-ї колонок.

ДОДАТКОВІ ВІДОМОСТІ :

Приклад заповнення одновимірного масиву А випадковими числами :

```
Randomize; {включимо генератор випадкових чисел}
for i:=1 to 10 do begin
  a[i] := Random(1000);
  a[i] := a[i]/2-250;
end;
```

1. Для визначення, чи має число дробову частину можна використовувати функцію **TRUNC(x)**, що відкидає дробову частину числа, тобто :

```
TRUNC(1.56) <> 1.56;
TRUNC(45)    = 45;
```

отже, щоб визначити, чи є, наприклад, елемент **G[i]** дробовим, треба записати :

```
if TRUNC( G[i] ) <> G[i] then ...
```

2. Для роботи з елементами двовимірного масиву треба скористатись подвійним вкладеним циклом, в якому змінні циклів є індексами масивів :

```
for i := 1 to 41 do
  for j := 1 to 14 do begin
    {тут пишеться дія, що має виконуватись з поточним
    елементом двовимірного масиву, наприклад,
    if S[i,j] > max then ... }
  end;
```

3. ЗРУЧНО ! Якщо при описуванні масивів (аналогічно і для змінних) надати їх елементам одразу початкові значення,

```
CONST
T : array[1..3] of real = (1.56, -7.456, -0.003);
Y : array[1..3,1..2] of integer = ((-1,7), (34,-65),
(86,-11));,
```

то тоді не треба в програмі їх вводити з клавіатури.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ:

1. Що таке масив даних ?
2. Яка структура даних утворює масив?
3. Які оператори використовуються для описання масивів ?
4. Які можуть існувати типи масивів ?
5. Як позначаються масиви в мові TURBO-PASCAL?
6. Що таке індекс і для чого він використовується ?
7. Чи можна в якості індекса масиву використовувати змінну ?
8. Які типи змінних можна використовувати в якості індексів масивів ?
9. Яким шляхом здійснюється доступ до елементів масиву ?
10. Чому при роботі з масивами зручно використовувати цикли?
11. Наведіть приклади з життя, в яких можна було б використовувати масиви.

ЛАБОРАТОРНА РОБОТА N 7

ТЕМА : СОРТУВАННЯ ЕЛЕМЕНТІВ В МАСИВАХ

МЕТА РОБОТИ: Навчитись розробляти алгоритми та програми, за допомогою яких виконується сортування елементів в масивах за певними правилами.

ЗАВДАННЯ 1. Створити масив з 10 цілих чисел (типу **INTEGER**). Масив заповнити за допомогою генератора випадкових чисел **RANDOM** та допоміжної процедури **RANDOMIZE**. Виконати над елементами масиву дії по сортуванню його елементів згідно з одержаним варіантом (Номер варіанту визначається за номером студента в журналі викладача).

ЗАВДАННЯ 2. Провести спостереження процесу сортування за допомогою налагодчика в режимі **DEBUGGER**. Результати показати викладачеві.

ЗАВДАННЯ 3. Вивести на екран відсортований масив у вигляді стовпчика, де в кожному рядку виводиться назва, індекс та значення елементу масива. Наприклад,

`U[3]=-67 і т.д.`

ВМІСТ ЗВІТУ:

1. Навести розроблені блок-схеми та програми згідно з завданнями 1-3 до лабораторної роботи.
2. Навести висновки за результатами роботи.

ВАРІАНТИ ЗАВДАНЬ :

-
1. Виконати сортування елементів масиву за зростанням, використовуючи *"бульбашковий метод"*.
-
2. Виконати сортування елементів масиву за зменшенням, використовуючи *"бульбашковий метод"*.
-
3. Виконати сортування елементів масиву за зростанням, використовуючи *метод послідовного пошуку екстремальних елементів*.
-
4. Виконати сортування елементів масиву за зменшенням, використовуючи *метод послідовного пошуку екстремальних елементів*.
-

ДОДАТКОВІ ВІДОМОСТІ :

Для розв'язання завдання масив зручно представити у вигляді лінійки елементів, в яких зберігаються числа. Наприклад :

Масив ABC

-4	16	12	0	-77	11	32	77	-41	2	12	-98	91	значення елем.
1	2	3	4	5	6	7	8	9	10	11	12	13	Індекси

При сортуванні, наприклад, за зростанням можна обрати два шляхи :

Перший метод :

а). Знайти в масиві ABC найменший елемент та його номер і встановити його на 1 місце, а 1 елемент - на місце, де був знайдений найменший елемент. В даному прикладі найменшим є 12 елемент, а його значення дорівнює : -98.

Після перестановки масив буде мати наступний вигляд :

Масив ABC

-98	16	12	0	-77	11	32	77	-41	2	12	-4	91	значення елем.
1	2	3	4	5	6	7	8	9	10	11	12	13	Індекси

Тепер дії п. а можна повторити, але пошук мінімального і перестановку треба почати тепер з другого елемента. В даному випадку мінімальним елемент буде п'ятий із значенням -77. Після перестановки масив буде мати вигляд :

Масив ABC

-98	-77	12	0	16	11	32	77	-41	2	12	-4	91	значення елем.
1	2	3	4	5	6	7	8	9	10	11	12	13	Індекси

Таким чином, дії п. а треба повторювати до тих пір, поки не буде досягнуто 12-го елемента. Після цього масив ABC буде відсортовано за зростанням.

Отже, якщо розмірність масиву дорівнює N, то треба створити два вкладених цикли, де зовнішній має N-1 разів повторити внутрішній. У внутрішньому циклі треба виконувати пошук найменшого елемента та його індекса. При цьому, після закінчення внутрішнього циклу треба виконувати дві дії :

- робити перестановку знайденого мін. елемента з елементом, що знаходиться на початку чергового пошуку.

- збільшувати на 1 спеціальну змінну BEG, яка має вказувати на початковий номер елемента, з якого в наступний раз буде відбуватись пошук мінімального елемента.

Наприклад :

```

BEG := 1; {починаємо шукати з 1 елемента : }
for i := BEG to 12 do begin
    Num_Min := BEG; {припустимо номер мін. елемента = BEG}
    Min := ABC[Num_Min]; {а це його значення}
    for k := BEG to 12 do
        if {.....} begin
            { тут відшукаємо номер найменшого елемента в масиві в
діапазоні індексів від BEG до 12 }
            end; {if}
        {переставимо значення елемента BEG та Num_Min}
    end;
end;

```



```

Z := ABC[BEG];
{запам'ятаємо значення початкового ел. в Z}
ABC[BEG] := ABC[Num_Min];
{перепишемо значення MIN в ел. BEG}
ABS[Num_Min] := Z;
{а значення ел. BEG - в ел. Num_Min}
BEG := BEG + 1;
{зсунемо початок наступного пошуку на 1}
end; {for}

```

Другий метод:

- Полягає в тому, що *попередньо* та *попарно* порівнюються елементи масиву, і, якщо наступний буде менше попереднього, то вони міняються місцями. Тобто найбільший з пари елемент "вспливає", тобто просувається в бік зростання значень масиву. Тому цей метод називається бульбашковим. Робота починається з першого та другого елементів. В даному випадку перший елемент менший за другий, отже, перестановку не виконуємо. Після цього порівнюємо другий і третій елементи. Треба робити перестановку. Тепер масив має вигляд:

Масив ABC

-4	12	16	0	-77	11	32	77	-41	2	12	-98	91	значення елем.
1	2	3	4	5	6	7	8	9	10	11	12	13	Індекси

Для третього і четвертого дії аналогічні:

Масив ABC

-4	12	0	16	-77	11	32	77	-41	2	12	-98	91	значення елем.
1	2	3	4	5	6	7	8	9	10	11	12	13	Індекси

Ці дії треба повторювати до порівняння 12 та 13 елементів, а потім цю всю процедуру (спочатку) треба повторювати ще 11 разів.

Тобто, якщо масив має розмірність N, то треба створити 2 вкладених цикли з кількістю проходів N-1 кожний. У внутрішньому циклі будуть порівнюватись та переставлятись за необхідністю сусідні елементи, а зовнішній - має його повторити N-1 разів.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ:

1. Яким чином можна поміняти значення пари змінних, наприклад, Y та U місцями?
2. В чому полягає метод послідовного пошуку екстремальних елементів при сортуванні масивів?
3. В чому полягає "бульбашковий метод" при сортуванні масивів?
4. Чому в "бульбашковому" методі треба повторювати (скільки разів?) процедуру перестановки елементів?
5. Порівняйте кількість проходів для першого і другого методів сортування масивів.

