

**Міністерство освіти і науки України
Дніпропетровський національний університет
ім. Олеся Гончара**

**ЛАБОРАТОРНИЙ ПРАКТИКУМ
ІЗ ПРОГРАМУВАННЯ ТА ЗАСТОСУВАННЯ
АЛГОРИТМІЧНИХ МОВ**

2009

**Міністерство освіти і науки України
Дніпропетровський національний університет
ім. Олеся Гончара**

Кафедра обчислювальної механіки і міцності конструкцій

**ЛАБОРАТОРНИЙ ПРАКТИКУМ
ІЗ ПРОГРАМУВАННЯ ТА ЗАСТОСУВАННЯ
АЛГОРИТМІЧНИХ МОВ**

**Дніпропетровськ
РВВ ДНУ
2009**

Лабораторний практикум містить матеріал, необхідний для виконання студентами лабораторних робіт із програмування. Наведені приклади розв'язання типових задач мовами Паскаль і Фортран, подані варіанти індивідуальних завдань, визначення та основні конструкції.

Для студентів ДНУ першого курсу механіко-математичного факультету та факультету прикладної математики.

Темплан 2009, поз. 23

**Лабораторний практикум
із програмування та застосування
алгоритмічних мов**

Укладачі: канд. фіз.-мат. наук, доц. Е. Л. Гарт
асист. Ю. О. Лецер

Редактор Г. В. Кіц
Техредактор Л. П. Замятіна
Коректор Г. В. Кіц

Підписано до друку 27.01.09. Формат 60×84/16. Папір друкарський.
Друк плоский. Ум. друк. арк. Ум. фарбовідб. Обл.-вид. арк. Тираж 100 пр.
Зам. № 114

РВВ ДНУ, просп. Гагаріна, 72, м. Дніпропетровськ, 49010.
Друкарня ДНУ, вул. Наукова, 5, м. Дніпропетровськ, 49050

ВСТУП

Сучасному механіку-досліднику часто доводиться мати справу із задачами, що вимагають від нього гарної комп'ютерної підготовки і твердих навичок і умінь у застосуванні різноманітних математичних методів під час використання комп'ютерної техніки. Для формування у майбутніх фахівців в галузі механіки професійних навичок та умінь самостійно розв'язувати складні практичні та наукові задачі за допомогою персонального комп'ютера (ПК) на лабораторний практикум із програмування та застосування алгоритмічних мов накладаються такі завдання:

- закріплення знань з лекційного курсу з основ програмування та алгоритмічних мов;
- формування практичних навичок з розробки та програмної реалізації проектів розв'язування задач на ПК;
- навчання роботі з науково-технічною літературою та технічною документацією, а також формування навичок розробки і оформлення окремих розділів програмної документації;
- набуття знань про сучасні методи розв'язання задач за допомогою ПК.

Під час виконання кожної лабораторної роботи студент обирає вихідні дані свого варіанту завдання із таблиць або із тексту згідно із індивідуальним шифром. Наприклад, якщо шифр 541703, то дані першого стовпчика таблиці беруться із п'ятого рядка цієї таблиці, дані другого стовпчика – із четвертого рядка і так далі. Якщо таблиця містить менше шести стовпчиків, то зайві цифри відкидаються, а якщо їх більше, то цифри шифру слід повторити, починаючи з першої.

Виконання кожного завдання складається:

- із аналізу поставленого завдання та визначення вимог щодо програми;
- розробки алгоритму розв'язування задачі на ПК, складання структурних блок-схем;
- програмної реалізації алгоритму, тестування та налагодження розробленої програми та складання інструкції щодо її експлуатації;
- аналізу результатів розрахунків;
- підготовки та оформлення звіту.

Звіт виконується на папері формату А4 і повинен мати наскрізну нумерацію (титульний лист входить до наскрізної нумерації, але не нумерується). Зразок оформлення звіту наведено у додатку А.

Базові мови виконання лабораторних робіт – Turbo Pascal 7.0, Visual Fortran 6.6.

Лабораторна робота 1

ОБЧИСЛЕННЯ СУМИ НЕСКІНЧЕННОГО РЯДУ ІЗ ЗАДАНОЮ ТОЧНІСТЮ

Мета:

- освоїти правила організації циклічного процесу з невідомою кількістю повторень;
- оволодіти навичками побудови структурних схем алгоритмів із використанням рекурентних залежностей та інших ефективних прийомів програмування;
- набути вмінь роботи з різними наборами даних.

Постановка завдання

Розробити структурну схему алгоритму та програму для обчислення суми нескінченного ряду

$$S = \sum_{n=0}^{\infty} u_n$$

зі заданою точністю ε , з урахуванням того, що задана точність вважається досягнутою, якщо черговий складник за абсолютним значенням стане меншим, ніж ε , тобто $|u_n| < \varepsilon$.

Вид складника u_n обирається із таблиці згідно з шифром.

Для написання програми мовою Паскаль використати: 1) оператор безумовного переходу; 2) оператор циклу з параметром; 3) оператор циклу з передумовою; 4) оператор циклу з післяумовою.

Для написання програми мовою Фортран використати: 1) оператор безумовного переходу GOTO; 2) оператор циклу DO; 3) оператор циклу DO WHILE.

Зауваження. У процесі обчислення членів ряду не використовувати операцію піднесення до степеня, а застосовувати рекурентні залежності. В ітераційному циклі не обчислювати функції для незалежних від параметру циклу аргументів. Такі засоби дозволяють суттєво зменшити витрати машинного часу. Якщо не вдається одержати загальний для усіх складових u_n коефіцієнт рекурентності, то треба знайти коефіцієнти рекурентності для окремих елементів кожного складника ряду і застосувати під час виконання поставленого завдання.

Варіанти завдань

№	u_n	C_1	C_2	C_3	C_4	x
1	$\frac{C_1 C_3}{C_2 + C_4}$	$(-1)^n x$	$n!$	$\sin^{n-2} x$	3^{n+1}	0.12; 0.22; 0.32; 0.42
2	$\frac{C_1 + C_3}{nC_2 + C_4}$	$(-1)^{n+1} x^2$	$(2n+1)!$	$\arcsin^{2n} x$	2^{n-1}	0.61; 0.72; 0.83; 0.94

№	u_n	C_1	C_2	C_3	C_4	x
3	$\frac{C_1 C_3}{C_2 + n^2 C_4}$	$(-1)^{n-1} x^3$	$(n+2)!$	$\cos^{n-1} x$	3^{n-1}	0.53; 0.63; 0.73; 0.83
4	$\frac{C_1 C_3 + C_3}{C_4 + n^2 C_2}$	$(-x)^{n+1}$	$(3n)!$	$\arccos^{n+2} x$	2^{n+1}	0.31; 0.35; 0.39; 0.43
5	$\frac{C_1}{C_2 C_3 + C_4}$	$(-x)^{n-1}$	$(2n+2)!$	$\operatorname{tg}^{2n-2} x$	3^{2n+1}	0.11; 0.32; 0.43; 0.54
6	$\frac{C_1 + C_3}{C_4 C_3 + C_2}$	$(-x)^{n+1}$	$(4n)!$	$\operatorname{ctg}^{n-1} x$	3^{-2n}	0.25; 0.35; 0.45; 0.55
7	$\frac{C_1 + C_4}{n^2 C_3 + C_2}$	$(-x)^{n+2}$	$(2n+3)!$	$\operatorname{arctg}^{n-1} x$	2^{-n}	0.16; 0.26; 0.46; 0.66
8	$\frac{C_1 + C_3}{C_4 + C_2 C_3}$	$(-1)^n x^{n-2}$	$(n+3)!$	$\operatorname{arcctg}^n x$	e^{-nx}	0.17; 0.27; 0.37; 0.47
9	$\frac{C_1 + C_4}{n^2 C_3 + C_2}$	$(-1)^{2n} x^{n+1}$	$(2n)!$	$\arccos^{n+1} x$	e^{2nx}	0.38; 0.48; 0.58; 0.68
0	$\frac{C_1 + n C_3}{n^2 C_4 + C_2}$	$(-1)^{n+1} x^n$	$(3n+1)!$	$\arcsin^{n-3} x$	$(n+1)x^5$	0.21; 0.33; 0.45; 0.56

Зміст звіту

1. Постановка завдання.
2. Аналіз завдання (опис алгоритму, який містить рекурентні залежності та структурну блок-схему).
3. Виконати за допомогою калькулятора 4 кроки ітераційного процесу, які демонструють його збіжність.
4. Лістинг програм із результатами розрахунків, оформлених у вигляді таблиці для різних значень ε (0,001; 0,0001; 0,00001; 0,000001) (див. додаток А).

Приклад. Розглянемо наведене завдання для ряду, що збігається:

$$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

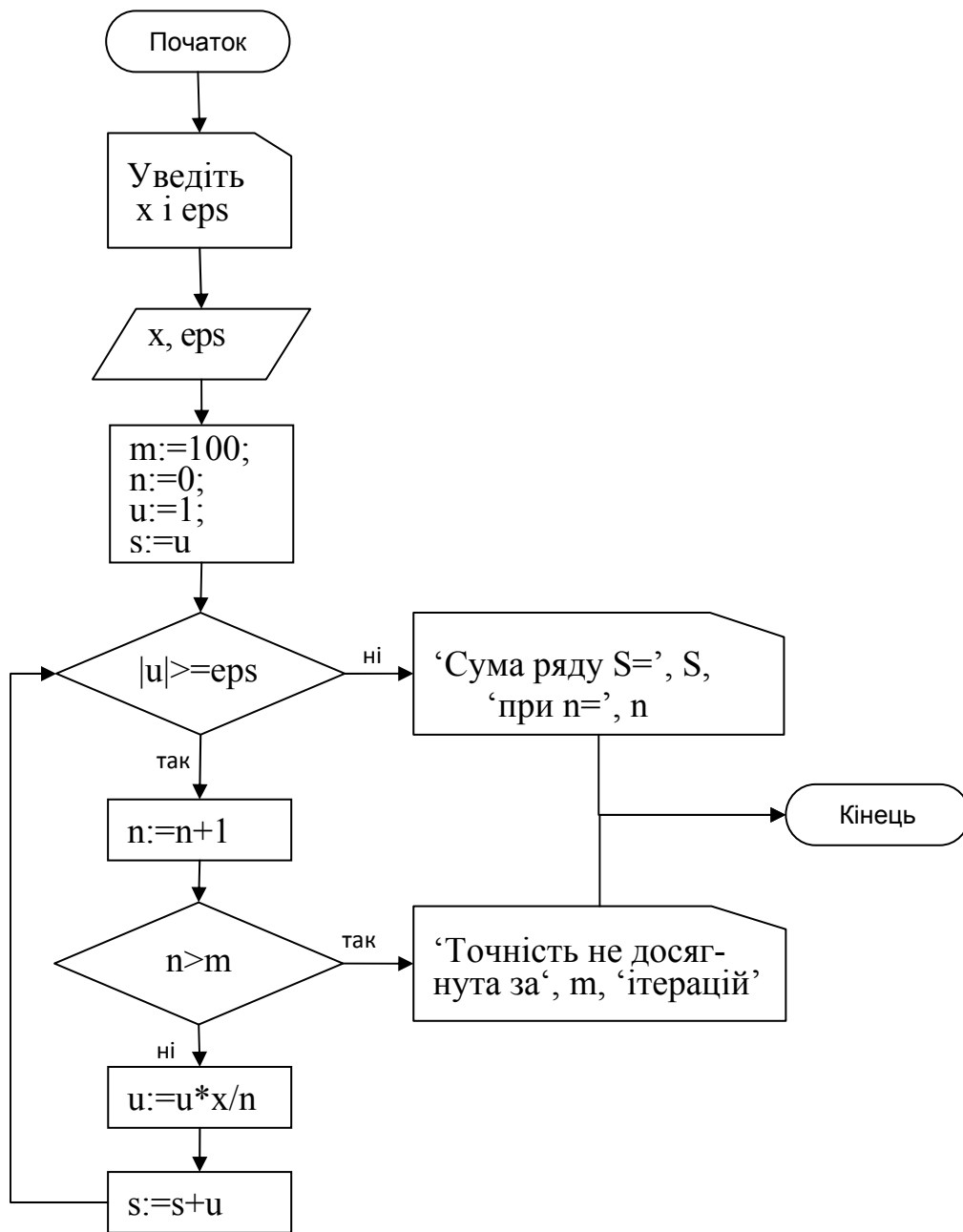
Для обчислення членів ряду застосуємо рекурентну залежність

$$u_n = k_n u_{n-1}, \quad u_0 = 1, \quad n=1, 2, \dots,$$

де $k_n = x/n$ – коефіцієнт рекурентності.

