

Лабораторна робота №7_8

Програмування алгоритмів обробки масивів даних та їх реалізація на мові ПР та C/C++

Мета роботи: Вивчити прийоми введення та виведення даних в одномірних та двомірних масивах. Засвоїти способи аналізу масивів даних та упорядкування даних у масивах.

План

1. [Ознайомитись з основними принципами програмної обробки одномірних масивів та опрацювати наведені приклади на ПК.](#)
2. [Ознайомитись з основними принципами програмної обробки двомірних масивів та опрацювати наведені приклади на ПК.](#)
3. [Скласти самостійно алгоритм та програму обробки одномірного масиву, згідно варіанту.](#)
4. [Скласти самостійно алгоритм та програму обробки двомірного масиву, згідно варіанту.](#)
5. [Дати відповіді на контрольні запитання.](#)
6. [Оформити звіт](#)

Теоретичні відомості

1. Одновимірні масиви

Поряд з поняттям простої змінної при програмуванні користуються поняттям масиву

Масивом називають упорядковану послідовність величин, визначених одним іменем.

Під упорядкованістю розуміють розміщення величин (елементів масива) у послідовних комірках пам'яті. У такому розумінні масив можна порівняти з ящиками, що з'єднані разом і мають єдиний ярлик з іменем масива (наприклад, А). Усі ящики пронумеровані, починаючи з 1. Ящики - це елементи масива, номер ящика - значення індекса елемента масива

A1	A2	A2	A4	...	A _n
----	----	----	----	-----	----------------

Для доступу до елемента масива необхідно назвати ім'я масива та індекс елемента.

При програмуванні мовою Паскаль масиву надається ім'я, а індекси позначаються у квадратних дужках. Масив має бути описаний у розділі змінних (VAR), або у розділі типів TYPE).

Опис масива у розділі змінних VAR має вигляд

VAR <ім'я масива>: ARRAY [t1..t2] OF [t3];

ARRAY (масив), OF (із) - службові слова мови Паскаль,
t1..t2 - граничні значення індекса,
t3 - тип елементів масива.

Приклад: опис масивів різних типів

VAR

M: ARRAY [1..N] OF REAL;

G: ARRAY [1..N] OF INTEGER;

V: ARRAY [1..M] OF BOOLEAN;

При наявності декількох масивів одного й того ж типу їх можна об'єднати у один список.

Приклад: VAR A,B,C : ARRAY [1..50] OF REAL;

Визначено списком три масиви A,B,C дійсних чисел, кожен із яких має по 50 елементів:

A[1], A[2], ... ,A[50], B[1], B[2], ... ,B[50], C[1], C[2], ... ,C[50].

Оголошення масиву на C

Формат оголошення масиву у C/C++ програмі наведений нижче:

<тип даних> <ім'я масиву> [<1-а розмірність>]... [<N-а розмірність>];

Розмірності масиву - це константні цілі значення, що задають число елементів.

При наявності декількох розмірностей кількість елементів відповідає їх добутку.

Наприклад:

//оголошення одновимірного цілого масиву з 10 елементів

int Mas[10];

//оголошення двовимірного символьного масиву з 25 елементів

char Matrix[5][5];

//оголошення тривимірного дійсного масиву з 30 елементів

float Matrix[2][3][5];

Елементи масиву можна як і звичайні змінні визначати при оголошенні і в програмі. При оголошенні для визначення елементів масиву застосовується наступний формат:

BP : <ім'я масиву>:array[t1..t2] of <тип даних> =(<список значень>);

C/C++ : <тип даних> <ім'я масиву> [<розмірність>] = { <список значень> };

Слід звернути увагу на поняття "індекс" і "тип індексу"

Обробка масивів виконується при зміні індексів елементів.

Оскільки елементи масиву в області пам'яті записані послідовно, то індекс елемента пов'язаний з адресою комірки пам'яті, де він розташований. Зміна індексу елемента викликає зміну адреси комірки, тобто **переедресацію**.

Масив А - вектор, що складається з 10 елементів цілого типу зі знаком

індекс елемента	1	2	3	4	5	6	7	8	9	10
значення	12	0	10	5	-8	7	5	19	-123	224

Приклад 1. Знайти середнє значення додатніх елементів в одновимірному масиві x(2.5, 0, -5.3, 0.05, 10.2, -1, 0, 4.08):

```

program Project1;
uses crt;
const n=8;
      x:array[1..n] of real=(2.5, 0, -5.3, 0.05, 10.2, -1, 0,
4.08);
var i:integer; s, sr:real;
begin
clrscr;
  for i:=1 to n do
    write(' x',i,'=',x[i]:5:2);
  writeln;
  s:=0;
  for i:=1 to n do
    if x[i]>0 then s:=s+x[i];
  sr:=s/n;
  writeln('sr=',sr:7:2);
  readln
end.

