Комп’ютерний практикум №8

МАСИВИ

БАГАТОВИМІРНІ МАСИВИ

*Мета роботи* - опанувати технологію використання двовимірних масивів даних (матриць), навчитися розробляти алгоритми та програми із застосуванням матриць.

* 1. Теоретичні відомостi

У загальному випадку масив може містити компоненти будь-якого типу, зокрема, також масиви. Такий масив у C/С++ трактується як масив, що складається з декількох масивів. Масив масивів є більш загальною структурою даних і дозволяє задати не тільки багатовимірний куб, а й порізану, ступінчасту структуру.

Формат оголошення такого двовимірного масиву у С/С++:

тип ім'я[кількість\_1][кількість\_2].

Наприклад,

int a[2][3].

Оскільки багатовимірні масиви в мові C/С++ - це масиви масивів, то при оголошенні таких масивів в оперативній пам'яті створюється кілька різних об'єктів (виділяється пам'ять під них):

* покажчик на двовимірний масив (його ім'я співпадає з іменем масиву);
* масив покажчиків, кожен із яких містить адресу масиву-рядка вихідного двовимірного масиву;
* двовимірний масив даних базового типу.

Наприклад, при виконанні оголошення двовимірного масиву int arr[4][3] в програмі створюється покажчик arr, який визначає в пам'яті місце розташування першого елемента масиву і, крім того, є покажчиком на масив з чотирьох покажчиків, кожен з яких містить адресу одновимірного масиву, що представляє собою рядок вихідного двовимірного масиву і складається з трьох елементів типу int (рис. 1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| А |  | | | |
| ⇓ |
| А [0] | ⇒ | А [0] [0] | А [0] [1] | А [0] [2] |
| А [1] | ⇒ | А [1] [0] | А [1] [1] | А **[1] [2]** |
| А **[2]** | ⇒ | А [2] [0] | А [2] [1] | А [2] [2] |
| А [3] | ⇒ | А [3] [0] | А [3] [1] | А [3] [2] |
| Рис. 1 Розподіл пам'яті для двовимірного масиву | | | | |

В оперативній пам’яті двовимірний масив даних базового типу зберігається як безперервна послідовність елементів. При цьому компонента з найменшим значенням індексу розміщується за найменшою адресою (самий правий індекс елементів багатовимірних масивів зростає найпершим). Наприклад, елементи матриці

int A[5][4];

в оперативній пам’яті будуть розміщені у такий спосіб:

A[0][0], ..., A[0][3], A[1][0], ..., A[1][3], ..., A[4][0], ..., A[4][3].

Доступ до окремого елемента двовимірного масиву може здійснюватися за допомогою операції індексування наступного формату:

ім'я [індекс\_1][індекс\_2].

Наприклад, А[1][2].

Як індекси можуть використовуватися константи і довільні вирази, що включають змінні порядкових типів.

Також доступ до елементів двовимірного масиву може здійснюватися за покажчиком. Наприклад,

int А[4][3];

int \*p;

p = &А[0][0];

p++; *//перехід до наступного елемента*.

Оскільки елементи багатовимірних масивів розташовуються в пам'яті підряд по рядках, то такий порядок дає можливість звертатися до будь-якого елементу багатовимірного масиву, використовуючи адресу його початкового елемента і тільки один індексний вираз. Наприклад, звернення до елементу А[1][2] можна здійснити за допомогою покажчика p, оголошеного у формі int \*p = А[0], як звернення p[5].

До цього елемента також можна звертатися у еквівалентних формах: \*(\*(А+1)+2) і (\*(А +1))[2].

Обробка елементів двовимірного масиву виконується у вкладених циклах for.

Як і одновимірні, так і двовимірні масиви можуть бути параметрами підпрограм. Передача такого параметра-масива у функцію здійснюється тільки за адресою. Можливі формати оголошення масивів-параметрів:

тип ім'я [кількість\_1][кількість\_2];

тип ім'я [][кількість\_2].

При передачі у функцію багатовимірного масиву розмірності усіх індексів, окрім першого, треба вказувати обов’язково

Перелік базових операцій над двовимірними масивами та їх елементами:

1. введення-виведення матриці, її ініціалізація;
2. створення нової матриці за заданим алгоритмом;
3. пошук елементів матриці за певним критерієм;
4. визначення, чи задовольняє матриця або окремі її елементи певній властивості;
5. виконання певних операцій над компонентами матриць (переставлення рядків і стовпців, множення матриць тощо).

Базові операції обробки матриць слід реалізовувати у вигляді функцій, які згодом можуть бути використані як «архітектурні блоки» при розв’язанні більш складних задач.