$$\text{Дійсно, при } n=1, 2, \dots \quad u_1 = u_0 \frac{x}{1} = x; \quad u_2 = u_1 \frac{x}{2} = \frac{x^2}{2}; \quad u_3 = u_2 \frac{x}{3} = \frac{x^3}{6}; \dots$$

Блок-схема алгоритму



Лістинг програми (мова Паскаль)

```
Program SUMM; {використання циклу з передумовою}  
Const m=100; {надання константи, що обмежує кількість ітерацій}  
Var x, eps, s, u: real; n: integer; {опис змінних, що використовуються}  
Begin  
  Writeln('Введіть x і eps');  
  Readln(x, eps);
```

```

n:=0; u:=1; s:=u;
While abs(u)>=eps do {перевірка умови виходу із циклу}
Begin
    Inc(n); {модифікація параметру циклу}
    If (n>m) then {запобігання “зациклюванню”}
    Begin
        Writeln(‘Точність не досягнута за’, m, ‘ітерацій’);
        Readln; Exit;
    End;
    u:=u*x/n;
    s:=s+u;
End;
Writeln(‘Сума ряду S=’, S:15:10, ‘при n=’, n);
Readln;
End.

```

Лістинг програми (мова Фортран)

```

*   Застосування циклу з передумовою
    Program SUMM
*   Надання константи, що обмежує кількість ітерацій
    PARAMETER (M=100)
    WRITE(*, *)‘Введіть x і eps’
    READ(*, *) x, eps
    N=0
    U=1.
    S=U
*   Перевірка умови виходу із циклу
    DO WHILE (ABS(U).GE.EPS)
        N=N+1
*   Запобігання “зациклюванню”
    IF (N>M) THEN
        WRITE(*, *) ‘Точність не досягнута за’, m, ‘ітерацій’

```



```

STOP
ELSE
U=U*X/N
S=S+U
ENDIF
ENDDO
WRITE(*, 33) S, N
33 FORMAT('Сума ряду S=', F15.10, 'при n=', I3)
STOP
END

```

Лабораторна робота 2

ДІЇ НАД МАТРИЦЯМИ

Мета:

- освоїти правила організації масивів;
- оволодіти навичками використання змінних з індексами.

Постановка завдання

Розробити структурну схему алгоритму та програму виконання дій над матрицею

$$A = \begin{pmatrix} 1.6 & 3.4 & -5.2 & 3.2 & 4.1 \\ 2.1 & 3.3 & -4.5 & -5.0 & -7.4 \\ -3.0 & -8.1 & 2.2 & -1.3 & 5.0 \\ 7.2 & -9.0 & 1.3 & 4.0 & 1.0 \\ 5.5 & 3.4 & 7.2 & -5.0 & 4.3 \end{pmatrix} + \frac{XX}{10} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

де *XX* – перші дві цифри шифру відповідно до нижченаведених пунктів.

I. Знайти і видати на екран і в файл значення:

- 1) сум модулів елементів кожного стовпчика;
- 2) сум модулів елементів кожного рядка;
- 3) сум елементів кожного стовпчика;
- 4) сум елементів кожного рядка;
- 5) добутків елементів кожного стовпчика;
- 6) добутків елементів кожного рядка;
- 7) квадратних коренів із сум квадратів елементів кожного стовпчика;
- 8) квадратних коренів із сум квадратів елементів кожного рядка;
- 9) квадратів сум квадратних коренів із модулів елементів кожного стовпчика;
- 0) квадратів сум квадратних коренів із модулів елементів кожного рядка.

II. Знайти і видати на екран і в файл значення:

- 1) середнього арифметичного найбільших елементів кожного стовпчика;
- 2) середнього арифметичного найменших елементів кожного рядка;
- 3) середнього геометричного модулів найбільших за модулем елементів кожного стовпчика;
- 4) середнього геометричного модулів найменших за модулем елементів кожного рядка;
- 5) добуток найбільших елементів головної діагоналі та третього рядка;
- 6) суми найменших елементів головної діагоналі та третього рядка
- 7) найбільшого за модулем елемента матриці (при цьому надрукувати його координати);
- 8) найменшого елемента матриці (при цьому надрукувати його координати);
- 9) суми квадратів найбільшого елемента першого стовпчика та найменшого за модулем елемента останнього рядка;
- 0) добутку найменшого за модулем елемента третього рядка і найбільшого елемента другого стовпчика.

III. Обчислити матрицю B , яка визначається за формулами

- | | |
|--------------------------------------|--------------------------------------|
| 1) $B = A \cdot A^T - A$; | 6) $B = (A^T - A) \cdot A^T$; |
| 2) $B = A - A^T \cdot A$; | 7) $B = A^T \cdot A^T - A \cdot A$; |
| 3) $B = A^T \cdot A - A \cdot A^T$; | 8) $B = (A - A^T) \cdot A$; |
| 4) $B = A^T \cdot A \cdot A^T$; | 9) $B = (A^T - A) \cdot (A + A^T)$; |
| 5) $B = A \cdot A^T \cdot A$; | 0) $B = A \cdot (A^T - A)$. |

IV. Провести над матрицею A вказані нижче операції:

- 1) замінити всі від'ємні елементи їх модулями;
- 2) замінити всі додатні елементи квадратними коренями із них;
- 3) поміняти місцями елементи симетричні відносно побічної діагоналі;
- 4) поміняти місцями найбільший елемент першого рядка і найменший елемент останнього стовпчика;
- 5) поміняти місцями найбільший за модулем елемент останнього рядка і найменший за модулем елемент першого стовпчика;
- 6) поміняти місцями найбільший і найменший елементи матриці;
- 7) поміняти місцями третій рядок і третій стовпчик;
- 8) поміняти місцями діагоналі матриці;
- 9) замінити другий рядок сумою першого й останнього стовпчиків;
- 0) замінити останній стовпчик часткою першого і третього рядків.

Зміст звіту

1. Постановка завдання.
2. Аналіз завдання (опис структурної блок-схеми алгоритму).
3. Виконані дії над вихідною та тестовою матрицями, які задані:
 - у файлі;
 - програмі як константи.
4. Протокол роботи програми (вхідні дані та результати розрахунків).
5. Лістинг програми.

Приклад. Написати програму перемноження двох дійсних матриць A і B 3-го порядку.

$$C = A \cdot B = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix},$$

$$C = \{c_{ij}\}, c_{ij} = \sum_{k=1}^N a_{ik} b_{kj}, \quad N = 3.$$

Лістинг програми (мова Паскаль)

```
Program MULT; {перемноження двох матриць}
Const n=3; {завдання виміру матриць}
Type Matr=array[1..n,1..n] of real; {завдання двовимірного масиву}
{визначення вихідних матриць}
const A: Matr=((1.1,1.1,1.1),(2.1,2.1,2.1),(3.1,3.1,3.1));
      B: Matr=((4.1,4.1,4.1),(3.1,3.1,3.1),(4.1,4.1,4.1));
{опис змінних, що використовуються}
var C: Matr; s: real; i,j: integer;
begin
  for i:=1 to n do
    for j:=1 to n do
      begin
        S:=0;
        for k:=1 to n do
          S:=S+A[i,k]*B[k,j];
        C[i,j]:=S;
      end;
    writeln ('C=AB');
```

```

for i:=1 to n do
begin
    for j:=1 to n do
        write (C[i,j]:4:1);
        writeln;
    end;
readln;
end.

```

Лістинг програми (мова Фортран)

```

*   Перемноження двох матриць A і B
PROGRAM MULT
*   Надання константи, що визначає порядок матриць
PARAMETER (N=3)
REAL A(N,N), B(N,N), C(N,N)
*   Визначення вихідних матриць
DATA ((A(I,J),J=1,N),I=1,N)/ 1.1, 1.1, 1.1, 2.1, 2.1, 2.1, 3.1,
A      3.1, 3.1/
DATA ((B(I,J),J=1,N),I=1,N)/ 4.1, 4.1, 4.1, 3.1, 3.1, 3.1, 4.1,
A      4.1, 4.1/
DO 20 I=1,N
DO 20 J=1,N
S=0.
DO 30 K=1,N
S=S+A(I,K)*B(K,J)
30  CONTINUE
C(I,J)=S
20  CONTINUE
WRITE(*, 100) ((A(I,J),J=1,N),I=1,N)
100  FORMAT('Матриця A' /(3X,<N>F4.1, 2X))
WRITE(*, 200) ((B(I,J),J=1,N),I=1,N)
200  FORMAT('Матриця B' /(3X,<N>F4.1, 2X))
WRITE(*, 300) ((C(I,J),J=1,N),I=1,N)
300  FORMAT('Результат перемноження матриць C=AB'/(3X,<N>F4.1, 2X))
END

```

Лабораторна робота 3

ЗАСТОСУВАННЯ ПРОЦЕДУР І ФУНКЦІЙ. РЕКУРСІЯ

Мета:

- розвинути навички структурного програмування;
- вивчити різні аспекти застосування процедур і функцій у практиці програмування.

Теоретичні відомості

Процедури та функції служать для здійснення однотипних дій.

Загальне визначення процедури мови Паскаль: це практично довільний складений оператор, винесений із основного блоку програми та оголошений як

```
Procedure <ім'я_процедури>;
```

В основному же блоці замість складеного оператора підставляється одне лише <ім'я_процедури>. Процедура може мати параметри, мітки переходу всередині себе і свої локальні змінні. Обов'язкові елементи процедур і функцій – заголовок і тіло.

```
Procedure <ім'я_процедури> (<параметр_знач1>:<тип1>; (<параметр_знач2>: <тип2>; Var <параметр_змін1>:<тип1>; Var <параметр_змін2>: <тип2>);
```

```
label {мітки всередині тіла процедури};
```

```
const {локальні константи};
```

```
type {локальні типи};
```

```
var {локальні змінні};
```

```
begin
```

```
{тіло процедури}
```

```
end;
```

Виклик процедури: <ім'я_процедури>(<параметр1>, <параметр2>,...);

У дужках знаходиться список фактичних параметрів, тобто змінних і значень, які підставляються на місце формальних параметрів процедури.

Загальна структура функції:

```
Function <ім'я_функції> (<список параметрів>): <ім'я_типу_функції>;
```

Порядок розташування розділів описів здійснюється за правилами побудови всієї програми.

Мова Паскаль дозволяє будувати рекурсивні алгоритми. *Рекурсією* вважається виклик функції (процедури) із тіла тієї ж функції (процедури).