```

[<Перейти до плану>](#)

2. Обробка двовірних масивів

Двовірні масиви являють собою набір елементів, упорядкованих у рядки, у кожному з яких міститься одна і та ж кількість елементів. Положення елементів визначається двома індексами: першим визначається номер рядка, другим - місце елемента у рядку (часто це місце називають стовбцем).

Масив В – матриця, що має 3 рядки і 4 стовпця, і складається з елементів дійсного типу.

рядок\стовпчик	1	2	3	4
1	2,1	0	-5,4-	3,14
2	7,8	9,81	0	0
3	4,6	8	11,2	-4,05

Оскільки ім'я масиву єдине для усіх елементів, для визначення адресації, необхідної при збереженні значень елементів у пам'яті комп'ютера, комп'ютеру слід повідомити про особливості використання імені масива. При програмуванні це виконується у описовій формі перед обробкою елементів.

Опис двохмірного масива (матриці) відбувається одним із способів:

у блоці VAR: var a:array [r1..r2,s1..s2] of <тип>;

у блоці TYPE: type a=array [r1..r2,s1..s2] of <тип>;

Звернення до елемента масива відбувається через ім'я масива по номеру рядка та стовбця: A[2,3], A[t,z], A[i+1,k*2] тощо.

Приклад 2. Знайти максимальне та мінімальне значення за стовпчиками двовірного масива:

$$x = \begin{bmatrix} 3 & 0 & 1 \\ 4.5 & 2.2 & -7.1 \\ 0 & 5.6 & -3 \\ 1.3 & -9.2 & 0 \\ -6.9 & 7.4 & 5.8 \end{bmatrix}$$

```

program Project2;
uses crt;
const n=5; m=3;
      x:array[1..n,1..m] of real=((3,0,1),(4.5,2.2,-7.1),(0,5.6,-
3),
      (1.3,-9.2,0),(-6.9,7.4,5.8));
var i,j:integer;
    min,max:real;
begin
  clrscr;
  writeln('Initial data:');
  for i:=1 to n do begin
    for j:=1 to m do
      write(x[i,j]:6:2);
      writeln;
    end;
    for j:=1 to m do begin
      min:=x[1,j];max:=x[1,j];
      for i:=2 to n do begin
        if x[i,j]<min then min:=x[i,j];
        if x[i,j]>max then max:=x[i,j];
      end;
      writeln('In column ',j,' min=',min:5:2,' max=',max:5:2);
    end;
    readln;
  end.

```

Приклад 3. Увести та упорядкувати масив x(2.5, 0, -5.3, 0.05, 10.2, -1, 0, 4.08) у порядку зростання значень його елементів:

```

program Progl;
uses Crt;
const n=8;
      x:array[1..n] of real=(2.5, 0, -5.3, 0.05, 10.2, -1, 0,
4.08);
var b:real;
    i,j:integer;
begin
  clrscr;
  writeln('Initial data:');
  for i:=1 to n do
    write(x[i]:6:2);
    for j:=1 to n do begin
      for i:=1 to n-j do
        if x[i]>x[i+1] then begin
          b:=x[i];x[i]:=x[i+1];x[i+1]:=b;
        end;
      end;
    end;
  end;

```

```
writeln('Result of sorting:');
for i:=1 to n do
  writeln('x',i,'=',x[i]:7:2);
readln
end.
```

Правила складання алгоритмів при розв'язанні задач з двовимірними масивами:

1. Для обробки n -мірних масивів використовуються вкладені цикли з глибиною вкладення n , причому логічний блок перевірки заданої умови і обчислювальний блок розміщуються у тілі внутрішнього циклу.

2. При знаходженні шуканої величини для всієї матриці початкове присвоєння здійснюється один раз до входу в зовнішній цикл, а результат виводиться після виходу із зовнішнього циклу.

3. Якщо шукана величина визначається для рядків або стовпців, то початкове присвоєння і виведення результату здійснюється число разів, рівне числу зовнішніх циклів. Блок початкового присвоєння розміщується на вході у внутрішній цикл, а блок виведення результатів - на виході з внутрішнього циклу.

4. Якщо шукана величина визначається для рядків, то зовнішній цикл організовується за рядками, а внутрішній - за стовпцями. Якщо шукана величина визначається для стовпців, то зовнішній цикл організовується за стовпцями, а внутрішній - за рядками.

При обробці матриць часто зустрічаються задачі, дії в яких здійснюються з елементами, розташованими певним чином відносно головної або побічної діагоналі, вертикальної або горизонтальної вісі симетрії матриці. Правильно скласти алгоритм рішення задачі у такому випадку допоможе таблиця 6.1, наведена нижче.