ЛАБОРАТОРНА РОБОТА N 8

ТЕМА: МОДЕЛЮВАННЯ ОБЛІКОВИХ ЗАДАЧ АГРОПРОМИСЛОВОГО НАПРЯМКУ В СЕРЕДОВИЩІ TURBO PASCAL

МЕТА РОБОТИ: Навчитись розробляти алгоритм та програму за допомогою яких формується певний обліковий документ та набутти навичок використання файлових структур даних

ЗАВДАННЯ:

1. Згідно варіанту скласти математичну модель задачі (ввести позначення змінних величин та записати формули). Номер варіанту визначається за номером прізвища студента в журналі викладача.
2. Розробити алгоритм задачі у вигляді блок-схеми.
3. Назву та шапку документа попередньо набрати у вікні редактора та записати в текстовий файл.
4. При розробці програми та вихідного документа мовою Turbo Pascal використати попередньо сформований текстовий файл.

ВАРІАНТИ ЗАВДАННЯ :

1. Обчислити процент вмісту окремих видів кормів в кормових одиницях від всіх витрат на одну корову

Вид корму	Зкормлено всього кормових одиниць, кг	В %% від всіх витрат кормів
Сіно	428,40	X
Сінаж	363,30	X
Силос	672,60	X
Концентрати	232,20	X
Пасов. Трави	646,20	X
Зелена підкормка	620,10	X
Пивна дробина	260,00	X
В сь о г о	У	100,0

2. Визначити витрати в людино-днях на виробництво 1ц продукції за такими даними

Вид продукції	Валовий збір, ц	Витрати, людино-днів	
		В сь о г о	На 1ц продукції
Зерно	45825	35697	X
Картопля	14062	13862	X
Молоко	26270	55720	X
Овочі	52536	54980	X
Молоко	4680	69229	X
В сь о г о	У	Z	W

3. Визначити, скільки азотних добрив необхідно внести в ґрунт, щоб отримати базову урожайність культур

К у л ь т у р а	Посівна площа, га	Норма внесення азотних добрив на 1га культури, кг	Всього азотних добрив, ц
Кукурудза на зерно	319	50	X
Кукурудза на з/к	287	30	X
Цукровий буряк	632	90	X
Соняшник	382	45	X
Озима пшениця	493	40	X
В с ь о г о	S	-	У

4. Обчислити, скільки потрібно кормів, щоб забезпечити необхідну кількість каротину

В и д к о р м у	Посівна площа, га	Норма внесення азотних добрив на 1га культури, кг	Всього азотних добрив, ц
Кукурудза	86	0,4	X
Сіно злакових	125	2,0	X
Овес	64	0,3	X
Силос злакових	96	2,0	X
Комбікорм	38	0,4	X
В с ь о г о	У	-	Z

5. Визначити розрахункову посівну площу, яка необхідна для вирощування 1ц товарної продукції рослинництва

Продукція	Урожайність, ц/га	Витрати на насіння, ц/га	Урожайність за відрахуванням витрат на насіння ц/га	Розрахункова площа на 1ц продукції, га
Пшениця	42,0	2,0	X	У
Кукурудза	50,3	0,3	X	У
Соняшник	24,0	0,2	X	У
Овочі	250,0	0,2	X	У
Виноград	80,0	-	X	У

6. Обчислити собівартість 1ц молока для кожного господарства

Ш и ф р госп-ва	Валовий надій молока, ц	Витрати на вироб ництво, грн.	Собівартість 1ц молока, грн
01	71755,6	1948668	X
02	71120,5	2253854	X
03	63189,8	1966804	X
04	75761,7	1863712	X
05	24467,3	713836	X
По р-ну	М	В	С

7. Розрахувати господарську придатність посівного матеріалу
овочевих культур для кожної культури за формулою
 $X = B * Ч / 100$

Н а з в а культури	Схожість, %(В)	Чистота, %(Ч)	Господарська придатність, %
Морква	80	97	X
Капуста	90	80	X
Огірки	82	96	X
Помідори	70	84	X
Цукрові буряки	90	98	X
Редиска	87	84	X

8. Визначити середню врожайність для кожної культури та
загальний валовий збір врожаю

Ш и ф р культури	Валовий збір, ц	Посівна площа, га	Середня врожайність, ц/га
201408	26200	521	В
201409	10427	467	В
202107	9863	412	В
203415	8677	283	В
206612	9039	316	В
207111	12096	296	В
В с ь о г о	У	С	-

9. Обчислити питому вагу валового збору врожаю для кожної
окремої культури

Ш и ф р культури	Валовий збір, ц	Питома вага збору врожаю, %%
201408	25700	X
202107	16830	X
203413	9871	X
206615	13586	X
206813	8074	X
В с ь о г о	У	100,0

10. Обчислити валовий збір для кожної культури окремо і в цілому для всіх культур

Ш и ф р культури	Посівна площа, га	Середня врожайність, ц/га	Валовий збір, ц
201405	479	49,6	В
201408	143	25,2	В
201502	713	21,4	В
203507	247	18,3	В
205101	169	19,7	В
Всього	С	-	У

11. Розрахувати витрати на посівний матеріал для кожної культури окремо і для всіх культур разом

Ш и ф р культури	Необхідна к-сть посівних матеріалів, ц	Ціна, грн./ц	Витрати для посівного матеріалу, грн.
181001	2800	70,6	Х
185001	12079	61,4	Х
187001	150	40,2	Х
188001	20	190,7	Х
189001	569	35,3	Х
Всього	-	-	У

12. Визначити, скільки протеїну міститься в кормах ВРХ

В и д к о р м у	К-сть корму, ц	К-сть протеїну в 1ц корму, кг	Всього протеїну, ц
Озимий ячмінь	86	8,0	Х
Комбікорм	125	10,0	Х
Силос кукурудзний	64	1,4	Х
Сіно злакових		4,0	Х
Сіно бобових	96	11,0	Х
Овес	38	8,5	Х
Всього	У	-	W

13. Визначити кількість кормових одиниць, які містяться в кормах для ВРХ

В и д к о р м у	К-сть корму, ц	К-сть кормових одиниць в 1ц корму, ц	Всього кормових одиниць, ц
Зерно	864	0,4	Х
Овочі	1417	0,1	Х
Силос	935	0,17	Х
Сіно багаторічних трав	489	0,45	Х
Комбікорм	391	0,9	Х
Всього	Z	-	У

14. Виконати розрахунок необхідної посівної площі для кожної культури окремо і в цілому для господарства, а також загальний валовий збір врожаю

Ш и ф р культури	Валовий збір, ц	Середня врожайність, ц/га	Посівна площа, га
201408	25100	50,2	S
202107	9719	24,8	S
203415	10406	38,5	S
206612	8921	32,4	S
207111	12038	42,8	S
Всього	Y	-	П

15. Виконати розрахунок виходу продукції на 100 га с/г угідь для кожного господарства окремо і для всіх господарств в цілому за формулою: $X = 100 * П / С$

Назва господарства	Одержано валової продукції (П), тис.грн.	К-сть с/г угідь (С), га	Вихід валової продукції на 100 га с/г, тис.грн.
КСП "Буг"	1597,4	3100	X
КСП "Ясний"	980,3	2140	X
КСП "Град"	1843,8	2980	X
КСП "Екран"	2624,3	3298	X
КСП "Дніпро"	1981,7	2859	X
Всього	-	-	Y

16. Визначити необхідну кількість насіння для кожної культури окремо і в цілому для всіх культур разом

Ш и ф р культури	Посівна площа, га	Норма посіву на 1 га, ц	Необхідна к- сть насіння, ц
181001	220	2,6	X
183001	156	2,4	X
183002	540	2,5	X
183003	170	2,2	X
185007	460	2,9	X
Всього	Z	-	Y

17. Обчислити вміст фосфору в кормах ВРХ

В и д корму	К-сть корму, ц	Вміст фосфору в 1ц корму, ц	Всього фосфору, ц
Кукурудза на зерно	833	0,3	X
Кормові коренеплоди	612	0,03	X
Силос кукурудзяний	486	0,06	X
Силос злакових	294	0,05	X
Озимий ячмінь	977	3,5	X
Кормовий преципитат	136	17,0	X
Всього	Z	-	Y