Приклад 1. Обчислення факторіала за формулою

$$n! = \begin{cases} 1, & \text{якщо } n = 0, \\ n(n-1)!, & \text{якщо } n > 0. \end{cases}$$

Рекурсивна функція має вигляд
function FACT (n: Word): LongInt;

```

begin
    if n=0 then FACT:=1
    else FACT:=n*FACT(n-1);
end;

```

Можливе застосування ітераційного алгоритму, який дозволяє економити пам'ять.

```

function FACT2 (n: Word): LongInt;
var i: Word, m: LongInt;
begin
    m:=1;
    for i:=1 to n do m:=m*i;
    FACT2:=m;
end;

```

Постановка завдання

1. Оформіть завдання лабораторної роботи 2 у вигляді двох процедур і двох функцій з параметрами. Організувати виклик цих підпрограм із однієї основної програми.

2. Згідно з номером за списком групи обрати по одному з обох частин наведених завдань (завдання II частини слід виконувати, реалізуючи рекурсивний алгоритм).

Варіанти завдань

I частина

1. Процедура **maxmin(x,y)** присвоює параметру x більше із дійсних чисел x, y, а параметру y – менше. Описати таку процедуру й застосувати її для перерозподілу значень дійсних змінних a,b,c так, щоб стало $a \geq b \geq c$.

2. const n=10;
type vector=array[1..n]of real;
var a,b,c: vector;

Процедура **sum(x,y,z)** присвоює вектору z суму векторів x, y. Описати таку процедуру та застосувати її для обчислення $d=a+b+c$.

3. Використовуючи задані дійсні числа $a_0, a_1, \dots, a_{10}; b_0, b_1, \dots, b_{10}; c_0, c_1, \dots, c_{10}; x, y, z$ обчислити величину

$$\frac{(a_0x^{10} + a_1x^9 + \dots + a_{10})^2 - (b_0y^{10} + b_1y^9 + \dots + b_{10})}{c_0(x+z)^{10} + c_1(x+z)^9 + \dots + c_{10}}$$

Описати процедуру для введення вектора та функцію для обчислення полінома з коефіцієнтами d_i ($i = 0, 10$) в точці t.

4. За заданими 10-елементними цілими масивами x, y обчислити

$$u = \begin{cases} \sum_{i=1}^{10} x_i^2 & \text{при } \sum_{i=1}^5 x_i y_i > 0, \\ \sum_{i=1}^{10} y_i^2 & \text{при } \sum_{i=1}^5 x_i y_i \leq 0. \end{cases}$$

Обчислення здійснити за допомогою процедури.

5. Дано три дійсні матриці 4-го порядку. Надрукувати ту з них, норма якої найменша (вважати, що така матриця одна). За норму матриці взяти максимум абсолютних величин її елементів.
6. Дано дві квадратні дійсні матриці 5-го порядку. Надрукувати квадрат тієї з них, в якій найменший слід (сума діагональних елементів), вважаючи, що така матриця одна.
7. Дано координати вершин двох трикутників. Визначити, який із них має більшу площу.
8. За заданими 10-елементними дійсними масивами a, b, c обчислити

$$t = \begin{cases} \frac{\min(b_i)}{\max(a_i)} + \frac{\max(c_i)}{\min(b_i + c_i)} & \text{при } \min(a_i) < \max(b_i), \\ \max(b_i + c_i) + \min(c_i) & \text{при } \min(a_i) \geq \max(b_i). \end{cases}$$

Обчислення здійснити за допомогою процедури.

9. Два прості числа називаються “близнюками”, якщо вони відрізняються одне від одного на 2 (наприклад, 41 і 43). Надрукувати всі пари “близнюків” із відрізка $[n, 2n]$, де n – задане ціле число, більше 2.
10. Дано коефіцієнти поліномів $P(x)$ і $Q(x)$ 15-го ступеня і дано дійсне число a . Обчислити величину $P(a + Q(a)P(a + 1))$.
11. Дано три цілі матриці розміром 9×4 . Надрукувати ту з них, в якій більше нульових рядків (якщо таких матриць декілька, надрукувати їх усі).
12. Дано натуральне число p і дійсні квадратні матриці A, B, C 4-го порядку. Одержати $(ABC)^p$.
13. Дано дві цілі квадратні матриці 6-го порядку. Визначити, чи можна відображеннями відносно головної та побічної діагоналей перетворити одну з них на іншу.
14. Дано коефіцієнти поліномів $P(x)$ і $Q(x)$ 10-го ступеня і дано дійсне число b . Обчислити величину $Q(b - P(b)Q(2b + 1))$.
15. Дано координати вершин двох рівнобічних трапецій. Визначити, яка з них має меншу площу.
16. Дано три цілі квадратні матриці 5-го порядку. Надрукувати ту з них, в якій більше від’ємних елементів.

17. Дано три дійсні квадратні матриці 4-го порядку. Надрукувати середнє арифметичне значення максимальних елементів головних діагоналей.
18. За заданими 10-елементними цілими масивами x , y обчислити

$$u = \begin{cases} \sum_{i=1}^{10} 2x_i & \text{при } \sum_{i=1}^5 x_i + y_i \geq 0, \\ \sum_{i=1}^{10} 2y_i & \text{при } \sum_{i=1}^5 x_i + y_i < 0. \end{cases}$$

Обчислення здійснити за допомогою процедури.

19. Процедура **mult(x,y,z)** присвоює вектору z добуток векторів x , y . Описати таку процедуру й застосувати її для обчислення вектора $q=abc$, де a , b , c – задані вектори із 15 елементів.
20. Дано натуральне число n і дійсні квадратні матриці A , B , C 3-го порядку. Одержати $(A + B + C)^{2n}$.

II частина

- Надрукувати в зворотному порядку заданий у вхідному файлі текст (за текстом іде точка).
- Обчислити значення виразу

$$5 + 1/(3 + 1/(3 + 1/(3 + 1/(3 + 1/(3 + 1/(3 + 1/3))))))$$
- У вхідному файлі записано (без помилок) таку формулу:
 <формула> ::= <цифра> | (<формула><знак><формула>);
 <знак> ::= + | - | *;
 <цифра> ::= 0|1|2|3|4|5|6|7|8|9.
 Ввести цю формулу й обчислити її значення. (Наприклад, 6→6, ((3-1)*5)→10, ((2-7)*(4+5))→-45).
- Дана послідовність ненульових цілих чисел, за якою йде 0. Надрукувати спочатку всі від'ємні числа цієї послідовності, а потім – всі додатні (у будь-якому порядку).
- У вхідному файлі дано текст, за яким іде точка. Перевірити, чи є цей текст правильним записом «формули»:
 <формула> ::= <цифра> | (<формула><знак><формула>);
 <знак> ::= + | - | *;
 <цифра> ::= 0|1|2|3|4|5|6|7|8|9.
 (Наприклад, 5; (5+2); ((5-30)*4)).
- type stroka=array[1..100] of char;
 Описати рекурсивну логічну функцію **sum(s,i,j)**, яка перевіряє, чи є симетричною частина рядка s , яка починається i -м елементом і закінчується j -м її елементом.

7. У вхідному файлі дано текст, за яким іде точка. Перевірити, чи задовольняє його структура таке визначення:
 $\langle \text{текст} \rangle ::= \langle \text{елемент} \rangle \mid \langle \text{елемент} \rangle \langle \text{текст} \rangle;$
 $\langle \text{елемент} \rangle ::= a \mid v(\langle \text{текст} \rangle) \mid [\langle \text{текст} \rangle] \mid \{\langle \text{текст} \rangle\}.$
8. Існує n населених пунктів, перенумерованих від 1 до n ($n=10$). Деякі пари пунктів поєднані дорогами. Визначити, чи можна потрапити цими дорогами із 1-го пункту в n -й. Інформація про дороги подається у вигляді послідовності пар чисел i, j ($i < j$), які указують, що i -й та j -й пункти поєднані дорогою; ознака кінця цієї послідовності – пара нулів.
9. Дано n різних натуральних чисел ($n=5$). Надрукувати всі перестановки цих чисел.
10. У вхідному файлі записано без помилок логічний вираз такого вигляду:
 $\langle \text{логічний вираз} \rangle ::= \text{true} \mid \text{false} \mid \langle \text{операція} \rangle (\langle \text{операнди} \rangle);$
 $\langle \text{операція} \rangle ::= \text{not} \mid \text{and} \mid \text{or};$
 $\langle \text{операнди} \rangle ::= \langle \text{операнд} \rangle \mid \langle \text{операнд} \rangle, \langle \text{операнди} \rangle;$
 $\langle \text{операнд} \rangle ::= \langle \text{логічний вираз} \rangle;$
 В операцій `and` та `or` може бути довільна кількість операндів, у `not` – тільки один. Увести цей вираз і обчислити його значення. Наприклад:
`not(or(false, not(false))), true, not(true)) → false.`
11. Заданий дійсний масив із n різних елементів ($n=100$) упорядкувати за зростанням методом *швидкого сортування*: обрати деякий (наприклад, середній) елемент масиву та переставити елементи цього масиву так, щоб зліва від обраного опинилися тільки менші елементи, а справа – тільки більші (таким чином, обраний елемент опиниться на своєму кінцевому місці), після чого рекурсивно застосувати цей метод до лівої та правої частин масиву.
12. У вхідному файлі задано непусту послідовність додатних дійсних чисел, за якою йде від’ємне число. Описати рекурсивну функцію без параметрів для знаходження суми цих додатних чисел.
13. Задача про 8 ферзей: на шахівниці розташувати 8 ферзей так, щоб вони не “били” один одного. Написати програму, яка друкує одне із таких розташувань.
14. Написати рекурсивну функцію без параметрів, яка підраховує кількість цифр у тексті, який заданий у вхідному файлі (за текстом іде точка).
15. Задана непуста послідовність додатних дійсних чисел, за якою йде від’ємне число. Описати рекурсивну функцію без параметрів для знаходження суми цих додатних чисел.
16. Рекурсивно описати функцію $C(m, n)$, де $0 \leq m \leq n$, для обчислення біноміального коефіцієнта C_n^m за такою формулою:

$$C_n^0 = C_n^n = 1, \quad C_n^m = C_{n-1}^m + C_{n-1}^{m-1}, \quad 0 < m < n .$$

17. Описати функцію **min(x)** для визначення мінімального елемента вектора **x**, вводячи допоміжну рекурсивну функцію **min1(k)**, яка знаходить мінімум серед останніх елементів вектора **x**, починаючи з **k**-го.
18. Процедура **Sk(a,b,p,q)** від цілих параметрів (**b**≠0) приводить дріб **a/b** до нескоротного вигляду **p/q**. Описати таку процедуру та застосувати її для приведення дробу $1+1/2+1/3+\dots+1/20$ до нескоротного вигляду **c/d**.
19. Написати рекурсивну функцію без параметрів, яка підраховує кількість літер 'а' в тексті, заданому у вхідному файлі (за текстом іде точка).
20. У вхідному файлі задано непусту послідовність додатних цілих чисел, за якою йде від'ємне число. Описати рекурсивну функцію без параметрів для знаходження додатка цих додатних чисел.

Зміст звіту

1. Постановка завдання.
2. Аналіз завдання (опис структурної блок-схеми алгоритму).
3. Протокол роботи програми (набір вхідних тестових даних і результати розрахунків).
4. Лістинг програми.

Приклад 2. Обчислити площу чотирикутника ABCD, для якого задані довжини його сторін.

Наведемо площу чотирикутника ABCD як суму площин двох трикутників ABC і ACD. За формулою Герона площа трикутника ABC дорівнює

$$S_{\Delta ABC} = \sqrt{p(p-a)(p-b)(p-c)},$$

де $p = (a+b+c)/2$, $a = AB$, $b = BC$, $c = AC$.

Тоді $S_{\Delta ABC} = S_{\Delta ABC} + S_{\Delta ACD}$.