Таблиця 1

№	Розташування елементів матриці	Умова при послідовній обробці	Обробка матриць		Примітка
			за рядками	за стовпцями	
1	На головній діагоналі	$i=j$	$i:=1, n$	$j:=1, n$	Матриця $N \times N$ 1 цикл
2	На побічній діагоналі	$i+j=n+1$	$i:=1, n$	$j:=1, n$	Матриця $N \times N$
3	Вище головної діагоналі	$i < j$	$i:=1, n-1$ $j:=i+1, n$	$j:=2, n$ $i:=1, j-1$	Матриця $N \times N$
4	Нижче головної діагоналі	$i > j$	$i:=2, n$ $j:=1, n-1$	$j:=1, n-1$ $i:=j+1, n$	Матриця $N \times N$
5	Вище побічної діагоналі	$i+j < n+1$	$i:=1, n-1$ $j:=1, n-i$	$j:=1, n-1$ $i:=1, n-j$	Матриця $N \times N$
6	Нижче побічної діагоналі	$i+j > n+1$	$i:=2, n$ $j:=n+2-i, n$	$j:=2, n$ $i:=n+2-j, n$	Матриця $N \times N$
7	Правіше вертикальної вісі симетрії	$j > (n+1)/2 = [n/2]$	$i:=1, m$ $j:=n/2, n$	$j:=n/2, n$ $i:=1, m$	Матриця прямокутна $M \times N$
8	Нижче горизонтальної вісі симетрії	$i > (m+1)/2 = [m/2]$	$i:=m/2, m$ $j:=1, n$	$j:=1, n$ $i:=m/2, m$	

[<Перейти до плану>](#)

ЗАВДАННЯ (ЛР_7)

Задано масив $X \{-12.4, 0, 12.3, -1, 0.01, 0, 123.3, -18.97, 0, 0.8, 10.01, -1.2, 11.2, 12.3, 13.6, 0, -5.2, 12.3, -111.3, 45.7, -6.7, 0, 2.1\}$.

Скласти алгоритм та програму, де передбачити наступні дії

1. Ввести поелементно масив у пам'ять комп'ютера у діалозі;
2. Поелементно обробити масив згідно індивідуальних завдань (див. Таблицю 5(1));
3. Вивести результуючий масив (якщо його було змінено) та результат роботи.
4. Переписати програму та відповідь з екрану в зошит та показати викладачеві.
5. Оформити звіт.

Таблиця (1). Варіанти індивідуальних завдань

Варіанти	Виконати аналіз та знайти значення наступних величин
1	Переписати в інший масив підряд всі від'ємні елементи даного масиву, вилучивши їх з нього і замінивши прогалини додатними елементами, які залишились у масиві !
2	Утворити інший масив: елементи якого є частковими сумами елементів даного масиву
3	Знайти найбільший елемент і поміняти його місцем з останнім
4	Середнє арифметичне натуральних логарифмів додатніх елементів з непарними номерами та їх кількість.
5	Упорядкувати елементи масиву по збільшенню
6	Упорядкувати елементи масиву по зменшенню
7	Записати в цей масив спочатку всі від'ємні числа а потім всі додатні, зберігаючи порядок розташування елементів
8	Записати в цей масив спочатку всі додатні числа а потім всі від'ємні, зберігаючи порядок розташування елементів
9	Знайти найменший елемент і поміняти його місцем з най-більшим
10	Переписати в інший масив підряд всі додатні елементи даного масиву, вилучивши їх з нього і замінивши прогалини від'ємними елементами, які залишились у масиві
11	Підрахувати добуток ненульових елементів та кількість додатніх
12	Кількість елементів, які належать до проміжку $[-1 \text{ до } 11]$, надрукувати також номери елементів та їх значення
13	Суму $S1$ елементів, які відрізняються від 0 з парними номерами та їх кількість $K1$, та суму $S2$ елементів, відрізняються від 0 та їх кількість $K2$.
14	Середнє арифметичне від'ємних елементів і їх кількість
15	Середнє геометричне додатніх елементів і їх кількість

Пояснення:

Середнє арифметичне елементів масиву з N елементів $S = \frac{1}{n} \sum_{i=1}^n X_i$

Середнє геометричне K елементів, які більші від 0 $P = \sqrt[k]{\prod_{X_i > 0} X_i}$

Часткова сума $S_k = \sum_{i=1}^k X_i, k=1,2..n$

Приклад реалізації завдання.

В одновимірному масиві, що складається з 7 елементів, знайти суму додатних елементів.

Рішення**1. Постановка задачі.**

Результатом рішення задачі є сума додатних елементів одновимірного масиву. Для розрахунку і зберігання результату буде використовуватися змінна-накопичувач S. Змінюючи індекс i елемента масиву x від початкового значення 1 до кінцевого значення 7 з кроком 1, при кожному повторенні циклу будемо перевіряти, чи є значення елемента масиву додатним. Якщо «так», то змінну S будемо сумувати зі значенням елемента масиву. Якщо «ні», то такий елемент масиву будемо пропускати. Отже, для рішення задачі має бути заданий масив x з 7 чисел.