18. Визначити відсоток виконання плану товарообігу господарством /щомісячно і за рік/

Місяць	План. товарообороту, грн	Фактичне виконання	
		млн. грн	%%
Січень	22300,0	23210,4	X
Лютий	21600,0	22526,0	X
Березень	23400,0	23775,5	X
Квітень	24000,0	24764,1	X
Травень	23700,0	23632,9	X
Червень	23800,0	22346,6	X
Липень	23500,0	23339,8	X
Серпень	23000,0	23866,2	X
Вересень	24560,0	23359,7	X
Жовтень	23800,0	24022,3	X
Листопад	24600,0	24229,2	X
Грудень	23500,0	25247,8	X
Всього	Т	Ф	W

19. Визначити, яка частка (в %%) від урожаю культур, що вирощені, залишилася в господарстві

Назва культури	Зібрано, т	Держзамовлення, т	Продано, т	Залишок, %%
Яблука	4630	1960	1390	X
Пшениця озима	4820	2000	1280	X
Кукурудза	2960	1800	470	X
Соняшник	1640	530	790	X
Овес	1370	-	720	
Всього	Z1	Z2	Z3	Z4

20. Визначити прибуток (збитки) господарства від реалізації овочів кожної культури окремо і в цілому всіх культур

Шифр культури	К-сть реалізованих овочів, ц	Факт. ціна реалізації, грн./ц	Планова собівартість, грн./ц	Прибуток, грн.
201506	370,5	51,8	50,4	П
202101	834,4	20,2	19,5	П
202109	480,7	28,4	29,9	П
202212	948,7	19,3	16,6	П
202433	1080,5	45,1	47,2	П
202517	627,1	37,4	37,6	П
Всього	К	-	-	S

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ:

- . До якого типу алгоритмів можна віднести алгоритм по даній роботі ?
- . Що таке файл ? Як сформувати текстовий файл незалежно від програми?
- . Як описується текстовий файл в програмі?
- . Яким оператором можна відкрити файл для читання?
- . Як працює оператор читання текстового файлу?
- . Як виводити дані в задану позицію рядка?
- . Яким чином можна сформувати шапку документа без використання текстового файлу?

ЛАБОРАТОРНА РОБОТА N 9

ТЕМА : ВИВЧЕННЯ ГРАФІЧНИХ МОЖЛИВОСТЕЙ TURBO PASCAL (частина I).

МЕТА РОБОТИ: Навчитись розробляти алгоритми та програми, що використовують графічні методи виведення інформації.

ЗАВДАННЯ 1. Провести дослідження за допомогою наладчика згідно із завданням графічних процедур TURBO PASCAL.

ЗАВДАННЯ 2. Розробити програму для відтворення зображення, наведеного в завданні. Результати показати викладачеві.

ВМІСТ ЗВІТУ:

1. Навести розроблені програми згідно з завданнями 1-2 до лабораторної роботи.
2. Навести висновки за результатами роботи.

ЗАВДАННЯ 1.

Дослідити роботу операторів, що наведені в програмі N1 за допомогою наладчика оболонки TURBO PASCAL. Для цього або зробіть копію її в файл-програму MYGRAPH.PAS за допомогою засобів TURBO PASCAL або наберіть її з клавіатури.

Програма N1

=====

```
Uses Graph;
var Grd, GrM : integer;
i, Radius : byte;
BEGIN
  GrD := Detect;
  InitGraph( GrD, GrM, 'z:\bp\bgi\'); {відкриємо графічний
режим}

  {Наступний оператор намалює точку червоного кольору (Red) по
центру екрану. Функція GetMaxX дає величину розміру екрану по
горизонталі, а GetMaxY - по вертикалі. За допомогою цілого
ділення div на 2 - одержимо координати центру екрану}
  PutPixel(GetMaxX div 2, GetMaxY div 2, Red);

  Rectangle(0, 0, GetMaxX, GetMaxY);
  { намалюємо рамку максимального розміру }

  {а тепер проведемо невеличку лінію ...}
  Line(100, 50, 250, 150);

  {згадайте, які координати вказуються в дужках. А тепер
скористаємось іншим оператором, що малює лінію}

  LineTo(50, 50);
```


{зверніть увагу на те, що оператор Line не пересунув невидимий курсор. Він так і залишився в початкових координатах 0,0. Але після оператору LineTo(50, 50) він вже знаходиться в координатах 50,50. Перевіримо це і використаємо ще один оператор LineTo}

LineTo(150, 250); { а тепер зробить відповідні висновки ... }

Bar(300,150,550,300); { а цей оператор малює зафарбований прямокутник }

{а ці два наступні - малюють прямокутний стовпчик в проекції
Bar3D(X1, Y1, X2, Y2, Z, TopOn (TopOff))

Bar3D(10,300,30,150, 10, TopOn);

Bar3D(50,300,70,150, 10, TopOn);

{ тепер намалюємо такі ж самі, але зсунуті вправо і без верхівки}

Bar3D(110,300,130,150, 10, TopOff);

Bar3D(150,300,170,150, 10, TopOff);

{оператор Circle(X,Y,R); малює коло з координатами X,Y та радіусом R}

Circle(300, 100, 20);

{оператор Ellipse(X,Y, Fil, Fi2, XRadius, YRadius) малює еліпс, що розташований в координатах X,Y, що починається з кута Fil і закінчується Fi2, а також має радіуси XRadius, YRadius }

Ellipse(400, 100, 0, 360, 30, 50);

Ellipse(400, 100, 0, 180, 50, 30);

{оператор Arc(X,Y, Fil, Fi2, Radius) малює дугу навколо координат X,Y з кута Fil по кут Fi2, що має радіус - Radius. Намалюємо декілька дуг навколо одного центру...}

for Radius := 1 to 6 do Arc(100, 400, 0, 90, Radius * 10);

{оператор PieSlice(X, Y, Fil, Fi2, Radius) має параметри, аналогічні оператору Arc, але малює сектор ... }

PieSlice(200, 400, 0, 140, 40);

{оператор Sector(X, Y, Fil, Fh2, XRadius, YRadius) малює зафарбований еліпсоїдальний сегмент...}

Sector(400, 450, 0, 170, 30, 100);

Readln;

CloseGraph; {закриємо графічний і повернемося в текстовий режим}

END.

ЗАВДАННЯ 2.

Розробити програму, за допомогою якої буде відображено контури та загальні форми наступних об'єктів (при розробці програми врахувати те, що розміри графічного екрану для комп'ютерів 486SX-25 або K6 будуть ключатись 480x640 точок, а для комп'ютерів типу "Пошук" - 200x640 точок.) :

ВАРІАНТИ ЗАВДАННЯ 2 (для фальтету МСТ):

1. Вантажний бортовий автомобіль типу ГАЗ-66.
2. Мікроавтобус типу DAF.
3. Легковий автомобіль типу "Волинь".
4. Легковий автомобіль типу ВАЗ-2109.
5. Легковий автомобіль з відкритим верхом (кабріолет).
6. Гусеничний трактор.
7. Колесний трактор.

ВАРІАНТИ ЗАВДАННЯ 2(для економічного факультету):

- робоче місце бухгалтера-економіста, що включає :
стіл, телефон, факс, комп'ютер з монітором, клавіатурою, мишкою, стопку паперів, чашку кави.

** В програмах мають бути максимально використані графічні оператори. Кількість графічних операторів в програмі має бути на менше 20. Перед розробкою тексту програми треба розробити проект зображення на папері з координатами, що відповідають кількості графічних точок монітора.*

ПИТАННЯ САМОКОНТРОЛЮ:

1. Як працюють графічні оператори **InitGraph, Detect, CloseGraph, Line, LineTo, Rectangle, MoveTo, Bar, Bar3D, Circle, Ellipse, Arc, PieSlice, Sector**
2. Які координати екрана монітора можуть існувати в графічних режимах комп'ютера?
3. Які кольори можна використовувати в графічному режимі ?

ЛАБОРАТОРНА РОБОТА N 10

ТЕМА : ВИВЧЕННЯ ГРАФІЧНИХ МОЖЛИВОСТЕЙ TURBO PASCAL (частина II).

МЕТА РОБОТИ: Навчитись використовувати методи нормування масивів для побудови графіків функцій.