Лістинг програми (мова Паскаль)

```

program S;
var AB, BC, CD, DA, AC, Pl: real;
function TR(a, b, c: real): real;
var p: real;
begin p:=(a+b+c)/2;
      TR:=sqrt (p*(p-a)*(p-b)*(p-c))
end; {of function}
begin
  read (AB, BC, CD, DA, AC);
  Pl:=TR(AB, BC, AC)+TR(CD, DA, AC)
  write ('Площа чотирикутника ABCD дорівнює' Pl:8:2)
end.

```

Лістинг програми (мова Фортран)

```
PROGRAM S
WRITE(*,*) 'Введіть довжини сторін'
READ(*,*) AB, BC, CD, DA, AC
PL=TR(AB, BC, AC)+TR(CD, DA, AC)
WRITE (*, 10) PL
10  FORMAT(10X, 'Площа чотирикутника ABCD дорівнює', F8.2)
END
*  Підпрограма визначення площі чотирикутника
FUNCTION TR (A, B, C)
P=(A + B + C)/2.
TR=SQRT (P*(P-A)*(P-B)*(P-C))
RETURN
END
```

Лабораторна робота 4 **ДІЇ НАД РЯДКАМИ**

Мета:

- оволодіти правилами використання символічних масивів.

Постановка завдання

Згідно із номером за списком групи обрати три із наведених завдань за правилом: перше завдання – порядковий номер за списком; друге – номер за списком + перший номер шифру; третє – номер другого завдання + другий номер шифру.

Варіанти завдань

- Розробити програму для введення з клавіатури тексту:
- 1) з 50 символів і виводу на екран всіх цифр, які входять в цей текст по одному разу;
 - 2) з 50 символів і виводу на екран цього тексту, замінивши повторні входження кожної літери пропусками;
 - 3) з 50 символів і виводу на екран загальної кількості літер, які входять в цей текст по одному разу;
 - 4) з 50 символів і виводу на екран цього тексту, замінивши повторні входження кожної цифри пропусками;
 - 5) з 50 символів і виводу на екран найбільшої кількості повторних входжень літер;

- 6) з 50 символів і виводу на екран найбільшої кількості повторних входжень цифр;
- 7) з 50 символів і виводу на екран кількості входжень в цей текст рядка 'abc';
- 8) з 50 символів і виводу на екран цього тексту, замінивши всі непарні цифри на парні, наступні по порядку;
- 9) з 50 символів і виводу на екран довжини найбільшого рядка в цьому тексті, що складається лише з цифр;
- 10) з 50 символів і виводу на екран найкоротшого рядка в цьому тексті, обмеженого зліва і справа пропусками;
- 11) з 15 символів і виводу на екран рядка 'YES', якщо цей текст є правильним записом дійсного числа з плаваючою точкою мови Паскаль. Інакше на екран виводиться рядок 'NO';
- 12) з 6 символів і виводу на екран рядка 'YES', якщо цей текст є правильним записом цілого десяткового числа без знаку, кратного 7. Інакше на екран виводиться рядок 'NO';
- 13) з 6 символів і виводу на екран рядка 'YES', якщо цей текст є правильним записом парного числа без знаку в сімковій системі числення. Інакше на екран виводиться рядок 'NO';
- 14) з 10 символів і виводу на екран рядка 'YES', якщо сума числових значень цифр, що входять у цей текст, дорівнює 20. Інакше на екран виводиться рядок 'NO';
- 15) з 20 символів і виводу на екран рядка 'YES', якщо цей текст починається з деякої ненульової цифри, за якою йдуть лише цифри, і їх кількість дорівнює числовому значенню цієї цифри. Інакше на екран виводиться рядок 'NO';
- 16) з 20 символів і виводу на екран рядка 'YES', якщо цей текст починається з k цифр, причому числове значення k -ї цифри дорівнює k . Інакше на екран виводиться рядок 'NO';
- 17) з 8 символів і виводу на екран рядка 'YES', якщо цей текст складається лише з цифр, причому їх числові значення утворюють арифметичну прогресію. Інакше на екран виводиться рядок 'NO';
- 18) з 20 символів і виводу на екран рядка 'YES', якщо числові значення цифр, що входять у цей текст, утворюють арифметичну прогресію. Інакше на екран виводиться рядок 'NO';
- 19) з 10 символів і виводу на екран цілого числа (integer), утвореного з послідовності цифр, що входять у введений текст. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 20) з 10 символів і виводу на екран цілого числа (integer), утвореного з послідовності цифр, що входять у введений текст. Порядок цифр в числі

- змінений на зворотний порівняно з вихідним текстом. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 21) з 20 символів і виводу на екран рядка 'YES', якщо у введеному тексті правильно розставлені круглі дужки (тобто кожній дужці, що закривається, в тексті передує відкриваюча дужка, а за кожною дужкою, що відкривається, йде дужка, що закривається, причому між ними можуть знаходитися літери, відмінні від круглих дужок). Інакше на екран виводиться рядок 'NO';
 - 22) з 50 символів і виводу на екран рядка 'YES', якщо в цей текст послідовно (але не обов'язково безпосередньо один після одного) входять літери 'a', 'b', 'c', 'd', 'e'. Інакше на екран виводиться рядок 'NO';
 - 23) з 50 символів і виводу на екран щонайдовшого рядка з цього тексту, який не містить пропусків;
 - 24) з 50 символів і виводу на екран усіх цифр, які входять в цей текст по два рази;
 - 25) з 10 символів. Якщо введений текст є правильним записом цілого числа без знака в трійковій системі числення, то вивести на екран це число в п'ятірковій системі числення. Інакше на екран виводиться рядок 'NULL';
 - 26) з 50 символів і виводу на екран цього тексту, замінивши повторні входження рядка 'abc' пропусками;
 - 27) з 50 символів і виводу на екран цього тексту, замінивши всі входження рядка 'abc', крім третього, пропусками;
 - 28) з 50 символів і виводу на екран цього тексту, замінивши повторні входження кожної парної цифри пропусками;
 - 29) з 50 символів і виводу на екран середнього арифметичного кількостей входжень парних і непарних цифр;
 - 30) з 10 символів і виводу на екран рядка 'YES', якщо різниця сум числових значень парних і непарних цифр, що входять в цей текст, дорівнює 6. Інакше на екран виводиться рядок 'NO';
 - 31) з 10 символів. Якщо введений текст є правильним записом цілого числа без знака в п'ятірковій системі числення, то вивести на екран це число в системі числення з основою 11. Інакше на екран виводиться рядок 'NULL';
 - 32) з 50 символів і виводу на екран найменшої кількості повторних входжень літер. Якщо в тексті відсутні літери, то на екран виводиться рядок 'NULL';
 - 33) з 50 символів і виводу на екран довжини найменшого рядка в цьому тексті, що складається лише з цифр. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';

- 34) з 8 символів і виводу на екран рядка 'YES', якщо цей текст є правильним записом непарного числа без знака в системі числення з основою 6. Інакше на екран виводиться рядок 'NO';
- 35) з 12 символів і виводу на екран рядка 'YES', якщо цей текст складається лише з цифр, причому їх числові значення утворюють геометричну прогресію. Інакше на екран виводиться рядок 'NO';
- 36) з 10 символів і виводу на екран подвоєного значення цілого числа, утвореного з послідовності цифр, що входять у введений текст. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 37) з 20 символів і виводу на екран потрійного значення цілого числа, утвореного з послідовності цифр, що входять у введений текст. Порядок цифр в числі змінений на зворотний порівняно із вхідним текстом. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 38) з 30 символів і виводу на екран цілого числа, утвореного з послідовності цифр, що входять у введений текст. Цифри в числі упорядкувати за зростанням. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 39) з 20 символів і виводу на екран цілого числа, утвореного з послідовності цифр, що входять у введений текст. Цифри в числі упорядкувати за убаванням. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 40) з 50 символів і виводу на екран усіх цифр, які входять у текст непарну кількість разів;
- 41) з 10 символів. Якщо введений текст є правильним записом цілого числа без знака в двійковій системі числення, то вивести на екран це число у вісімковій системі числення. Інакше на екран виводиться рядок 'NULL';
- 42) з 50 символів і виводу на екран цього тексту, замінивши повторні входження кожної парної цифри пропусками;
- 43) з 50 символів і виводу на екран “ущільненого” тексту в зворотному порядку, усунувши пропуски;
- 44) з 50 символів і виводу на екран середнього геометричного всіх парних цифр, що входять у текст. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 45) з 50 символів і виводу на екран середнього арифметичного всіх непарних цифр, що входять у текст. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 46) з 20 символів і виводу на екран рядка 'YES', якщо сума числових значень парних і непарних цифр, що входять в цей текст, рівна 18. Інакше на екран виводиться рядок 'NO';
- 47) з 50 символів і виводу на екран усіх літер, які входять у текст, непарну кількість разів;

- 48) з 50 символів і виводу на екран середнього зваженого значення всіх цифр, що входять в текст, обчисленого за правилом відношення суми добутоків цифр на кількість їх входжень в текст до суми цифр. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 49) з 50 символів і виводу на екран цілого числа, одержаного як сума добутоків непарних цифр, що входять в текст, на їх порядкові номери. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL';
- 50) з 50 символів і виводу на екран усіх парних цифр, що входять в текст, в зворотному порядку. Якщо в тексті відсутні цифри, то на екран виводиться рядок 'NULL'.

Зміст звіту

1. Постановка завдання.
2. Аналіз завдання (опис структурної блок-схеми алгоритму).
3. Протокол роботи програми (набір вхідних тестових даних і результати розрахунків).
4. Лістинг програми.

Приклад. Дано рядок із 100 символів. Визначити кількість входжень у цей рядок сполучення 'abc'.

Лістинг програми (мова Паскаль)

```

program Stroka;
const n=100;
type STR=array[1..n] of char;
var b: STR; m, i: integer;
begin
  writeln ('Введіть строку');
  for i:=1 to n do
    read (b[i]);
  for i:=1 to n-2 do
    if ((b[i]='a') and (b[i+1]='b') and (b[i+2]='c')) then
      Inc(m);
  writeln ('Кількість входжень 'abc' у вхідний рядок m=', m);
  readln;
end.

```

Лістинг програми (мова Фортран)

```

PROGRAM STROKA;
PARAMETER (N=100);
CHARACTER B(N)
WRITE(*,*) 'Введіть строку'
READ (*, 10) (B(I), I=1,N)
10  FORMAT (<N>A1)
DO 20 I=1, N-2

```

```

IF (B[I]='A' AND B[I+1]='B' AND B[I+2]='C') M=M+1
20 CONTINUE
WRITE (*, 30) M
30 FORMAT (5X, 'Кількість входжень 'abc' у вхідний рядок m=', I3)
END

```

Лабораторна робота 5

ФАЙЛОВА СИСТЕМА. РОБОТА З ЗАПИСАМИ

Мета:

- вивчити тип “запис” (Record);
- оволодіти навичками роботи з текстовими файлами.

Теоретичні відомості

Запис – це структура даних для компактного подання комбінацій різно-типних даних. Кожний запис складається із оголошеного числа полів. Тип “запис” мови Паскаль визначається конструкцією

```

record
  <поле 1>: <тип поля 1>;
  <поле 2>: <тип поля 2>;
  ...
  <поле N>: <тип поля N>
end;

```

Наприклад, задамо такий тип “запис”

```

type TP= record x, y: integer;
           z: real

```

```

end;

```

```

var Point: TP; {оголошення змінної типу “запис”}

```

До кожного поля можна звернутися так: Point.x, Point.y, Point.z. Поле “записи” може мати практично довільний тип (масив, другий запис, множина). Порядок опису полів у визначенні “записи” задає їх порядок зберігання у пам’яті. Наприклад, значення полів змінної Point зберігаються як 10 послідовних байтів (2 {x, тип integer} + 2 {y, тип integer} + 6 {z, тип real}).