2. Побудова математичної моделі.

Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Масив з 7 чисел	Дійсний	x	Вихідне значення
Змінна циклу	Цілий	i	Допоміжна змінна
Сума додатних елементів	Дійсний	S	Результат

Початкове значення суми додатних елементів дорівнює 0. Для накопичення суми будемо використовувати формулу $S=S+x_i$.

3. Розробка алгоритму розв'язання задачі.

Для реалізації алгоритму будемо використовувати варіант циклу з параметром (Рис.6.1).

У блоці 1 здійснюється введення елементів одновимірного масиву. У блоці 2 змінній S присвоюється початкове значення.

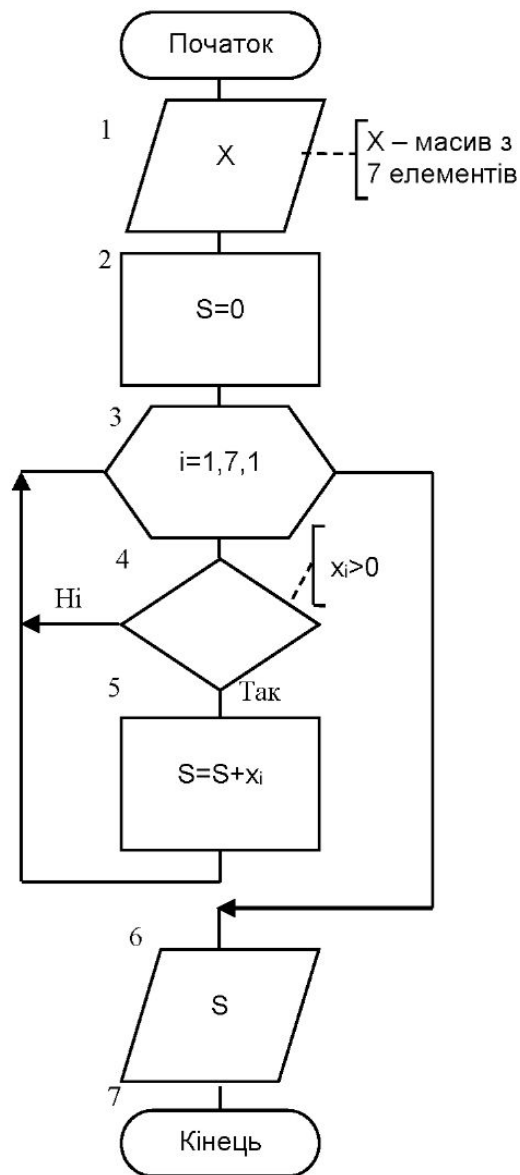


Рис.6.1

У блоці 3 задаємо параметри роботи циклу з переадресацією. Блоки 4-5 є тілом циклу. Блок 4 Рішення перевіряє, чи є поточний елемент масиву x додатним. Якщо відповідь «так», то в блоці 5 відбувається накопичення суми додатних елементів S .

Блок 6 служить для виведення результатів рішення задачі.

Перевіримо правильність алгоритму на конкретних, довільно взятих значеннях вихідних даних, наприклад $x=(-1;3;-5;7;8;9;-10)$. Виконання дій за блок-схемою оформимо у вигляді таблиці.

Блок	Цикл 1	Цикл 2	Цикл 3	Цикл 4	Цикл 5	Цикл 6	Цикл 7	Цикл 8
	Початок							
1	Введення: (-1;3; -5;7;8; 9;-10)							
2	S=0							
3	i=1 1<=7 так (перехід на блок 4)	i=2 2<=7 так (перехід на блок 4)	i=3 3<=7 так (перехід на блок 4)	i=4 4<=7 так (перехід на блок 4)	i=5 5<=7 так (перехід на блок 4)	i=6 6<=7 так (перехід на блок 4)	i=7 7<=7 так (перехід на блок 4)	i=8 8<=7 ні (перехід на блок 6)
4	-1>0 ні (перехід на блок 3)	3>0 так (перехід на блок 5)	-5>0 ні (перехід на блок 3)	7>0 так (перехід на блок 5)	8>0 так (перехід на блок 5)	9>0 так (перехід на блок 5)	-10>0 ні (перехід на блок 3)	
5		S=0+3=3 (повернення на блок 3)		S=3+7=10 (повернення на блок 3)	S=10+8=18 (повернення на блок 3)	S=18+9=27 (повернення на блок 3)		
6								Виведення 27
7								Кінець

Отриманий результат збігається з очікуванням, отже, схема алгоритму складена вірно.