ЗАВДАННЯ 1. Побудувати вісі координат XOY та графік функції $Y=F(X)$, використовуючи методи табулювання і нормування її під відповідний тип графічного екрану згідно з варіантом.

ЗАВДАННЯ 2. Розробити блок-схему та програму для відтворення графічного зображення функції, наведеного в завданні 1. Порівняти графічне зображення на екрані з зображенням в звіті по лабораторній роботі 4. Результати показати викладачеві.

ВМІСТ ЗВІТУ:

1. Навести розроблену блок-схему та програму за завданням.
2. Навести висновки за результатами роботи.

Варіант завдання функції вибрати з таблиці 1, лабораторна робота 4.

ДОДАТКОВІ ВІДОМОСТІ :

При виконанні завдання треба користуватись наступною схемою роботи :

1. Визначити крок зміни аргументу функції виходячи із початкових та кінцевих його значень, а також із режиму обраного графічного екрану (у Вашому випадку — це 640 точок по горизонталі).
2. Виконати табуляцію заданої функції в масив з кількістю елементів, що відповідає розміру екрану по горизонталі.
3. Знайти мінімальне та максимальне значення занесеної в масив функції.
4. Порахувати масштабний коефіцієнт по вертикалі, який має дорівнювати $480/(MAX-MIN)$, де 480 - це кількість точок по вертикалі (для комп'ютерів типу "ПОШУК" це число - 200).
5. Побудувати графік табульованої функції, використовуючи процедуру виведення точки на екран PutPixel(...), а також використовуючи її мінімальне значення та масштабний коефіцієнт. Зверніть увагу на те, що **МІНІМАЛЬНЕ** значення функції має **ОБОВ'ЯЗКОВО ТОРКАТИСЬ** вісі абсцис (як це зробити ?). Окрім того зверніть увагу на те, що зростання координат Y іде вниз. Як зробити, щоб графік не був перевернутий ?.
6. Побудувати вісі координат (Процедури Line(...) або LineTo(...)).

7. Зробити підписи до графіку (процедури OutText(...) або OutTextXY(...)), графік будувати за допомогою операторів PutPixel, а потім замінити його оператором LineTo(...).

ПИТАННЯ САМОКОНТРОЛЮ:

1. Які проблеми виникають при будіванні графіків на екрані комп'ютера?
2. Для чого потрібно знати мінімальне значення функції?
3. Що таке масштабний коефіцієнт і для чого він потрібен?
4. Чому, якщо не вжити спеціальних мір, графіки будуються у перевернутому вигляді?
5. Які оператори дозволяють виводити текстову інформацію в графічному режимі?
6. Які координати можна використовувати для операторів п.5 ?
7. Перерахуйте оператори, що працюють в графічному режимі.

ЛАБОРАТОРНА РОБОТА N 11

ТЕМА : ВИВЧЕННЯ ФУНКЦІЙ РОБОТИ З РЯДКОВИМИ ЗМІННИМИ (Електронний довідник з рядковими масивами)

МЕТА РОБОТИ: Навчитись використовувати методи роботи з рядковими змінними.

ЗАВДАННЯ 1. Набрати програму N1 з прикладу (або скопіювати її в файл під назвою STR). Зверніть увагу, що в рядку Uses мають бути вказані дві бібліотеки - CRT та PalCRT. Дослідити роботу рядкових функцій з цього прикладу.

ЗАВДАННЯ 2. Набрати програму N2 з прикладу (або скопіювати її в файл під назвою STR1). Дослідити роботу рядкових функцій з цього прикладу.

ЗАВДАННЯ 3. Розробити програму та блок-схему електронного довідника згідно з одержаним варіантом. Результати показати викладачеві.

ВМІСТ ЗВІТУ:

1. Навести розроблену блок-схему та програму за завданням.
2. Навести висновки за результатами роботи.
3. Навести приклади, які пояснюють роботу функцій Pos, Delete, Insert, Copy.

ВАРІАНТИ ЗАВДАНЬ :

1. Розробити програму електронного довідника деканату. Робочі поля - *прізвище, ім'я, по-батькові, місце проживання, дата народження* (у вигляді DD.MM.PP), .

2. Розробити програму електронного довідника ДАІ. Робочі поля - *номерний знак, марка автомобіля, власник, місце проживання.*

3. Розробити програму електронного англо- (або німецько-, або французько- ...) - українського словника. Робочі поля - *слово на іноземній мові, його переклад, коментар або пояснення.*

4. Розробити програму електронного телефонного довідника. Робочі поля - *номер телефону, ПІБ власника телефону., назва вулиці, номер дому, номер квартири.*

5. Розробити програму електронного довідника з автобусних рейсів. Робочі поля — *напрямок (напр. "Літин"), номер рейсу, час відправлення, час прибуття в пункт призначення.*

6. Розробити програму електронного довідника для залізничного вокзалу. Робочі поля — *напрямок (напр. "Мариупіль"), номер рейсу, час відправлення, час прибуття в пункт призначення.*

7. Розробити програму електронного бібліотечного каталогу. Робочі поля — назва книги, автор, рік видання, класифікатор.

8. Розробити програму електронного довідника з товарів в магазині. Робочі поля — назва товару, ціна, залишок, постачальник.

*** Для парних варіантів цикл меню має бути створено на основі операторів *While ... do ...*, а для непарних - на основі операторів *Repeat ... Until ...*

ХІД ВИКОНАННЯ РОБОТИ :

При виконанні завдання треба користуватись наступною схемою:

1. Створити блок даних за допомогою оператора напередвизначених змінних CONST. Кожне поле бази даних має складатись з окремого масиву рядкових змінних, наприклад, для довідника аптеки :

```
CONST
{Назва препарату}
NM : array [1..5] of string = ('Бісептол', 'Парацетамол',
'Гепарин', 'Новокаїн', 'Дексаматазон');

{ціна}
Price : array [1..5] of real = (1.45, 0.80, 2.32, 1.45, 3.47);

{упаковка}
Encasement : array [1..5] of string =
('Табл.', 'Табл.', 'Ампул.', 'Ампул.', 'Драже');
```

2. Створити просте меню користувача, що пропонувало б варіанти пошуку даних в базі даних довідника або вихід з програми.

3. В залежності від обраного режиму пошуку (в даному випадку - це пошук за назвою, або за ціною або за упаковкою), запустити певний модуль програми, який, перебираючи в циклі рядки певного масиву, знайшов би той, який співпадає з введеним для пошуку значенням (наприклад, знайти препарат "Гепарин"). Знайдений таким чином рядок з масиву NM буде мати свій індекс (в даному випадку — 3). За цим індексом вивести на екран відповідні значення з всіх масивів, наприклад :

Гепарин ціна : 2.32 грн., упаковка : Ампул.

Пошук продовжити до кінця масиву : гепаринів може бути декілька !

4. В тому випадку, коли препарат не буде знайдено, треба вивести на екран повідомлення про невдалий пошук, наприклад :

Препарат не знайдено ...

5. Модулі, що будуть здійснювати пошук по іншим полям (в даному випадку - це за ціною, або за упаковкою), мають працювати аналогічно п.3.

6. Після виведення повідомлень програма має повертатись знову в меню.

7. Після створення програми ввести в неї функції StrLower_ або StrUpper_, які дозволять в програмі ігнорувати регістри маленьких\великих букв. Для цього дослідити приклад, наведений в програмі з додаткових відомостей.

8. Використовуючи функцію Pos, додати до створеної програми можливість пошуку за фрагментом слова, наприклад, знайти препарат, в якому зустрічається фрагмент "вокаїн".