Змінна-запис може брати участь тільки в операціях присвоювання. Але поле “записи” можна використовувати у всіх операціях, які застосовуються до типу цього поля. Недоліком роботи з записами є те, що неправильні дії ніяк не діагностуються.

Постановка завдання

Відповідно до номера за списком групи обрати по одному з обох частин наведених завдань.

Варіанти завдань

I частина

1. Розробити програму, що дозволяє визначити, чи є в школі два учні, які мають одне ім'я і прізвище. Дані про учня йдуть у такому порядку: ім'я, прізвище, клас (рік навчання – буква). Якщо прізвище або ім'я записується менш ніж 15 буквами, то кінцеві елементи, що залишилися, – пропуски. Дані про різних учнів йдуть у деякій послідовності, про яку заздалегідь нічого невідомо. Передбачається, що в школі навчаються 500 учнів.

2. Розробити програму, що дозволяє визначити, чи є однофамільці, які навчаються в одному класі. Дані про учня йдуть у такому порядку: ім'я, прізвище, клас (рік навчання – буква). Якщо прізвище або ім'я записується менш ніж 15 буквами, то кінцеві елементи, що залишилися, – пропуски. Дані про різних учнів подані у деякій послідовності, про яку заздалегідь нічого невідомо. Передбачається, що в школі навчаються 500 учнів.

3. У магазині "Іграшка" є асортимент дитячих іграшок, який має такі характеристики: назва іграшки (лялька, конструктор, кубики та ін.), ціна (у гривнях) і вікові межі (молодший вік, старший вік). Розробити програму, за допомогою якої видаються такі відомості:

– назви іграшок, вартість яких не перевищує 4 грн і які призначені для дітей 5 років;

– вартість найдорожчого конструктора;

– назви іграшок, розрахованих як на дітей 4 років, так і дітей 10 років.

4. Відомості про автомобіль складаються з назви його марки, номера, прізвища власника. Дано відомості про 20 автомобілів. Розробити програму, що дозволяє знайти:

– прізвища власників і номери автомобілів даної марки;

– кількість автомобілів кожної марки.

5. Узявши спосіб зображення раціонального числа у вигляді запису з двома полями: чисельник, знаменник (integer), розробити програму, що дозволяє:

– визначити, чи є серед 50 раціональних чисел рівні;

– обчислити найбільші із заданих раціональних чисел (числа не обов'язково мають нескоротну форму).

6. Багаж кожного пасажера характеризується числом речей і загальною вагою. Розробити програму, що дозволяє:

– з'ясувати, чи є серед 80 пасажирів такий, чий багаж перевищує багаж кожного з останніх пасажирів і за кількістю речей, і за вагою;

– знайти багаж, в якому середня вага однієї речі відрізняється не більше ніж на 0,3 кг від загальної середньої ваги речей.

7. Надані відомості про товари, що експортуються: найменування, країна, обсяг партії в штуках. Розробити програму, яка дозволяє знайти країни, в які експортується даний товар, і загальний обсяг його експорту.

8. Дано довідник, який містить номери телефонів співробітників установи, їх прізвища, ініціали. Розробити програму, яка дозволяє знайти телефон співробітника за його прізвищем та ініціалами. Передбачити випадок наявності однофамільців з однаковими ініціалами.

9. В універмазі у відділі іграшок продаються кубики. Відомі їх розмір (довжина ребра в сантиметрах), колір (червоний, жовтий, зелений або синій), матеріал (дерево, пластмаса, картон, хутро). Розробити програму, яка дозволяє знайти кількість:

- кубиків кожного з перерахованих кольорів та їх сумарний об'єм;
- дерев'яних кубиків із ребром 3 см і металевих кубиків із ребром більше 5 см.

10. Дано відомості про речовини, що зберігаються на складі: назва, питома вага, провідність (провідник, напівпровідник, ізолятор). Розробити програму, що дозволяє:

- знайти назву та питому вагу всіх напівпровідників;
- вибрати дані про провідники й упорядкувати їх за убутанням питомої ваги.

11. У магазині “Іграшка” є асортимент дитячих іграшок, який має такі характеристики: назва (лялька, конструктор, кубики та ін.), ціна (у копійках) і вікові межі (наприклад, від 2 до 5 років і так далі). Розробити програму, яка дозволяє:

- отримати назви іграшок, призначених як для дітей 10 років, так і для дітей 4 років;
- з'ясувати, чи можна підібрати іграшку, будь-яку, окрім м'яча, призначену для дитини 3 років, і додатково м'яч так, щоб сумарна вартість іграшок не перевищувала 25 грн .

12. У вхідному файлі записана така інформація про кожного з 50 спортсменів: прізвище, стать, вік, вид спорту, результат, причому в прізвищі має бути не більше 12 букв, стать зазначається буквами “ч” або “ж”, вік – ціле число від 16 до 35, вид спорту – не більше 12 позицій, результат – дійсне число. Відомості про спортсменів відокремлені крапкою з комою.

Використовуючи комбінований тип даних, розробити програму, яка вводить цю інформацію і друкує такі дані:

- для кожного виду спорту – кращий результат і прізвище спортсмена (окремо для жінок і чоловіків);
- для кожного виду спорту – прізвище і результат найстаршого спортсмена;
- найменування найбільш популярних видів спорту серед чоловіків і жінок.

13. Анкета школяра включає: прізвище, вік, номер школи, клас і оцінки з 4 предметів. Розробити програму, яка дозволяє:

- визначити номер школи, в якій навчається найуспішніший учень із 10-х класів;
- вказати клас, учні якого мають “3” не більш ніж з одного предмета.

14. Дано 100 оформлених бланків вимог до книги з відомостями: шифр, автор, назва, відомості про читача (номер читацького квитка, прізвище, дата замовлення). Розробити програму, яка дозволяє:

- отримати інформацію про те, скільки читачів замовили одну й ту ж книгу;
- вказати прізвище читача, який замовив найбільшу кількість книг у даний день.

15. На першому курсі в 2-му семестрі передбачено проведення 60 семінарських занять. Відомості про семінар включають: предмет, прізвище викладача, номер групи, день тижня, початок заняття, аудиторію. Розробити програму, що дозволяє:

- з'ясувати, з яких предметів у кожної групи є семінар в середу;
- отримати інформацію про те, в якій групі на даному тижні найбільша кількість семінарських занять.

16. Екзаменаційна відомість містить: назву предмета, номер групи, дату іспиту, 25 рядків із полями: прізвище студента, номер залікової книжки, оцінку. Розробити програму, яка дозволяє:

- отримати інформацію про студентів даної групи, які склали всі іспити на “відмінно”;
- вказати неуспішних студентів, які отримали “2” хоча б на одному іспиті;
- вказати предмет, з якого успішність найвища.

17. Дано список 15 мешканців: прізвище, місто, адреса (вулиця, будинок, квартира). Розробити програму, що дозволяє визначити прізвища двох мешканців із списку, які живуть у різних містах за однаковою адресою.

18. Дано відомості про групу: ім'я, стать, зріст. Розробити програму, яка дозволяє визначити:

- середній зріст жінок у групі;
- ім'я найвищого чоловіка в групі;
- чи є в групі хоча б дві людини однакового зросту.

19. Відомості із записника: прізвище, номер телефону. Вважаючи, що на кожній сторінці зазначені прізвища, які починаються з однієї і тієї ж букви, розробити програму, що дозволяє:

- визначити, чи є в книжці відомості про людину на прізвище Гупало;
- знайти прізвище людини, номер телефону якої 777-55-55.

20. Інформація про підсумки сесії на 1-му курсі: прізвище (12 букв), група (ціле число від 1 до 16), чотири оцінки (“2”, “3”, “4”, “5”). Усього 100 студентів. Розробити програму, яка дозволяє знайти:

- прізвища студентів, які мають хоч би одну “2”;
- якість успішності, тобто відсоток студентів, які склали всі іспити на “4”, “5”;
- назву предмета, який був складений найкраще;
- номери груп за порядком убавання середньої успішності.

II частина

Варіанти 1–5

type число=1..31; місяць=1..12; рік=1..2020;
дата=record ч: число; м: місяць; р: рік; end;
день_тижня=(пн, вт, ср, чт, пт, сб, нд);

Вважаючи, що всі дати подані за григоріанським календарем (“за новим стилем”), описати і викликати:

1) функцію останнє_число(d), яка обчислює кількість днів у тому місяці, якому належить дата d;

2) логічну функцію правильна_дата (d), яка перевіряє правильність дати d (тобто, щоб не було 31 червня і т. ін.);

3) функцію ЧД1(d), яка рахує, скільки днів минуло від 1-го січня 1-го року нашої ери до дати d;

4) функцію ДТ(d) для визначення дня тижня, на який припадає дата d (врахувати, що 1-го січня 1-го року нашої ери був понеділок);

5) функцію ЧД2(d), яка рахує, скільки днів залишилося від дати d до 31-го грудня 2020-го року.

Варіанти 6–10

У вхідному файлі записана інформація про 50 автомобілів:

<марка>, <номер>, <кольор>, <рік випуску>, <власник>, <швидкість>, причому в найменуванні марки, кольорі, імені власника міститься не більше 12 букв, номер – ціле число від 0000 до 9999, рік випуску – ціле від 1980 до 2008, швидкість – дійсне від 0,0 до 200,0. Відомості про автомобілі відокремлені знаком “;”.

Використовуючи комбінований тип даних, розробити програму, яка вводить цю інформацію і друкує такі дані:

6) найменування найрозповсюдженішої марки автомашини та найменування найбільш популярного кольору автомобіля цієї марки;

7) прізвища власників двох і більше автомобілів одного кольору;

8) номери найбільш старих автомобілів кожної марки;

9) середню швидкість автомобілів кожного кольору;

10) рік випуску найшвидшого автомобіля кожної марки.

Варіанти 11–15

У вхідному файлі записана інформація про 50 спортсменів:

<прізвище>, <стать>, <вік>, <вид спорту>, <результат>, при цьому у прізвищі, виді спорту – не більше 12 букв, стать зазначається буквами Ч і Ж, вік – ціле від 16 до 35, результат – дійсне. Відомості про спортсменів відокремлені знаком “;”.

Використовуючи комбінований тип даних, розробити програму, яка вводить цю інформацію і друкує такі дані:

11) для кожного виду спорту – кращий результат і прізвище спортсмена (окремо для жінок і чоловіків);

12) для кожного виду спорту – прізвище і результат найстаршого спортсмена;

13) найменування найбільш популярних видів спорту серед чоловіків і жінок;

14) середній вік спортсменів, які мають найкращі результати з кожного виду спорту;

15) прізвище наймолодшого спортсмена, який має найкращий результат з якого-небудь виду спорту.

Варіанти 16–20

У вхідному файлі записана інформація про підсумки зимової сесії на 1-му курсі. Відомості про 60 студентів-першокурсників задані у такому вигляді:

<прізвище>, <номер групи>, <оцінка1>, <оцінка2>, <оцінка3>, при цьому у прізвищі – не більше 12 букв, номер групи – ціле число від 101 до 105, кожна оцінка – це “2”, “3”, “4”, “5”, причому перша оцінка – за екзамен з програмування, друга – з алгебри, третя – з математичного аналізу. Відомості про студентів відокремлені знаком “;”.

Використовуючи комбінований тип, розробити програму, яка вводить цю інформацію і друкує такі дані:

16) прізвища студентів, які мають заборгованість хоч з одного предмета;

17) “якість” навчання, тобто процент студентів, які поклали всі екзамени на “4” і “5”;

18) найменування предмета, який був складений найкраще;

19) номери груп за порядком зменшення середньої успішності їх студентів;

20) прізвища студентів-відмінників.