TP

```
program p_2_3_3;
uses CRT;
var x:array [1..7] of real; {об'явлення одновимірною масиву x, що складається з семи дійсних елементів }
```

```
    i:integer; s:real;
begin
    clrscr;
    for i:=1 to 7 do begin {починаючи з першого по сьомий, вводяться елементи масиву x }
        writeln('Введіть елемент масиву x');
        readln(x[i]);
    end;
    s:=0; { початкове значення суми дорівнює нулю }
    for i:=1 to 7 do begin { починаючи з першого по сьомий, додатні елементи масиву сумуються }
        if x[i]>0 then s:=s+x[i];
    end;
    writeln(s); {виведення значення суми }
    readln
end.
```

C++

```
#include<iostream>
#include<locale>
```

```

using namespace std;
int main()
{
    double S;
    double x[8];
    int i;
    setlocale(LC_ALL, "UKR");

    cout<<"Введіть елементи масиву x ";

    for(i=1;i<=7;i++) //введення масиву x
    {
        cout<<"x["<<i<<"]=";
        cin>>x[i];
    }

    S=0;

    for(i=1;i<=7;i++)//з першого по 7-ий елемент масиву x з кроком 1 робити
        if (x[i]>0) S+=x[i];//якщо x>0, то накопичувати суму

    cout<<"S="<<S<<endl;//виведення результату

    system("pause");
    return 0;
}

```

[<Перейти до плану>](#)

ЗАВДАННЯ (ЛР_8)

Скласти алгоритм та програму, де передбачити наступні дії

1. Ввести поелементно масив у пам'ять комп'ютера у діалозі;
2. Поелементно обробити масив згідно індивідуальних завдань(див. Таблицю5(2));
3. Вивести результуючий масив (якщо його було змінено) та результат роботи.
4. Переписати програму та відповідь з екрану в зошит та показати викладачеві.
5. Оформити звіт.

Таблиця (2). Варіанти індивідуальних завдань

Виконати обробку масива даних:

1. Увести початкові дані.
2. Виконати завдання згідно варіанту.
3. Вивести на екран початкові дані у вигляді матриці та результат обробки масиву.

№ варіанта	Початкові дані	Завдання
1	$A = \begin{bmatrix} 2.5 & -3.6 & 7.9 & 0.1 \\ 0 & 8.2 & 9.6 & -5.0 \\ 0 & 0 & 0 & 3.9 \end{bmatrix}$	Підрахувати кількість елементів матриці, що дорівнюють нулю. Вивести індекси цих елементів.

2	$B = \begin{bmatrix} 2 & 5 & 0 & 3 \\ 4 & -8 & 0 & 6 \\ 9 & 7 & -4 & 3 \\ 5 & -4 & 0 & 6 \end{bmatrix}$	Підрахувати суму елементів матриці, розміщених на головній діагоналі
3	$C = \begin{bmatrix} 0 & -3 & 0 \\ 0 & 0 & 4 \\ 1 & 5 & 0 \\ 2 & 6 & 0 \end{bmatrix}$	Підрахувати кількість нульових елементів у кожному стовпчику матриці.
4	$D = \begin{bmatrix} 7.2 & -8.3 & 0 & 9.1 & 8.7 \\ 0 & 4.4 & 6.8 & -5.4 & 9.2 \end{bmatrix}$	Знайти суму елементів у кожному рядку матриці.
5	$E = \begin{bmatrix} 2.1 & 0 & -3.2 \\ 6.8 & 7.4 & 0 \\ 5.8 & 2.9 & -6.1 \end{bmatrix}$	Знайти добуток елементів матриці, що знаходяться на головній діагоналі.
6	$F = \begin{bmatrix} 5 & 6 & 4 \\ 3 & 2 & 1 \\ 0 & -8 & 7 \\ -2 & 0 & 6 \end{bmatrix}$	Знайти максимальний елемент у кожному стовпчику матриці.
7	$G = \begin{bmatrix} 2 & 4 & -5 & -1 \\ 0 & 3 & -8 & 7 \\ 9 & 6 & 0 & -5 \end{bmatrix}$	Знайти мінімальний елемент у кожному рядку матриці.
8	$H = \begin{bmatrix} 2 & 0 & 3 & 1 & 6 \\ -7 & 0 & 4 & 5 & -2 \\ 3 & 6 & -5 & 1 & 7 \\ 0 & 5 & 4 & 3 & 2 \\ 1 & 0 & -8 & 6 & 9 \end{bmatrix}$	Знайти добуток ненульових елементів у кожному рядку матриці.
9	$I = \begin{bmatrix} 8 & 7 & 5 \\ 6 & 4 & 3 \\ 2 & 1 & 0 \\ 1 & 2 & 3 \end{bmatrix}$	Упорядкувати елементи у кожному стовпчику матриці за зростанням значень.
10	$J = \begin{bmatrix} 7 & 5 & 8 & 0 \\ 1 & 4 & -6 & 2 \\ -3 & 2 & 1 & 0 \end{bmatrix}$	Упорядкувати елементи у кожному рядку матриці за зменшенням значень.
11	$K = \begin{bmatrix} 5.1 & -8.2 & 9.6 & 0 \\ 4.3 & 7.6 & -2.5 & -1.1 \\ 0 & 3.6 & 4.8 & 0 \end{bmatrix}$	Знайти середнє значення для кожного рядка матриці.
12	$L = \begin{bmatrix} 7 & -5 & 2 & 0 \\ 8 & 0 & 6 & 4 \\ -3 & 2 & 1 & 0 \end{bmatrix}$	Знайти мінімальний елемент для кожного стовпчика матриці.