ДОДАТКОВІ ВІДОМОСТІ:

```
{приклад створення програми на основі циклу Repeat...Until...}
Repeat
  ClrScr;
  writeln(' - Шукати за назвою препарата (1)');
  writeln(' - Шукати за ціною препарата (2)');
  writeln(' - Шукати за назвою упаковки (3)');
  writeln(' - Закінчити роботу (0)');
  writeln(' Введіть режим роботи довідника : ');
  readln(mode);

  if mode = 1 then begin
    ..... {шукаємо за назвою}
    ..... {виводимо результати пошуку на екран}
    readln; {призупинемо програму до натискання
    ENTER, щоб подивитись на результати}
  end;

  if mode = 2 then begin
    ..... {шукаємо за ціною}
    ..... {виводимо результати пошуку на екран}
    readln; {призупинемо програму до натискання
    ENTER, щоб подивитись на результати}
  end;
  ..... {шукаємо за іншими ознаками}
until mode = 0; {виводимо з циклу, коли введений режим = 0}
```


(*
*)

Програма N 1

```
{пояснення роботи функцій StrLower_, StrUpper_, Pos}
Uses Crt, PalCrt;
var st, st1, st2 : string;

BEGIN
  ClrScr;
  st := 'АБРАКАДАБРА';
  st1 := 'АБРАКАДАБРА';
  st2 := 'абракадабра';

  writeln('В рядковій змінній st записано : ', st);
  writeln('В рядковій змінній st1 записано : ', st1);
  writeln('В рядковій змінній st2 записано : ', st2);
  writeln;
  writeln('Якщо рядок st збігається з рядком st1');
  writeln('або рядок st збігається з рядком st2');
  writeln('або рядок st1 збігається з рядком st2, то включимо звук');

  if (st = st1) or (st=st2) or (st1=st2) then write('^G');
  writeln(' => Як бачимо, збігу не відбулося');

  writeln;
  writeln('А от функція StrUpper_ дає рядок тільки з великими літерами : ', StrUpper_(st));
  writeln('а функція StrLower_ дає рядок тільки з маленькими літерами : ', StrLower_(st));

  writeln;
  writeln('Отже, якщо порівнювати рядки "скрізь" функції StrLower_ або StrUpper_');
  writeln('то тоді можна ігнорувати регістри маленьких і великих букв ...');
  writeln;
  writeln('А тепер ...');

  if (StrLower_(st) = StrLower_(st1)) then writeln('^G, 'st та st1 збігаються.');
```

```
  if (StrLower_(st) = StrLower_(st2)) then writeln('^G, 'st та st2 збігаються.');
```

```
  if (StrLower_(st1) = StrLower_(st2)) then writeln('^G, 'st1 та st2 збігаються.');
```

```
  writeln;
  writeln('Функція POS(st, st1) цікава тим, що вона дорівнює числу, яке є номер символу,');
```

```
  writeln('з якого відбулась збіжність (або входження) рядку st до рядку st1');
```

```
  writeln('Якщо збіжності немає, то вона дорівнює 0.');
```

```
  writeln('Збіжність "бра" та "абракада" відбулась з ', Pos('бра', 'абракада'), ' символу.');
```

```
  writeln('а збіжність "чін" та "Ячмінь" відбулась з ', Pos('чін', 'Ячмінь'), ' символу,');
```

```
  writeln('тобто збіжності немає');
```

```
(* Отже, якщо є збіжність, то POS більше за нуль ...
```



```

if Pos(st1, st2) > 0 then begin
    ..... { є збіжність }
end;

{ а якщо ігнорувати регістр букв, то }
if Pos(StrLower_(st1), StrLower_(st2)) > 0 then begin
    ..... { є збіжність }
end;

{ або }
if Pos(StrUpper_(st1), StrUpper_(st2)) > 0 then begin
    ..... { є збіжність }
end;
*)

readln;

ClrScr;
writeln;
writeln('Процедура Delete(st,i,k) дозволяє вилучити з рядку
st з i-ї позиції k-символів');
st := 'ABCDEFGH IJ';
writeln('Зараз в змінній st записано : ', st);
Delete(st, 4, 2);
writeln('А після виконання Delete(st, 4, 2); буде : ', st,
Зрозуміло ?');

writeln;
writeln('Функція Copy(st,i,k) дозволяє скопіювати (або
одержати "вирізку" ) з рядку st з i-ї позиції в k-символів');
st := 'Бухгалтерія';
writeln('Зараз в змінній st записано : ', st);
st1 := Copy(st, 4, 2);
writeln('А після виконання Copy(st, 4, 2); в змінній st1
буде : ', st1, ' Зрозуміло ?');

writeln;
writeln('Процедура Insert(st1,st,k) дозволяє вставити рядок
st1 в рядок st з i-ї позиції');
st := 'Трактор';
writeln('Зараз в змінній st записано : ', st);
Insert('то', st, 4);
writeln('А після виконання Insert(`то`, st, 4); в змінній
st буде : ', st);

readln;

END.

```



```
(*  
*)
```

Програма N 2

```
Uses CRT;  
var r,r1,r2 : string;  
i : integer;  
BEGIN  
  ClrScr;  
  writeln('Програма, що демонструє роботу рядкових функцій  
Pos, Insert, Delete, Length ');  
  writeln('Склав доц. В.М.Носолюк');  
  for i := 1 to 40 do write('='); writeln; writeln;  
  writeln('Найпростіший редактор рядку. ');  
  write('Введіть початковий текст : '); readln( r );  
  write('Замінити в ньому фрагмент : '); readln( r1 );  
  {якщо рядок r1 не входить до рядку r, то закінчимо програму}  
  if pos(r1, r) = 0 then begin  
    writeln('^G, а такого фрагменту в тексті немає ...');  
    delay(2000); { затримка }  
    Halt; { примусове закінчення програми }  
  end;  
  
  write('на : '); readln( r2 );  
  
  {шукаємо в рядку r рядки r1}  
  while pos(r1, r) <> 0 do begin  
    {визначаємо позицію (i), з якої r1 входить до r}  
    i := pos(r1, r);  
  
    {вирижемо з рядку r кусок довжиною в r1  
    ( довжину визначає функція length(r1) ) }  
    delete(r,i,length(r1));  
  
    {вставимо рядок r2 в рядок r з позиції i }  
    insert(r2,r,i);  
  end; {while}  
  
  writeln('Новий текст : ', r);  
  readln;  
  
END.
```

ПИТАННЯ САМОКОНТРОЛЮ :

1. Що таке рядкові змінні та масиви рядкових змінних ?
2. Чи можна додавати рядкові змінні ?
3. Чи можна порівнювати між собою рядкові змінні, рядок та рядкову змінну, рядок і рядок ?
4. Чи буде збіжність, якщо порівнювати два рядки з одним і тим же записом, що написані різними за розміром символами ?
5. Які функції дозволяють переводити регістр символів ? Наведіть приклади.
6. Як працюють оператори Pos, Delete, Insert, Copy?

ЛАБОРАТОРНА РОБОТА N 12

ТЕМА : ВИВЧЕННЯ ФУНКЦІЙ РОБОТИ З РЯДКОВИМИ ЗМІННИМИ (Електронний довідник з файловою базою даних)

МЕТА РОБОТИ: Навчитись працювати з текстовими файлами.

ЗАВДАННЯ :

Створити електронний довідник згідно з варіантом до лабораторної роботи N10. Використати з неї розроблене меню команд, а також методи порівняння рядків. Базу даних розташувати в тестовому файлі BASE.PAS.

ВМІСТ ЗВІТУ:

1. Навести розроблену блок-схему та програму за завданням.
2. Навести висновки за результатами роботи.

РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ РОБОТИ :

При виконанні завдання треба користуватись наступною схемою:

1. Відкрити новий файл під назвою BASE.PAS і в нього записати свою базу даних. При цьому кожний запис має починатись з нового рядка. Наприклад, для випадку аптечного довідника інформацію про препарат 'Гепарин' за ціною 2.32 грн., що має упаковку типу 'Ампули' в файл BASE.PAS треба записати :

Гепарин
2.32
Ампули

Для двох препаратів запис буде мати такий вигляд :

Гепарин
2.32
Ампули
Дексаметазон
1.78
Таблетки

Звичайно, що в такому файлі має спостерігатись певна циклічність, що буде визначатись кількістю параметрів про об'єкт. В даному випадку вона складе 3.

2. В програмі, що має шукати певні записи в базі даних є сенс використати меню з лаб. роботи N10, а пошук створити в циклі такого вигляду :

Assign(f, 'BASE.PAS'); {цей оператор назначає змінній f файл
BASE.PAS}

Reset(f); {цей оператор відкриває файл для читання і встановлює
вказівник
читання на перший рядок файлу }


```

repeat
  Readln(f, st1); {тут з файлу f читається рядок в змінну st1 і
вказівник переводиться на наступний рядок}
  Readln(f, st2);      {аналогічні дії, але читається наступний
рядок в змінну st2 }
  Readln(f, st3);      { ... аналогічно}

  {після такої порції читання в рядках st1, st2, st3 ми будем
мати слова 'Гепарин', '2.32', 'Ампули', які за аналогічними до
лаб. роботи N11 правилами можна порівнювати з рядком пошуку, що
був введений з клавіатури}

  {.....}

until Eof; {повторювати читання, поки не закінчиться файл}

Close(f); {закрити файл}

```

Зверніть увагу на те, що в такій програмі повністю відсутні масиви, а їх роль виконують файл BASE.PAS і три допоміжні змінні st1, st2, st3. Таким чином програма-довідник стає незалежною від самого довідника бази даних. Вона лише шукає в ньому потрібні дані !