Варіанти 21–25

У вхідному файлі записана інформація про 50 студентів університету:

<прізвище>, <ім’я>, <по батькові>, <стать>, <вік>, <курс>, при цьому у прізвищі, імені, по батькові – не більше 12 букв, стать указується буквами Ч і Ж, вік – ціле число від 16 до 35, курс – ціле число від 1 до 5. Відомості про студентів відокремлюються “;”.

Використовуючи комбінований тип, розробити програму, яка вводить цю інформацію і друкує такі дані:

21) номер курсу із найбільшим процентом чоловіків;

22) найбільш поширені чоловіче та жіноче імена;

23) прізвища (в алфавітному порядку) та ініціали всіх студентів, вік і по батькові яких є одночасно найпоширенішими;

24) середній вік студентів і студенток на кожному курсі;

25) відомості про всіх однофамільців.

Зміст звіту

1. Постановка завдання.

2. Аналіз завдання (опис структурної блок-схеми алгоритму).

3. Протокол роботи програми (набір вхідних тестових даних і результати розрахунків).

4. Лістинг програми.

Приклад. Розробити програму, яка дозволяє визначити, чи є на факультеті в яких-небудь паралельних групах однофамільці. Дані про студентів розміщені у такому порядку: прізвище, ім'я, рік навчання, шифр групи. Якщо прізвище або ім'я записується менше ніж 15 буквами, то кінцеві елементи, що залишаються, – це прогалини.

Лістинг програми (мова Паскаль)

```

Program ppl;
Label 1;
Const n=15; m=60;
Type Grappa=record jeer: integer;
                kod: char;
            end;
Fio=array [1..n] of char;
Stud= record f, name: Fio;
        gr: Grappa;
    end;
Facultet=array[1..m] of Stud;
Var s: Facultet; i,j,k: integer; g: text;
begin
    assign(g, 'dan.txt');
    {$I-}
    Reset(g);
    If (IoResult<>0) then
        begin
            writeln('помилка відкриття файлу dan.txt');
            exit;
        end;
    {$I+}
    for i:=1 to m do
        begin
            {введення прізвища і-го студента}
            for j:=1 to n do
                read (g, S[i].f[j]);
            {введення імені і-го студента}
            for j:=1 to n do
                read (g, S[i].name[j]);
            {введення року навчання та шифру}
            read (g, S[i].gr.jeer, S[i].gr.kod)
        end;
        for i:=1 to m do
            for j:=i+1 to m do
                begin
                    if(S[i].gr.jeer= S[j].gr.jeer)and(S[i].gr.kod<>S[j].gr.kod)
                    then
                        begin
                            for k:=1 to n do
                                if(S[i].f[k]<>S[j].f[k]) then goto 2;

```

```

write('на факультеті є однофамільці');
exit;
readln;
end;
1:   end;
write('на факультеті немає однофамільців');
readln;
end.

```

Зауваження. В процесі виконання програми спочатку заповнюється масив записів S даними із файлу про m студентів факультету. Під час пошуку інформації про однофамільців значення $S[i]$ порівнюється з $S[i+1], \dots, S[m]$ до першого знайденого однофамільця у паралельній групі.

Лабораторна робота 6

ЗАСТОСУВАННЯ СТАНДАРТНИХ ПІДПРОГРАМ ДЛЯ РОЗВ'ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРИЧНИХ РІВНЯНЬ

Мета:

- розвинути навички застосування стандартних підпрограм;
- закріпити навички роботи з масивами.

Постановка завдання

Написати процедуру перетворення заданої у табл. 6.1 матриці C 5-го порядку на матрицю A 4-го порядку та вектор вільних параметрів B системи лінійних алгебричних рівнянь (СЛАР) і основну програму, яка викликає згадану процедуру і розв'язує СЛАР $Ax = B$, де $x = (x_1, x_2, x_3, x_4, \dots)$ – вектор невідомих системи – за допомогою стандартних підпрограм SIMQ і GELG. Зробити перевірку одержаного розв'язку.

Варіанти завдань обираються із табл. 6.2 згідно з номером шифру (кожна цифра шифру – це номер у першому стовпчику, відповідно до якого обираються послідовно значення $N1, N2, N3, N4, N5$ як цифри на перетині відповідних стовпчика і рядка).

Матриця С

Таблиця 6.1

N	1	2	3	4	5
1	5.3	1.4	-0.5	6.1	-7.3
2	1.5	-2.5	6.4	5.3	-1.6
3	-8.4	5.6	-1.3	7.7	1.5
4	4.7	-1.8	7.2	-3.6	3.4
5	-3.1	2.9	0.9	-4.2	8.7

Варіанти завдань

Таблиця 6.2

№	$N1$	$N2$	$N3$	$N4$	$N5$
1	5	1	3	1	0
2	4	3	2	2	9
3	3	5	1	3	8
4	2	4	5	4	7
5	1	2	4	5	6
6	5	1	3	6	5
7	4	2	1	7	4
8	3	5	2	8	3
9	2	4	5	9	2
0	1	3	4	0	1

Примітка. $N1, N2$ – номери рядка і стовпчика відповідно, які слід вилучити із матриці C для одержання проміжної матриці D , що використовується для формування вихідної матриці A ; $N3$ – номер стовпчика матриці C , за винятком елемента у рядку $N1$, який береться за проміжний вектор R , що використовується для формування вихідного вектора B ; $N4$ – варіант перетворення матриці D на матрицю A ; $N5$ – варіант перетворення вектора R на вектор B .

I. Варіанти перетворення матриці D на матрицю A :

- 1) елементи рядків матриці D розташовуються в порядку зростання;
- 2) елементи рядків матриці D розташовуються в порядку спадання;
- 3) елементи стовпчиків матриці D розташовуються у порядку зростання;
- 4) елементи стовпчиків матриці D розташовуються у порядку спадання;
- 5) елементи всіх рядків матриці A одержують як півсуму двох сусідніх елементів того ж рядка матриці D (у першого й останнього елементів рядка змінюють знак);
- 6) елементи всіх стовпчиків матриці A одержують як півсуму двох сусідніх елементів того ж стовпчика матриці D (у першого і останнього елементів стовпчика змінюють знак);
- 7) елементи матриці A одержують у результаті ділення відповідних елементів матриці D на суму елементів рядка, в якому вони знаходяться;
- 8) елементи матриці A одержують внаслідок ділення відповідних елементів матриці D на суму елементів стовпчика, в якому вони знаходяться;
- 9) елементи матриці A одержують в процесі ділення відповідних елементів матриці D на середнє геометричне модулів елементів рядка, в якому вони знаходяться;
- 0) елементи матриці A одержують у результаті ділення відповідних елементів матриці D на середнє геометричне модулів елементів стовпчика, в якому вони знаходяться.

II. Варіанти перетворення вектора R на вектор B

Складники b_i вектора B дорівнюють:

- 1) сумі r_i та мінімального елемента i -го рядка матриці D ;
- 2) сумі r_i та мінімального елемента i -го стовпчика матриці D ;
- 3) подвійному максимальному елементу i -го рядка матриці D , поділеному на r_i ;
- 4) подвійному максимальному елементу i -го стовпчика матриці D , поділеному на r_i ;
- 5) різниці r_i^2 і середнього арифметичного елементів i -го рядка матриці D ;
- 6) різниці r_i^2 і середнього арифметичного елементів i -го стовпчика матриці D ;
- 7) додатку елементів i -го рядка матриці D , поділеному на r_i^2 ;
- 8) додатку елементів i -го стовпчика матриці D , поділеному на r_i^2 ;
- 9) сумі елементів i -го рядка матриці D , поділеній на $2r_i$;
- 0) сумі елементів i -го стовпчика матриці D , поділеній на $4r_i$.

Зміст звіту

1. Постановка завдання.
2. Аналіз завдання (опис структурної блок-схеми алгоритму).
3. Опис параметрів використаних підпрограм.
4. Протокол роботи програми (вхідні дані, матриця A , вектор B , розв'язок СЛАР у разі використання підпрограм SIMQ і GELG, вектор відхилю $R=AX-B$).
4. Лістинг програми.

Приклад. Увести з файлу матрицю A та вектор вільних членів B , розв'язати систему $AX=B$ за допомогою стандартної підпрограми SIMQ.

Лістинг програми (мова Фортран)

```
PROGRAM SLAR;  
PARAMETER (N=5);  
DIMENSION A(N, N), B(N)  
OPEN(3, FILE='DATA.TXT', STATUS=OLD)  
READ(3, 10) ((A(I, J), I=1,N), J=1,N)  
READ(3, 20) (B(I), I=1,N)  
CLOSE(3, STATUS=KEEP)
```

```

CALL SIMQ(A, B, N, KS)
IF (KS.NE.0)
WRITE(*,*) 'Система не має розв'язків'
GOTO 40
ENDIF
WRITE(*, 30) (B(I), I=1,N)
10  FORMAT (<N>(<N>F10.4))
20  FORMAT (<N>F10.4)
30  FORMAT (<N>(F10.4, 2X))
40  CONTINUE
END

```

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

- Бартеньев, О. В. Современный Фортран [Текст] / О. В. Бартеньев. – М.: Диалог-МИФИ, 2005. – 400 с.
- Задания и методические указания к выполнению курсовых работ и решению задач по вычислительной технике и программированию [Текст] / А. А. Бобылёв, Д. Г. Галочкин, А. П. Дзюба, В. Г. Мусияка. – Д.: ДГУ, 1991. – 44 с.
- Калоін, В. А. Основи інформатики. Структурне програмування на Паскалі [Текст]: практикум / В. А. Калоін, О. І. Черняк, О. М. Харитонов. – К.: Центр учбов. літ., 2007. – 248 с.
- Немнюгин, С. А. Современный Фортран [Текст]: самоучитель / С. А. Немнюгин, О. Л. Стесик. – СПб: БХВ–Петербург, 2005. – 496 с.
- Поляков, Д. Б. Программирование в среде Турбо Паскаль (версия 5.5) [Текст]: справоч.-метод. пособие / Д. Б. Поляков, И. Ю. Круглов. – М.: Изд-во МАИ, 1992. – 576 с.
- Шикова, О. М. Основи програмування мовою Паскаль у прикладах і завданнях [Текст]: навч. посіб. / О. М. Шикова. – К.: МАУП, 2004. – 112 с.
- Шишук, В. В. Основи програмування на алгоритмічній мові PASCAL [Текст]: навч. посіб. для вузів / В. В. Шишук. – К.: Кондор, 2006. – 224 с.

Зразок оформлення звіту про виконання лабораторних робіт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпропетровський національний університет
ім. Олеся Гончара
Механіко-математичний факультет
Кафедра обчислювальної механіки і міцності конструкцій

З В І Т

про виконання лабораторної роботи №
<НАЗВА>

ШИФР <номер>

Виконавець: студент групи МХ-08-1
Прийняв:

<П.І.Б.>
<П.І.Б. викладача>

Завдання отримано <дата>
Звіт зданий <дата>

Дніпропетровськ
2009

ЗМІСТ

Анотація	2
Основна частина.....	2
1. Постановка завдання.....	2
2. Аналіз завдання.....	2
3. Структурна блок-схема алгоритму.....	3
4. Інструкція щодо експлуатації програми.....	4
5. Результати розрахунків і аналіз.....	5
Список використаної літератури.....	7
Додаток. Лістинг програми.....	7

АНОТАЦІЯ

1. Постановка задачі

Обчислити суму нескінченного ряду $S = \sum_{n=0}^{\infty} \frac{(-1)^n \cos^{n+1} x}{(2n+1)!}$ зі заданою

точністю ε і видати значення S і n .

2. Тип комп'ютера. Операційна система

Програма розрахована для ПК типу Pentium IV. Реалізована в ОС Windows XP.

3. Мова програмування – Turbo Pascal 7.0.

4. Допоміжні програми – немає.

5. Стан програми – налагоджена.