13	$M = \begin{bmatrix} 5 & 2 & 1 \\ 0 & -3 & 6 \\ 8 & 2 & 0 \end{bmatrix}$	Визначити суми елементів для кожного стовпчика матриці.
14	$N = \begin{bmatrix} 2 & 3 & -8 & 7 & 0 \\ 6 & -5 & 4 & 3 & 2 \\ 1 & 0 & 5 & -3 & 6 \end{bmatrix}$	Вивести на екран індекси від'ємних елементів матриці.
15	$O = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \\ 0 & 6 & 5 & 4 \end{bmatrix}$	Підрахувати кількість нульових елементів для кожного рядка матриці.

Приклад реалізації завдання.

Умова задачі: Визначити добуток додатних і кількість від'ємних елементів у прямокутній матриці A , що складається з m рядків і n стовпців. Прийняти, що нульових елементів у матриці немає.

Рішення

1. Постановка задачі.

Для рішення цієї задачі необхідно перебрати в будь-якому порядку елементи матриці A , з додатних елементів у процесі перебору накопичити добуток, а для від'ємних елементів організувати лічильник, що підраховує їх кількість.

2. Побудова математичної моделі.

Складемо таблицю імен змінних (таб.6.2).

Таблиця 6.2

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Дійсний	A	Вихідне дане
Кількість рядків	Цілий	M	Вихідне дане
Кількість стовпців	Цілий	N	Вихідне дане
Змінна циклу за рядками	Цілий	i	Допоміжна змінна
Змінна циклу за стовпцями	Цілий	j	Допоміжна змінна
Добуток додатних елементів	Дійсний	P	Результат
Кількість від'ємних елементів	Цілий	K	Результат

Початкове значення добутку додатних елементів дорівнює 1. Для накопичення добутку будемо використовувати формулу $P=P*A_{i,j}$. Початкове значення кількості від'ємних елементів дорівнює 0. Для підрахунку кількості будемо використовувати формулу $K=K+1$.

3. Розробка алгоритму розв'язання задачі.

Для реалізації алгоритму будемо використовувати варіант вкладених циклів з параметром (Рис. 6.4.).

У блоці 1 виконується введення кількості рядків і стовпців матриці, а також введення значень елементів двовимірного масиву. Потрібно пам'ятати, що таке позначення введення двовимірного масиву є умовним. Насправді, введення і виведення елементів матриці - це не одноразова дія, а циклічний процес, що здійснюється у вкладених циклах. Блок-схема введення-виведення двовимірного масиву зображена на рисунку 6.3.

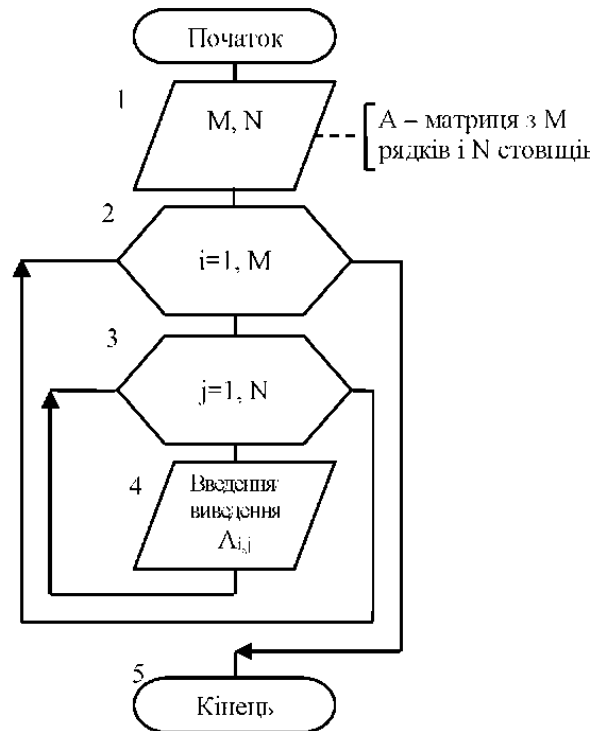


Рисунок 6.3.