3. Не забудьте, що в блоці описання змінних текстові файли треба описати так як, наприклад, наведено нижче :

```

Var   f, w   : text; {f і w - це текстові файли}

```

ПИТАННЯ САМОКОНТРОЛЮ :

1. Що таке файл ? Як описується текстовий файл в блоці VAR ?
2. Яким чином об'являється файл в програмі ?
3. Яким оператором можна відкрити файл для читання ?
4. Яким оператором можна відкрити файл для запису ?
5. Чи можна з файлу, що вікритий, наприклад для запису, читати інформацію ?
6. Як працює оператор читання текстового файлу ?
7. Як працює оператор запису в текстовий файл ?
8. Які оператори ставлять вказівник читання\запису на початок файлу (на перший його рядок)?
9. Який оператор дозволяє визначити кінець файлу ? Коли він спрацьовує?

ЛАБОРАТОРНА РОБОТА N 13

ТЕМА : ВИВЧЕННЯ ОПЕРАТОРІВ РОБОТИ З ТЕКСТОВИМ ЕКРАНОМ. СТВОРЕННЯ МЕНЮ.

МЕТА РОБОТИ : Вивчити методи створення меню користувача.

ЗАВДАННЯ :

1. Розібрати наведені приклади за допомогою наладчика TURBO-DEBUGGER та з'ясувати, як вони працюють.
2. Розробити меню до програми електронного довідника.
3. Зробити спеціальні функції для введення текстової інформації в програму.
4. Створити спеціальні процедури для виведення результатів роботи на екран.
5. Додати до електронного довідника звукові процедури.

ВМІСТ ЗВІТУ:

1. Навести розроблену блок-схему та програму за завданням.
2. Навести висновки за результатами роботи.

ТЕОРЕТИЧНІ ВІДОМОСТІ :

В тому випадку коли кількість операторів в програмі сягає певної кількості, то все частіше виникають випадки, коли може повторюватись один і той самий фрагмент програми. Очевидно, що його оператори повторювати нерационально. Окрім цього великі програми, якщо вони "написані" методом лінійного програмування читати досить складно. Що ж робити? Елементарно ! Програмісти придумали таку чудову штуку, яка на Pascal'і називається **процедура**, а в інших мовах програмування - **підпрограма**. Процедура - це спеціальний блок в програмі, який має бути розташованим **перед** викликаючою програмою і оформляється за допомогою спеціального оператора **procedure**, наприклад:

```
Procedure ABC;  
begin  
  {тут ідуть оператори самої процедури}  
end;
```

Як видно цей блок починається словом Procedure, за яким іде її ім'я. В даному випадку це ABC. Між операторами **begin-end** ідуть оператори самої процедури. Ними можуть бути і оператори присвоєння, і цикли, і оператори переходів, і навіть імена інших процедур! Головне (і це буде найбільш грамотно) робити процедуру так, щоб вона була максимально незалежна від програми, з якої її будуть викликати (або запускати).

Як викликати процедуру? За допомогою її імені, яке треба як оператор написати в викликаючій програмі. Цікаво те, що Ви вже це робили в попередніх роботах, але про користування процедурами не знали. Пам'ятаєте оператори

Write, ClrScr, CloseGraph? Це і є звичайні процедури, але вони знаходились в бібліотечних модулях **Crt** та **Graph** і тому їх тексту в Ваших програмах не було.

А тепер подивіться на фрагмент програми, в якій є процедура, що виведе на екран повідомлення про призначення нашої програми :

```
Uses CRT;
var .....
const ....
{і так далі ...}

Procedure CBA;
begin
  ClrScr;
  Writeln('Ця програма призначена для вивчення процедур');
  Writeln('Виконав студент Панасюк Н.А.');
```

end;

```
BEGIN
  CBA;
  {а тут пішла програма ...}
END.
```

До речі, Ви помітили, з процедури **CBA** викликаються інші процедури **ClrScr** та **Writeln** ? Получилось щось типу програмної "матрьошки". От і виходить, що наша програма складається всього з трьох операторів **BEGIN-CBA;-END**. Правда вона читається дуже легко? А тепер зверніть увагу на те, що процедура розташована **перед** програмою. Якщо в Вашій програмі треба мати декілька процедур, то розташуйте їх одна під одною в будь-якому порядку за умови, що одна процедура не викликається з іншої. В протилежному випадку процедура, яку викликає інша процедура, має бути розташована перед нею.

Мабуть Ви помітили, що в програмі процедура **Writeln** використовується з дужками. В них пишеться те, що має потрапити в саму процедуру для її майбутньої роботи. Можна вважати, що це є пошта для неї. А якщо казати точно, то в процедуру передаються параметри. Кількість параметрів, які передаються та їх типи можуть бути будь-якими. Важливо тільки зберігати їх порядок слідування. Ви теж можете написати свої процедури, що користуються параметрами. А правила такі, що в дужках треба їх описати, щось нахалт оператору **var** :

```
Procedure MyProc(i, k : integer; a : string; d, f : real);
begin
  i := i - k;
  d := Sin(f) + i;
  ClrScr;
  Writeln(a, d : 4 : 6);
end;
```

```
BEGIN
  {.....}
  MyProc(-78, j, 'Результат складає : ', 4.67, z);
END.
```

Подивіться уважно на наведений фрагмент і спробуйте його виконати. Получається наступне : головна програма запускає процедуру **MyProc** і передає

її -78, значення своєї змінної *j*, стрічку 'Результат складає : ', 4.67 та значення своєї змінної *z*. Увага! Те, що передалось в процедуру потрапило в її змінні *i*, *k*, *a*, *d*, *f* відповідно! Тобто *i* набуло значення -78, *k* - значення змінної *j* головної програми (яке саме - тут визначити неможливо), *a* - значення 'Результат складає : ', *d* - значення 4.67, *f* - значення змінної *z* головної програми (яке саме - теж визначити неможливо).

А тепер згадайте цікаві оператори, як наприклад **Sin(...)** або **Cos(...)**. Це що процедури? Схоже, але ні. Після спрацювання їх назва дорівнює числу. Згадаємо, як ми користувались ними. Ось декілька прикладів :

```
x := Sin(0.678);
writeln(Sin(-2.543));
z := Cox(s)*Sin(567.88);
```

Так от, процедури, в назві яких залишається результат їх власних дій, називаються функціями. Як Ви знаєте, в Turbo Pascal'і немає математичної функції тангенсу. А ми її створимо :

```
Function Tan( x : real) : real;
var y : real;
begin
  y := Sin(x)/Cos(x);
  Tan := y; {перед виходом передамо функції значення y}
end;
```

Зверніть увагу, що функція описується словом **Function** і закінчується обов'язково вказівником її типу, який в даному випадку є дійсним числом. Тип функції, як і змінної може бути будь-яким. В наступних прикладах ви зустрінете типи **Integer** та **String**. Як і в процедуру в функцію можна або взагалі не передавати ніякі параметри, або будь-яку їх кількість з будь-яких комбінацій типів.

І на останнє - деякі хитрощі користування змінними.

1. Якщо процедура або функція знаходиться в середині програми (як в наших прикладах), а не в бібліотечних модулях, то вона може напряму (функції), то вони є спільними як для користуватись змінними головної програми. Через те, що їх об'явили до процедури головної програми та і для процедур (або функцій). Такі змінні називаються глобальними.

2. Якщо в процедурі (або в функції) за допомогою оператора **var** об'явлені свої змінні, то ними може користуватись лише ця процедура (або функція). Такі змінні називаються локальними. Не тіште себе надією, що якщо Ви об'явили в головній програмі, наприклад, змінну **YY** і в процедурі теж змінну **YY** і це буде *одне і теж*, то Ви помиляєтесь. Це будуть **абсолютно різні** змінні, хоча і будуть мати однакове ім'я.