ОСНОВНА ЧАСТИНА

1. Постановка завдання

Розробити структурну схему алгоритму та програму обчислення суми нескінченного ряду

$$S = \cos x - \frac{\cos^2 x}{3!} + \frac{\cos^3 x}{5!} - \frac{\cos^4 x}{7!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n \cos^{n+1} x}{(2n+1)!}$$

зі заданою точністю ε ($\varepsilon=10^{-3}$; 10^{-4} ; 10^{-5} ; 10^{-6}), вважаючи, що задана точність досягнута, якщо черговий складник за абсолютним значенням стане меншим, ніж ε , і видати вхідні дані – x і ε , та вихідні дані – S і N на екран.

2. Аналіз завдання

Для зменшення витрат машинного часу на визначення суми нескінченного ряду S черговий складник будемо обчислювати за рекурентними залежностями. Введемо позначки:

$$A_n = (-1)^n, B_n = \cos^{n+1} x, C_n = (2n+1)!$$

Рекурентні формули мають вигляд

$$A_n = -A_{n-1}, A_0 = 1, n = 1, 2, 3 \dots$$

$$B_n = \cos x B_{n-1}, B_0 = \cos x, n = 1, 2, 3 \dots$$

$$C_n = (2n+1)2nC_{n-1}, C_0 = 1, n = 1, 2, 3 \dots$$

Загальний член ряду дорівнює $u_n = (A_n B_n) / C_n$.

Тестування проведемо при $x = 2\pi$. У результаті обчислення шести перших складників бачимо спадання їх значень:

$$u_0 = 1, u_1 = -0.1666667, u_2 = 0.0083333, u_3 = -0.0001984, \\ u_4 = 0.0000028, u_5 = -0.000000025455,$$

при цьому одержимо $S = 0.0841471, N = 6$.

3. Структурна блок-схема алгоритму

4. Інструкція щодо експлуатації програми

Програма *sumt.pas* знаходиться на дискеті у кореневому каталозі.

Вхідні дані:

x – аргумент; EPS – задана точність ε .

Вихідні дані:

S – значення обчисленої суми ряду; N – кількість складників ряду при заданій точності.

На екран вхідні та вихідні дані виводяться у вигляді

$$x = \quad, EPS = \\ S = \quad, N =$$

5. Результати розрахунків і аналіз

Значення суми ряду і кількість складників у разі $x = 2\pi$, знайдених для різних значень EPS , наведені у таблиці.

Результати розрахунків

EPS	S	N
0.001	0.8416667	3
0.0001	0.8414682	4
0.00001	0.8414710	5
0.000001	0.8414710	5

Слід відзначити, що у випадку зростання заданої точності обчислень EPS зростає кількість складників і стабілізується значення суми S .

Завдання для розрахункової роботи

Мета:

- навчитись використовувати динамічні структури даних;
- навчитись використовувати процедурні типи;
- закріпити навички роботи з типом “запис”.

Постановка завдання

Визначити структуру даних відповідно до завдання. Розмістити структури в динамічній пам'яті, об'єднавши їх в групу. Зберегти групу у файлі.

Теоретичні відомості

Група – це об'єкт (структура), до якого включені інші об'єкти (структури). Об'єкти, що належать до групи, називаються елементами групи. Елементи групи, у свою чергу, можуть бути групою.

Приклади груп:

- 1) вікно в інтерактивній програмі, яке містить такі елементи, як поля введення і редагування даних, кнопки, списки вибору, діалогові вікна і т. ін.;
- 2) агрегат, що складається з дрібніших вузлів;
- 3) город, на якому передбачені рослини, системи поливу та план вирощування;
- 4) деяка організаційна структура (наприклад, ФАКУЛЬТЕТ, КАФЕДРА, СТУДЕНТСЬКА ГРУПА).

Реалізувати групу можна таким чином.

Створюється зв'язаний список записів типу *TItem*:

```

Type
  PItem = ^TItem;
  TItem = record
    next: PItem;
    item: PObject;
  end;

```

Поле *item* вказує на об'єкт, включений у групу. Група містить поле *last* типу *PItem*, яке вказує на початок зв'язаного списку об'єктів, включених у групу.

Для роботи з групою мають бути визначені такі процедури та функції:

- 1) *Procedure GroupInsert (var group: TGroup; p: PObject)* включає в групу *group* структуру, на яку вказує *p*;
- 2) *Procedure GroupShow (group: TGroup)* дозволяє проглянути групу;
- 3) *Function GroupEmpty (group: TGroup):boolean* показує, чи є хоча б один елемент у групі;
- 4) *Procedure GroupDelete (var group: TGroup; p: PObject)* видаляє елемент із групи.

Ітератори дозволяють виконувати деякі дії для кожного елемента певного набору даних.

Такий цикл міг би бути виконаний для всього набору, наприклад, щоб надрукувати всі елементи останнього. Або міг би шукати деякий елемент, який задовольняє певну умову, і в цьому випадку такий цикл може закінчитися, як тільки буде знайдений необхідний елемент.

Ми будемо розглядати ітератори як спеціальні процедури групи, що дозволяють виконувати деякі дії для всіх об'єктів, включених в групу. Нам потрібно мати такий ітератор, який дозволяв би виконувати над усіма елементами групи дії, задані довільною процедурою або функцією користувача. Такий ітератор можна реалізувати, якщо цю процедуру (функцію) передавати йому через параметр процедурного типу:

```
Type TProc = procedure (p: PObject);
```

Процедура-ітератор може оголошуватися таким чином:

```
Procedure GroupIter (group: TGroup; proc: TProc);
```

Процедура або функція, що передається ітератору, має бути відкомпільована з атрибутом `far`.

```
Procedure MyProc (p: PObject); far;
```

Ітератор можна зробити зручнішим, якщо передавати йому вказівник на процедуру:

```
Procedure GroupIter (group: TGroup; proc: pointer);
```

Процедурні типи. У Borland Pascal допускається інтерпретація процедур і функцій як об'єктів, які можна зберігати в змінних і передавати як параметри. Такі дії можна виконувати за допомогою процедурних типів.

У описі процедурного типу задаються параметри, а для функції – ще і результат функції.

Формат опису процедурного типу:

```
procedure (<список формальних параметрів>)
```

або

```
function (<список формальних параметрів>): <тип результату>.
```

Синтаксис запису процедурного типу збігається із записом заголовка процедури або функції, лише пропускається ідентифікатор після ключового слова *procedure* або *function*. Наведемо деякі приклади описів процедурного типу:

```
type
```

```
Proc = procedure;
```

```
SwapProc = procedure (var X, Y: Integer);
```

```
StrProc = procedure (S: String);  
MathFunc = function (X: Real): Real;  
DeviceFunc = function (var F: text): Integer;  
MaxFunc = function (A, B: Real; F: MathFunc): Real;
```

У змінній процедурного типу можна зберігати процедурні значення, які можуть бути такими:

- значення nil;
- вказівник на змінну процедурного типу;
- ідентифікатор процедури або функції.

У контексті процедурних значень опис процедури або функції можна розглядати як спеціальний вигляд описів констант, коли значенням константи є процедура або функція. Значення nil вказує на те, що в процедурній змінній не зберігається ніяке значення.

Параметри процедурного типу. Оскільки процедурні типи допускається використовувати в будь-якому контексті, то можна описувати процедури або функції, які сприймають інші процедури і функції як параметри.

Параметри процедурного типу особливо корисні у тому випадку, коли потрібно виконати якісь загальні дії над множиною деяких об'єктів. У наступному прикладі показано використання параметрів процедурного типу для виведення трьох таблиць різних арифметичних функцій:

```
program Tables;  
type  
  Func = function (X, Y: integer): integer;  
  
function Add (X, Y: integer): integer; far;  
begin  
  Add := X + Y;  
end;  
  
function Multiply (X, Y: integer): integer; far;  
begin  
  Multiply := X*Y;  
end;  
  
function Funny (X, Y: integer): integer; far;  
begin  
  Funny := (X+Y) * (X-Y);  
end;  
  
procedure PrintTable (W, H: integer; Operation: Func);  
var
```



```

    X, Y : integer;
begin
  for Y := 1 to H do
    begin
      for X := 1 to W do Write (Operation (X, Y): 5);
        Writeln;
      end;
      Writeln;
    end;

begin
  PrintTable (10, 10, Add);
  PrintTable (10, 10, Multiply);
  PrintTable (10, 10, Funny);
end.

```

У даному випадку процедура *PrintTable* є загальною дією, що виконується над функціями *Add*, *Multiply* і *Funny*.

Порядок виконання роботи

1. Визначити структуру (відповідно до варіанта завдання), а також процедури і функції для роботи з нею. Помістити визначення в модуль unit1.
2. Визначити типи, процедури та функції для організації роботи з групою. Помістити визначення в модуль unit2.
3. Визначити типи, процедури і функції для організації роботи з файлом (збереження групи у файлі та завантаження її з файлу). Помістити визначення в модуль unit3.
4. Визначити ітератор (відповідно до варіанта завдання) і помістити визначення в модуль unit2.
5. Розробити програму, в якій створюються структури, поміщаються в групу, група друкується, а потім зберігається у файлі. Після збереження група знову створюється шляхом зчитування з файлу і друкується.
6. Налогодити розроблену програму.
7. Написати звіт.

Зміст звіту

1. Титульний аркуш.
2. Постановка завдання згідно з варіантом (тип структури та ітератор).
3. Інтерфейсні частини всіх модулів.
4. Пояснити виконання таких процедур:
 - перегляд структури;
 - перегляд групи;
 - включення структури в групу;
 - видалення структур із групи;

- завантаження групи з файлу.
5. Пояснити виконання реалізованого ітератора.
 6. Додаток (на диску):
 - а) лістинги всіх модулів та основна програма: pas-файли;
 - б) exe-файл програми.

Варіанти завдань

Завдання обираються із таблиці. Номер рядка в таблиці відповідає номеру в групі за списком, номер структури збігається з першою цифрою шифру.

Варіанти завдань

<i>№</i>	<i>Структура</i>	<i>Ітератор</i>
1	СТУДЕНТ	Імена всіх хлопців (дівчат)
2	СТУДЕНТ	Імена студентів зазначеного курсу
3	СТУДЕНТ	Кількість хлопців (дівчат)
4	СТУДЕНТ	Кількість студентів на зазначеному курсі
5	СЛУЖБОВЕЦЬ	Імена службовців зі стажем не меншим за заданий
6	СЛУЖБОВЕЦЬ	Імена службовців заданої професії
7	СЛУЖБОВЕЦЬ	Кількість службовців зі стажем не меншим за заданий
8	СЛУЖБОВЕЦЬ	Кількість службовців заданої професії
9	КАДРИ	Імена робітників у заданому цеху
10	КАДРИ	Імена робітників заданого розряду
11	ЦЕХ	Кількість продукції заданого найменування
12	ЦЕХ	Найменування продукції, кількість якої не менша за задану величину
13	БІБЛІОТЕКА	Найменування книг, вартість яких вища за задану величину
14	БІБЛІОТЕКА	Кількість книг вказаного автора
15	ІСПИТ	Імена студентів, які склали іспит на відмінно
16	ІСПИТ	Імена студентів, які склали іспит у зазначений день
17	ІСПИТ	Кількість студентів, які не склали іспит
18	АДРЕСА	Імена тих, хто мешкає на заданій вулиці

№	Структура	Ітератор
19	АДРЕСА	Імена тих, хто мешкає на парному боці заданої вулиці
20	АДРЕСА	Кількість мешканців заданої вулиці
21	ТОВАР	Кількість товару заданого найменування
22	ТОВАР	Найменування товару, кількість якого перевищує задану величину
23	ТОВАР	Найменування товару вартістю не вище заданої величини
24	КВИТАНЦІЯ	Номери квитанцій вказаної дати
25	КВИТАНЦІЯ	Загальна сума всіх квитанцій вказаної дати