У блоці 2 змінним P і K присвоюються початкові значення.

У блоці 3 задаємо параметри роботи зовнішнього циклу з переадресацією. У блоці 4 задаємо параметри роботи внутрішнього циклу з переадресацією. Блоки 5-7 є тілом внутрішнього циклу. Блок 5 Рішення перевіряє, чи є поточний елемент масиву A додатним. Якщо відповідь «так», то в блоці 6 відбувається накопичення добутку додатних елементів P . Якщо відповідь «ні», то в блоці 7 відбувається підрахунок кількості від'ємних елементів K .

Блок 8 використовується для виведення результатів рішення задачі.

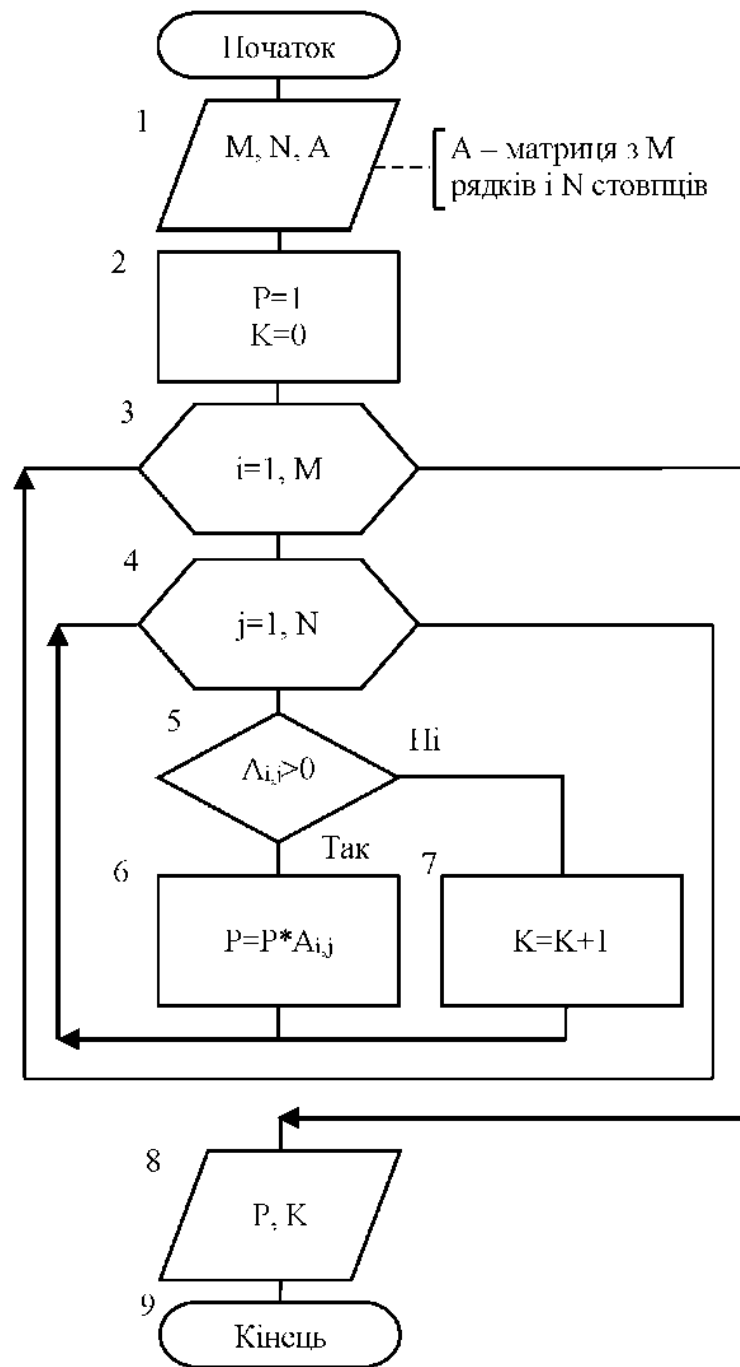


Рисунок 6.4.

Перевіримо правильність алгоритму на конкретних, довільно взятих значеннях вихідних даних, наприклад $M=2$, $N=2$, $A=(-1;3;5;-7)$. Виконання дій за блок-схемою оформимо у вигляді таблиці (табл. 6.3.).