3. Бувають випадки, коли процедура має існувати незалежно від програм, які будуть її викликати, але при цьому вона повинна мати з програмою, яка її викликала спільні змінні. Згадайте, як працює процедура **ReadLn(...)**. Вона читає те, що було набрано на клавіатурі і результат після натискання клавіші **Enter** розміщує в змінній, що ми поставимо в дужках. Як це робиться? Адже ми

знаємо, що через дужки значення потрапляють тільки в процедуру, як пошта в поштову скриньку. А для того, щоб змінна в дужках стала спільною і для процедури і для програми, що нею користується, треба використати слово **var** в дужках передачі змінних. Зробимо спільними наприклад змінні **d** і **f**.

```
Procedure MyProc( i, k : integer; a : string; var d, f : real);
```

```
begin
```

```
  i := i - k;
```

```
  d := Sin(f) + i;
```

```
  ClrScr;
```

```
  Writeln(a, d : 4 : 6);
```

```
end;
```

```
BEGIN
```

```
{.....}
```

```
  MyProc(-78, j, 'Результат складає : ', C, Z);
```

```
END.
```

Після того, як спрацює рядок **MyProc(-78, j, 'Результат складає : ', C, Z);** в головній програмі, то спочатку значення **C** та **Z** потраплять в процедуру **MyProc** у вигляді змінних **d** та **f**, а після закінчення процедури **MyProc** - змінні **d** та **f** передадуть своє нове значення назад змінним **C** та **Z**. Цікаво, що в цьому випадку ми маємо абсолютно протилежну ситуацію - імена змінних різні, а насправді це одне і теж. Як людина, що має два прізвища...

Викладені теоретичні відомості уважно простежте в прикладах, що наведені в додатку до цієї лабораторної роботи.

Будь-яка програма має певний блок команд, за допомогою яких вона веде діалог з користувачем. Ці команди створюють так званий інтерфейс користувача. В залежності від того, наскільки грамотно він розроблений, людина, працюючи з програмою відчуває певний комфорт або навпаки незручність. Чим простіший діалог веде програма, тим важче зрозуміти, що треба робити. (Згадайте свої перші програми, коли у вас на екрані з'являлась найпростіша позначка для введення інформації, або взагалі її не було, якщо Ви використовували лише оператор **Readln**).

І навпаки, чим більше Ви використовували часу на створення діалогового блоку програми, використовуючи оператори **ClrScr**, **Writeln**, **GoToXY** тощо, тим приємніше і легше ставало працювати з нею.

В цій лабораторній роботі необхідно до електронного довідника (лаб.роб.10-11) замість старого меню створити нове, в якому б режими роботи обирались лише за допомогою клавіш пересування курсору та клавіші **Enter**.

ХІД ВИКОНАННЯ РОБОТИ :

1. Скопіювати блок з програмами до цієї лабораторної роботи в файл **MENU.PAS**.

2. Знищити фігурні дужки коментарів навколо оператора **ShowResults('Анальгин');**.

Уважно дослідити роботу цієї процедури за допомогою **TURBO-DEBUGGER**'а. З'ясувати, яким чином спрацьовують оператори **Window**, **TextBackGround**, **ClrScr**, **TextColor**, **GoToXY** та рядок **(28-12-Length(st)) div 2 + 1**, що центрує стрічку **st**.

3. Знову закрити фігурними дужками оператор `ShowResults('Анальгин')`; і відкрити наступний оператор `stt := GetStr('Введіть назву')`; Виконайте аналогічне дослідження. Зверніть увагу на те, що при виході з функції `GetStr` її значення передається в змінну `stt` основної програми.

4. Провести аналогічне дослідження з функцією `Option`.

5. Послідовно відкриваючи процедури `Alarm`, `Alarm1`..`Alarm4` і запускаючи програму в режимі прогону, проведіть дослідження, які звуки видають ці процедури. Проаналізуйте їх текст.

6. Використовуючи принципи побудови з прикладу, розробіть своє меню та підключіть його до своєї програми електронного довідника. Процедури звуків, рамок, меню скопіюйте з цієї методички до свого файлу з програмою.

7. Розробіть процедури і функції для введення-виведення інформації з електронного довідника. Зробіть заміну абсолютних координат з прикладів на відносні за допомогою додаткових змінних типу `X1`, `X2`, `Y1`, `Y2` і т.і. Наприклад, якщо у вас є оператори :

```
Window(10,15,30,17);  
GoToXY((30-10-Length(st)) div 2 + 1, 1) ...
```

то їх краще переписати таким чином :

```
Window(X1,Y1,X2,Y2);  
GoToXY((X2-X1-Length(st)) div 2 + 1, 1) ...
```

В такому випадку, змінивши на початку процедури `X1`, `X2`, `Y1`, `Y2`, можна буде просто пересувати цілком все вікно, не переписуючі повністю весь текст процедури.

8. Додайте в створений довідник звукові процедури за вашим смаком.

ПИТАННЯ САМОКОНТРОЛЮ:

1. Чим відрізняється процедура від функції ?
2. Якими двома способами можна передавати значення в процедури та функції ?
3. Чи можна викликати одну процедуру (або функцію) з іншої ?
4. Чим відрізняється функція від процедури ? Які типи параметрів може повертати функція ?
5. В чому полягає різниця в цих двох записах :

```
Procedure MMM( Numer : integer );  
Procedure MMM( var Numer : integer ); ?
```
6. Як працюють оператори `Window`, `TextBackGround`, `ClrScr`, `TextColor`, `GoToXY`, `WhereX`, `WhereY` ?
7. Як перетворюються координати на екрані після оператора `Window` ?
8. Як працює меню вашого довідника ?

ДОДАТОК:

```
uses crt;

var i : byte;
stt : string;

Procedure Alarm;
{ Подача звукового сигнала }
var i : Byte;
begin
  for i:=1 to 2 do begin
    Sound( 1760 );
    Delay( 100 );
    Sound( 1047 );
    Delay( 100 );
    Sound( 1318 );
    Delay( 100 )
  end;
  NoSound;
end; { Alarm }

Procedure Alarm1;
{ Подача звукового сигнала }
var i, j : Integer;
Begin
  for i:=1 to 10 do begin
    Sound(300+i*50 );
    Delay( 20 );
    Sound(370+j*60 );
    Delay( 27 )
  end;
  NoSound;
end; { Alarm1 }

Procedure Alarm2;
{ Подача звукового сигнала }
var i : Word;
begin
  i:=3;
  while(i<300) do begin
    i:=i+7;
    sound(10*i);
    delay(5);
    nosound;
  end;
end; { Alarm2 }

Procedure Alarm3;
Begin
  Sound(1230);
  Delay(100);
  NoSound;
  Delay(100);
  Sound(2001);
  Delay(100);
  NoSound;
  Delay(100);
  Sound(700);
  Delay(100);
```



```

    NoSound;
end;

Procedure Alarm4;
var i : Byte;
begin
    for i:=1 to 40 do
    begin;
        sound(440+i*45);
        delay(15);
        nosound;
        delay(9);
    end;
    sound(700);
    delay(200);
    nosound;
end; { Alarm4 }

Procedure Pad;
{процедура, що створює рамки для програмного меню}
begin
    TextBackGround ( Black ); ClrScr;
    Window ( 10, 3, 72, 19 ); TextBackGround ( White ); ClrScr;
    Window ( 16, 5, 67, 17 ); TextBackGround ( Blue ); ClrScr;
    TextBackGround(3); Window(1,1,80,25);
    TextColor(1);
    GotoXY ( 11, 24 );
    Write('Приклад створення програмн. меню.Вверх/Вниз/Enter' );
    TextColor( White ); TextBackGround ( Blue );
end; { Pad }

Function Option : integer;
{ Виведення головного меню і вибір режиму роботи }
Var
i : integer;
Ch : Char;
St : array[1..5] of string;
BEGIN
    Pad; {намалюємо рамки до меню}

    St[ 1 ]:=' Режим 1 ';
    St[ 2 ]:=' Режим 2 ';
    St[ 3 ]:=' Режим 3 ';
    St[ 4 ]:=' Режим 4 ';
    St[ 5 ]:=' В И Х І Д ';

    for i:=1 to 5 do begin
        GotoXY( 20, 5 + i*2 ); Write( St[i] )
    end;

    { ----- }
    GotoXY( 18, 7 ); Write( '>' );
    While True do begin
        Ch := ReadKey;
        Case Ch of
        { ----- }
        'P': begin { Пересування стрілки-вказівника вверх }
            GotoXY( WhereX-1, WhereY );
            Write(' ');
            if ( WhereY<15 ) then
                begin