Описи структур

1. СТУДЕНТ

ім'я : string
курс : integer
стать : char

2. СЛУЖБОВЕЦЬ

ім'я : string
професія : string
робочий стаж : integer

3. КАДРИ

ім'я : string
номер цеху : integer
розряд : integer

4. ЦЕХ

назва : string
шифр : string
кількість : integer

5. БІБЛІОТЕКА

назва : string
автор : string
вартість : real

6. ІСПИТ

ім'я студента : string
дата : integer
оцінка : integer

7. АДРЕСА

ім'я : string
вулиця : string
будинок : integer

8. ТОВАР

назва : string
кількість : integer
вартість : real

9. КВИТАНЦІЯ

номер : integer
дата : integer
сума : real

Методичні вказівки щодо виконання розрахункової роботи

1. Модуль unit1 містить визначення структури, а також процедури та функції для роботи з нею. У Pascal структура – це запис (record). Наприклад, структура ПЕРСОНА визначається так:

```
TPerson = record
    name: string;    {ім'я}
    age: integer;    {вік}
end;
```

Слід визначити також тип вказівника на структуру:

```
PPerson = ^TPerson;
```

2. Для роботи зі структурою слід описати такі процедури та функції:
– функцію, яка розміщує структуру в пам'яті і повертає вказівник на неї, наприклад:

```
function PersonCreate (name1: string; age1: integer): PPerson;
```

– процедуру, що видаляє структуру з пам'яті, наприклад (*p* – вказівник на структуру, яка видаляється):

```
procedure PersonDestroy (Var p: PPerson);
```

– процедури для ініціалізації даних структури, наприклад:

```
procedure PersonInit (p: PPerson; name1: string; age1: integer);
```

```
procedure SetName (p: PPerson; name1: string);
```

```
procedure SetAge (p: PPerson; age1: integer);
```

– функції, що повертають значення структури, наприклад:

```
function GetName (p: PPerson): string;
```

```
function GetAge (p: PPerson): integer;
```

– процедуру для перегляду структури (виведення її значень), наприклад: *procedure PersonShow(p: PPerson);*

3. Другий модуль (unit2) містить визначення типів, процедур і функцій для організації роботи з групою.

4. Слід визначити процедуру, яка включає структуру в групу:

```
procedure GroupInsert (var group: TGroup ;p: PPerson);
```

Потрібно визначити також процедури для створення (ініціалізації) групи, її перегляду та видалення: *GroupCreate, GroupShow, GroupDestroy.*

5. Третій модуль (unit3) містить процедури, необхідні для збереження групи у файлі і завантаження її з файлу. Тут необхідно визначити такі процедури:

AssignFile – зв'язати файловою змінну із зовнішнім файлом;
OpenFileRead – відкрити файл для читання;
OpenFileWrite – відкрити файл для запису;
ReadPersons – читати записи з файлу і помістити їх в групу;
WritePersons – зберегти групу у файлі;
CloseFile – закрити файл.

6. Приклад побудови основної програми:

```
uses Unit1, Unit2, Unit3;  
Type TProc = procedure(person: TPerson);  
Var p: PPerson;  
    a, b: TGroup;  
    f: TFile;  
Procedure Show (person: PPerson; age1: integer); far;  
{Показує імена студентів, чий вік більше age1}  
begin  
    if person^.age>age1 then PersonShow (person);  
end;
```

Begin

```
AssignFile (f1, 'person1.dat');  
AssignFile (f2, 'person2.dat');  
GroupCreate (a, 'ГРУПА А');  
GroupCreate (b, 'ГРУПА Б');  
p:=PersonCreate ('Іванов',25);  
GroupInsert (a, p);  
p:=PersonCreate ('Петров', 35);  
GroupInsert (a, p);  
p:=PersonCreate ('Сидоров',45);  
GroupInsert (a, p);  
{аналогічно для групи b}  
GroupShow (a);  
OpenFileWrite (f1);  
WritePersons (f1, a);  
CloseFile (f1);  
GroupDestroy (a);  
GroupShow (a); {показуємо групу, щоб переконатися, що вона  
                порожня}  
{ аналогічно для групи b }  
OpenFileRead (f1);  
ReadPersons (f1, a);
```

```

GroupShow (a);
{ аналогічно для групи b }
GroupIter (a, Show, 35);
GroupDestroy (a);
GroupDestroy (b);
End.

```

Додаток Г

Опис стандартних підпрограм SIMQ і GELG

Підпрограма **SIMQ** розв'язує систему N лінійних алгебричних рівнянь $Ax = B$ методом виключення з обранням головного елемента зі стовпчика. Змінну x_N одержують у разі прямого ходу розв'язання за N кроків. Решту змінних одержують за допомогою послідовних підстановок у випадку оберненого ходу розв'язання.

Виклик підпрограми має вигляд

CALL SIMQ (A, B, N, KS),

де **A** – масив, який є матрицею коефіцієнтів системи рівнянь N -го порядку, розташованою за стовпчиками;

B – масив довжини N , який містить праві частини системи рівнянь. Під час виходу із підпрограми у ньому знаходиться розв'язок системи, тобто вектор x ;

N – кількість рівнянь системи та змінних ($N > 1$);

KS – індикатор помилки, який може набувати таких значень: $KS=0$ означає, що помилок немає; $KS=1$ – матриця A системи рівнянь особлива.

Лістинг підпрограми SIMQ

```

SUBROUTINE SIMQ(A,B,N,KS)
REAL A(1), B(1), TOL, BIGA, SAVE
TOL=0.0
KS=0
JJ=-N
DO 65 J=1,N
  JY=J+1
  JJ=JJ+N+1
  BIGA=0.0
  IT=JJ-J
  DO 30 I=J,N
    IJ=IT+I
    IF(ABS(BIGA)-ABS(A(IJ)))20,30,30
    A(I2)=SAVE
50  A(I1)=A(I1)/BIGA
    SAVE=B(IMAX)
    B(IMAX)=B(J)
    B(J)=SAVE/BIGA
    IF(J-N)55,70,55
55  I=N*(J-1)
    DO 65 IX=JY,N
      I1=I+IX
      IT=J-IX
      DO 60 JX=JY,N
        IJ=N*(JX-1)+IX
        I2=IJ+IT

```

```

20 BIGA=A(IJ)
   IMAX=I
30 CONTINUE
   IF(ABS(BIGA)-TOL)35,35,40
35 KS=1
   RETURN
40 I1=J+N*(J-2)
   IT=IMAX-J
   DO 50 I=J,N
   I1=I1+N
   I2=I1+IT
   SAVE=A(I1)
   A(I1)=A(I2)
60 A(IJ)=A(IJ)-(A(I1)*A(I2))
65 B(IJ)=B(IJ)-(B(I1)*A(I2))
70 JY=N-1
   IT=N*N
   DO 80 J=1,JY
   I=IT-J
   I1=N-J
   I2=N
   DO 80 JJ=1,J
   B(I1)=B(I1)-A(I)*B(I2)
   I=I-N
80 I2=I2-1
   END

```

Підпрограма **GELG** розв'язує систему N лінійних алгебричних рівнянь $Ax = B$ ітераційним методом. За один виклик підпрограми можливо розв'язати декілька систем з однією і тією ж матрицею коефіцієнтів, але з різними правими частинами.

Виклик підпрограми має вигляд

CALL GELG (R, A , M, N, EPS , IER),

де **R** – матриця N правих частин у вигляді масиву $M \times N$. Під час виходу із підпрограми в цьому масиві містяться розв'язки системи, тобто вектори x ;

A – масив, який містить матрицю коефіцієнтів системи рівнянь M -го порядку, розташовану за стовпчиками;

M – кількість рівнянь системи та змінних ($M > 1$);

N – кількість правих частин системи ($N \geq 1$);

EPS – вхідна константа, яка використовується для оцінки відносної похибки під час перевірки втрати точності;

IER – індикатор помилки, який може набувати таких значень: $IER=0$ означає, що помилок немає; $IER=1$ – матриця A системи рівнянь особлива або $M < 1$; $IER=K$ – на k -му кроці ітерацій можлива втрата точності, оскільки головний елемент матриці виявився меншим за добуток точності EPS і найбільшого за модулем елемента матриці A .

Лістинг підпрограми GELG

```

SUBROUTINE GELG(R,A,M,N,EPS,IER)  13 A(L)=TB
DIMENSION A(1),R(1)              A(LST)=J
IF(M)23,23,1                      PIV=0.
1  IER=0                           LST=LST+1
   PIV=0.                            J=0
   MM=M*M                            DO 16 II=LST,LEND

```

```

NM=N*M
DO 3 L=1,MM
TB=ABS(A(L))
IF(TB-PIV)3,3,2
2 PIV=TB
I=L
3 CONTINUE
TOL=EPS*PIV
LST=1
DO 17 K=1,M
IF(PIV)23,23,4
4 IF(IER)7,5,7
5 IF(PIV-TOL)6,6,7
6 IER=K-1
7 PIVI=1./A(I)
J=(I-1)/M
I=I-J*M-K
J=J+1-K
DO 8 L=K,NM,M
LL=L+I
TB=PIVI*R(LL)
R(LL)=R(L)
8 R(L)=TB
IF(K-M)9,18,18
9 LEND=LST+M-K
IF(J)12,12,10
10 II=J*M
DO 11 L=LST,LEND
TB=A(L)
LL=L+II
A(L)=A(LL)
11 A(LL)=TB
12 DO 13 L=LST,MM,M
LL=L+I
TB=PIVI*A(LL)
A(LL)=A(L)
PIVI=-A(II)
IST=II+M
J=J+1
DO 15 L=IST,MM,M
LL=L-J
A(L)=A(L)+PIVI*A(LL)
TB=ABS(A(L))
IF(TB-PIV)15,15,14
14 PIV=TB
I=L
15 CONTINUE
DO 16 L=K,NM,M
LL=L+J
16 R(LL)=R(LL)+PIVI*R(L)
17 LST=LST+M
18 IF(M-1)23,22,19
19 IST=MM+M
LST=M+1
DO 21 I=2,M
II=LST-I
IST=IST-LST
L=IST-M
L=A(L)+.5
DO 21 J=II,NM,M
TB=R(J)
LL=J
DO 20 K=IST,MM,M
LL=LL+1
20 TB=TB-A(K)*R(LL)
K=J+L
R(J)=R(K)
21 R(K)=TB
22 RETURN
23 IER=-1
RETURN
END

```


ЗМІСТ

Вступ	3
Лабораторна робота 1. ОБЧИСЛЕННЯ СУМИ НЕСКІНЧЕННОГО РЯДУ ІЗ ЗАДАНОЮ ТОЧНІСТЮ	4
Лабораторна робота 2. ДІЇ НАД МАТРИЦЯМИ	8
Лабораторна робота 3. ЗАСТОСУВАННЯ ПРОЦЕДУР І ФУНКЦІЙ. РЕКУРСІЯ	12
Лабораторна робота 4. ДІЇ НАД РЯДКАМИ	18
Лабораторна робота 5. ФАЙЛОВА СИСТЕМА. РОБОТА З ЗАПИСАМИ	23
Лабораторна робота 6. ЗАСТОСУВАННЯ СТАНДАРТНИХ ПІДПРОГРАМ ДЛЯ РОЗВ’ЯЗУВАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРИЧНИХ РІВНЯНЬ	30
Список рекомендованої літератури	33
Додаток А. Зразок оформлення звіту про виконання лабораторних робіт ...	34
Додаток Б. Завдання для розрахункової роботи	37
Додаток В. Методичні вказівки щодо виконання розрахункової роботи	43
Додаток Г. Опис стандартних підпрограм SIMQ і GELG	45