Блок	Цикл 1	Цикл 2	Цикл 3	Цикл 4	Цикл 5	Цикл 6	Цикл 7
	Початок						
1	Введення: 2;2; (-1;3; 5;-7)						
2	P=1 K=0						
3	i=1 1<=2 так (перехід на блок 4)			i=2 2<=2 так (перехід на блок 4)			i=3 3<=2 ні (перехід на блок 8)
4	j=1 1<=2 так (перехід на блок 5)	j=2 2<=2 так (перехід на блок 5)	j=3 3<=2 ні (перехід на блок 3)	j=1 1<=2 так (перехід на блок 5)	j=2 2<=2 так (перехід на блок 5)	j=3 3<=2 ні (перехід на блок 3)	
5	-1>0 ні (перехід на блок 7)	3>0 так (перехід на блок 6)		5>0 так (перехід на блок 6)	-7>0 ні (перехід на блок 7)		
6		P=1*3=3 (перехід на блок 4)		P=3*5=15 (перехід на блок 4)			
7	K=0+1=1 (перехід на блок 4)				K=1+1=2 (перехід на блок 4)		
8							Виведення 15; 2
9							Кінець

Отриманий результат збігається з очікуванням, отже схема алгоритму вірна.

Програмна реалізація

BP

```

program p_matr;
uses CRT;
var
  A:array [1..100,1..100] of real; {об'ява дійсного двовимірного
                                     масиву/матриці A, що складається з 100 рядків та 100
                                     стовпців }
  i, j, m, n, k:integer; p:real;
begin
  clrscr;
  writeln('Введіть значення m,n, та матрицю A');
  readln(m,n); { введення кількості рядків m та кількості стовпців n }
  for i:=1 to m do { зовнішній цикл по рядках, номер рядка змінюється
                    від 1 до m з кроком 1 }
    for j:=1 to n do { вкладений цикл по стовпцях, номер стовпця
                     змінюється від 1 до n }
      readln(a[i,j]); { введення елементів матриці A }
  k:=0; p:=1; {початкове значення добутку додатних елементів
               матриці дорівнює 1, а кількість від'ємних - 0 }

```

```

for i:=1 to m do { зовнішній цикл по рядках, номер рядка змінюється
від 1 до m з кроком 1 }
  for j:=1 to n do { вкладений цикл по стовпцях, номер стовпця
                    змінюється від 1 до n }
    if A[i,j]>0 then p:=p*A[i,j] else k:=k+1; {якщо елемент
        додатній, накопичується добуток p, інакше -
        кількість від'ємних елементів k збільшується на
        одиницю}
  writeln (p, k); { виведення значення p та k }
  readln
end.

```

C++

```

#include<stdio.h>
#include<conio.h>
#include<iostream.h>

void main()
{
    float P;
    int i,j,M,N,K;
    cout<<"M= "; cin>>M;
    cout<<"N= "; cin>>N;
    float A[M][N];
    clrscr();
    for(i=1;i<=M;i++) //введення матриці A по рядкам
        for(j=1;j<=N;j++)
        {
            cout<<"A["<<i<<"]["<<j<<"]=" ";
            cin>>A[i][j];
        }
    P=1;
    K=0;
    for(i=1;i<=M;i++) //обробка матриці
        for(j=1;j<=N;j++)
            if(A[i][j]>0) P*=A[i][j];
            else K++;
    printf("Добуток додатніх елементів = %f\n",P); //виведення результату
    printf("Кількість від'ємних елементів = %d\n",K);
    getch()
}

```

[<Перейти до плану>](#)

Запитання для самоконтролю

1. Що називається масивом?
2. Що таке ідентифікатор масиву?
3. Як ввести одновимірний масив у діалозі?
4. Як вивести одновимірний масив у рядок, у стовбець?
5. Як ввести двовимірний масив у діалозі?
6. Як вивести двовимірний масив у формі таблиці?
7. Що таке розмірність масиву, як задається?
8. Як отримати доступ до елементів масиву, навести приклади?
9. Як визначити добуток елементів кожного окремого стовпця матриці?
10. Як визначити добуток елементів кожного, окремого рядка матриці?
11. Як визначити максимальний елемент вектора?
12. Які способи опису масивів вам відомі, навести приклади?
13. Які алгоритми сортування масиву вам відомі. Навести приклади фрагменту програми .
14. Як поміняти сусідні елементи масиву місцями.
15. Якій умові відповідають індекси діагональних елементів матриці (головної і другорядної).
16. Якій умові відповідають індекси елементів матриці, що знаходяться вище, нижче головної діагоналі.

[<Перейти до плану>](#)

Вимоги до оформлення звіту

1. Титульний лист :

- - Міністерство освіти і науки ЦНТУ кафедра ПКСМ;
- № лаб. роботи, назва теми роботи ;
- Мета роботи, завдання до роботи;
- ПППб виконавця та № варіанту;
- дата виконання та дата оформлення звіту;
- ПППб викладача, що перевірів.

З 2-го листа

1. Умова задачі
2. Постановка задачі.
3. Формалізація змісту задачі та метод розв'язку.
4. Блок-схема алгоритму та опис його роботи;
5. Листинги програм мовами ВР та C/C++ (роздруківка, або від руки переписати з екрану ПК).
6. Протокол виконання програми та висновки.

Власний підпис