```



```

        GotoXY( WhereX-1, WhereY+2 );
        Write( '>' )
    end else begin
        GotoXY( 18, 7 );
        Write( '>' )
    end
end;
end;
'H' : begin { Пересування стрілки-вказівника вниз }
        GotoXY( WhereX-1, WhereY ); Write( ' ' );
        if ( WhereY>7 ) then begin
            GotoXY( WhereX-1, WhereY-2 );
            Write( '>' )
        end else begin
            GotoXY( 18, 15 );
            Write( '>' )
        end
    end;
end;
#13      : begin { Вибір режиму, реєстрація опції }
        Option :=(WhereY-5) div 2;
        Exit;
    end;
end; { Case }
end; { While }

END; { Options }

Function GetStr( st : string; x, y : byte ) : string;
{ця функція дорівнює введеному з клавіатури рядку. Малюється
кольорова рамка, в верхній частині якої виводиться підказка з
рядку st. X та Y - це координати лівого верхнього кута рамки на
екрані}

var t : string;
begin

    {визначимо розміри зовнішнього вікна і зафарбуємо його
    кольором фону }

    Window(x, y, x+30 ,y+2); TextBackGround(Green); ClrScr;

    {встановимо курсор на рамці вікна так, щоб слово
    розташувалось по її центру. Зверніть увагу, що функція Length
    дає довжину рядка st. div 2 - це ціле ділення на 2}

    GoToXY((30-Length(st)) div 2 + 1, 1);

    {далі зрозуміло ...}
    TextBackGround(Black); Write(st);

    {визначимо розміри внутрішнього вікна і зафарбуємо його
    кольором іншого фону }

    Window(x+2,y+1,x+28,y+1); TextBackGround(Black); ClrScr;
    {а тепер введемо значення рядку st в другому вікні}
    TextColor(Yellow); GoToXY(x+3,y+1); Readln(t);

    {передамо функції GetStr значення рядку t}
    GetStr := t;

    {перед виходом з процедури перейдемо на максимальний розмір
    вікна}

```



```

Window(1,1,80,25);
end; {getstr}

```

```

Procedure ShowResults( stt : string; x,y:byte );
var    z : byte;
{ця процедура виводить значення рядку st в кольоровій рамці. x
та y - це координати лівого верхнього кута рамки на екрані}
begin
    {визначимо розміри майбутнього вікна на підставі довжини
    рядку stt}
    z := Length(stt)+4;
    {визначимо розміри зовнішнього вікна і зафарбуємо його
    кольором фону }
    Window(x,y,x+z,y+2); TextBackGround(Green); ClrScr;
    {визначимо розміри внутрішнього вікна і зафарбуємо його
    кольором іншого фону }
    Window(x+2,y+1,x+z-2,y+1); TextBackGround(Black); ClrScr;
    {а тепер введемо значення рядку st в другому вікні, зверніть
    увагу, що тут позиція курсора рахується аналогічно}
    GoToXY(1,1);
    TextColor(Yellow); Write(stt); Readln;
    {перед виходом з процедури включимо максимальний розмір
    вікна}
    Window(1,1,80,25);
end;

```

```

BEGIN

```

```

    {поступово познімайте дужки коментарів з наступних операторів
    та дослідіть як працює програма}
    {ShowResults('Анальгин', 1,1);}
    {stt := GetStr('Введіть назву', 5,6);}
    {i := Option;}
    {Alarm;}
    {Alarm1;}
    {Alarm2;}
    {Alarm3;}
    {Alarm4;}

```

```

END.

```


ПИТАННЯ ДО ІСПИТУ З КУРСУ :

КОМП'ЮТЕРНА ТЕХНІКА ТА ПРОГРАМУВАННЯ (ІНФОРМАТИКА ТА КОМП'ЮТЕРНА ТЕХНІКА)

1. Мета та завдання курсу.
2. Комп'ютер як електронний автомат. Принцип виконання команд.
3. Узагальнена структура комп'ютера. Призначення його основних вузлів.
4. Електронні компоненти комп'ютерної техніки. Типи процесорів і комп'ютерів.
5. Електронні компоненти комп'ютерної техніки. Класифікація гнучких носіїв інформації та їх основні характеристики.
6. Електронні компоненти комп'ютерної техніки. Накопичувачі на жорстких магнітних дисках та їх основні параметри.
7. Носії інформації. Принцип розташування інформації на дискових носіях. Запис та зчитування інформації.
8. Порти комп'ютерів. Додаткові зовнішні пристрої (периферія).
9. Класифікація друкуючих пристроїв.
10. Операційна система MS DOS. Функції та її склад. Основні файли ОС MS - DOS.
11. Структура файлів. Їх назва, розширення, дата тощо.
12. Каталогова система ОС MS DOS.
13. Принцип розташування файлів на магнітних носіях. Призначення секторів та службових зон (BOOT, FAT, ROOT).
14. Принцип читання файлів з диску.
15. Принцип запису файлів на диск.
16. Основні команди операційної системи MS DOS.
17. Принцип виконання команд операційною системою. Програмні файли.
18. Поняття про алгоритмічні мови програмування.
19. Трансляція програм. Інтерпретатори та компілятори, наладчики.
20. Абетка, синтаксис та семантика TURBO-PASCAL.
21. Змінні та їх типи. Властивості змінних.
22. Оператор присвоювання.
23. Алгебраїчні вирази. Пріоритет операцій.
24. Оператор виведення інформації та правила його використання.
25. Оператор введення інформації та правила його використання.
26. Основні арифметичні функції TURBO-PASCAL та приклади їх використання.
27. Оператор безумовного переходу та мітки.
28. Оператори розгалуження програм та приклади їх застосування.
29. Селектор CASE та приклад його застосування.
30. Логічні операції та порядок їх виконання.

31. Оператор циклу FOR та принцип його роботи. Блок-схема циклів при збільшенні та зменшенні лічильника цикла.
32. Оператор циклу REPEAT-UNTIL та принцип його роботи. Блок-схема циклів при збільшенні та зменшенні лічильника цикла.
33. Оператор циклу WHILE-DO та принцип його роботи. Блок-схема циклів при збільшенні та зменшенні лічильника цикла.
34. Поняття про масиви та їх види. Індеси масивів. Правила присвоєння при роботі з масивами.
35. Рядкові та символьні змінні. Оператори роботи з рядковими змінними та принцип їх дії.
36. Текстовий режим екрану. Основні функції керування курсором, кольорами, виведенням та редагуванням інформації в текстовому режимі.
37. Текстовий режим екрану. Вікна та принципи роботи з ними.
38. Види графічних режимів та оператори що їх вмикають\вимикають. Система координат в графічному режимі.
39. Основні графічні оператори та їх принцип дії.
40. Оператори встановлення типу тексту та його виведення на екран в графічному режимі.
41. Поняття про текстовий файл. Оператори та принципи роботи з текстовими файлами.
42. Типізовані файли. Принцип зберігання інформації в таких файлах, особливість використання операторів Reset та Rewrite.
43. Оператори запису та читання інформації з типізованих файлів.
44. Модульність програми. Принципи створення та викликання процедур.
45. Модульність програми. Принципи створення та викликання функцій.
46. Передача параметрів в процедуру та з неї. Глобальні параметри.
47. Передача параметрів в функцію та з неї.
48. Перспективи використання обчислювальної техніки в агропромисловому комплексі України.
49. В чому полягає метод послідовного пошуку екстремальних елементів при сортуванні масивів?
50. В чому полягає метод "бульбашковий метод" при сортуванні масивів?

ЛІТЕРАТУРА

1. Фигурнов В.Э. IBM PC для пользователя. - М.: Инфра - М, 1995.
2. Д.Прайс, Программирование на персональных ЭВМ в среде Турбо Паскаль. - М.: МГТУ, 1992.
3. Петров А.В, Вычислительная техника и программирование. - М.: Высшая школа, 1990.
4. Бартків А.В., Гринчишин Я.Т. та ін., TURBO PASCAL: алгоритми і програми, - К.: Вища школа, 1992.
5. Жалдак М.І., Рамський Ю.С., Інформатика: Навчальний посібник. - К.: Вища школа, 1991.