* 1. Вимоги до програми

1. Відповідність структурному підходу.
2. Виведення вхідних, проміжкових і вихідних даних.
3. Наявність прототипів функцій.
4. Генерація масивів випадковим чином.
5. Використання параметрів-масивів.
   1. Приклад програми

***Задача***. У заданій матриці *А(n* х *m)* видалити вказаний рядок*.*

#include <iostream.h>

#include <stdlib.h>

#include <time.h>

#include <iomanip.h>

using namespace std;

typedef int Matrix[10][10]; //тип матриці

Matrix А; //вихідна матриця

int n, m; //кількість рядків і стовпчиків матриці

void input(Matrix); // генерація матриці

void output(Matrix); // виведення матриці

void delete\_row(Matrix,int); //видалення рядка матриці

//=============== головна функцiя ================

int main()

{ int k; //номер рядка, що видаляється

cout<<"Input n, m: ";

cin>>n>>m; //ввести розмірність матриці

input(А); //генерувати матрицю

output(А); //вивести згенеровану матрицю

cout<<"Input number of row:";

cin>>k; //ввести номер рядка, що видаляється

delete\_row(А, k); //видалити вказаний рядок

output(А); //вивести отриману матрицю

system("pause");

}

//============== генерація матриці =================

void input(Matrix matr)

{ srand(time(NULL));

for (int i=0; i<n; i++)

for (int j=0; j<m; j++)

matr[i][j]=rand()%9; //генерація поточного елемента матриці

}

//=============== виведення матриці =================

void output(Matrix matr)

{ for (int i=0; i<n; i++)

{ for (int j=0; j<m; j++)

cout<<setw(4)<<matr[i][j]; //виведення елементів рядка

cout<<endl; //перехід на новий рядок

}

}

//======== видалення вказаного рядка матриці ========

void delete\_row(Matrix matr, int k1)

{ for (int i=k1-1; i<n-1; i++) //перегляд рядків, які будуть зсуватися

for (int j=0; j<m; j++)

matr[i][j]=matr[i+1][j]; //зсув рядка

n--; //зменшення загальної кількості рядків

}

* 1. Варіанти завдань

1. Доповнити матрицю *A(m x n) (m+1)*-м рядком та *(n+1)*-м стовпцем, в які записати суми елементів відповідних рядків та стовпців вихідної матриці. В елемент *A (m+1, n+1)* помістити суму всіх елементів матриці *A*.
2. На основі заданої квадратної матриці *А(n x n)* побудувати матрицю *В(n x n)*, елементи *bij*якої визначаються наступним чином: через *aij* проводяться діагоналі, паралельні головній та побічній; значення *bij* дорівнює максимальному елементу затушованої частини.



1. На основі заданої матриці *А (n х n)* побудувати вектор *Х(n)*, елементи якого *хi* - скалярний добуток *і-*го рядка матриці *А* на стовпець, що містить найбільший елемент в цьому рядку.
2. Задана дійсна квадратна матриця розмірності *n*, яка містить і від’ємні елементи. Знайти най­­­більший за модулем її елемент. Отримати квадратну матрицю розмірності *n-1* шляхом вилучення із вхідної матриці рядка та стовпчика, на пе­­­ретині яких знаходиться знайдений елемент.
3. На основі заданої матриці *А (n х n)* побудувати вектор *Х(n)* наступним чином: рядки матриці впорядкувати по зменшенню елементів її першого стовпця і як *Х* прийняти головну діагональ перетвореної матриці.
4. Для заданих дійсних квадратних матриць *A (n* x *n)* та *B (n* x *n)* визначити середнє арифметичне їх *М*-норм, які обчислюються за формулою

 .

1. Знайти максимальний елемент і суму елементів квадратної матриці розмірності *n* в її затушованій частині. Вивести ці значення, а також елементи, розміщені в затушованій частині.
2. Задані три дійсні квадратні матриці розмірності *n х n*, що містять від'ємні елементи. З’ясувати, яка з цих матриць має найбільшу суму елементів, що розміщені вище її головної діагоналі. Замінити отриманим значенням всі від'ємні елементи цієї матриці.



1. Для введеної квадратної матриці *А* розмірності *n* одержати значення на перетині головної та побічної діагоналей. Сформувати нову матрицю *В(n х n)*, помінявши місцями затушовані частини матриці *А*.
2. Задані три цілочисельні матриці розмірності *m x m*, що містять від'ємні елементи. З’ясувати, яка з цих матриць має найменшу суму модулів діагональних елементів тих рядків, що починаються з від'ємного елемента. Упорядкувати рядки визначеної матриці за збільшенням значень її елементів.
3. Задані дійсні матриці *С(n x m)* та *В(n x m)*. Oдержати нові матриці *Y(n x m)* та *Z(n x m)* шляхом обміну місцями стовпців, що містять найбільше і найменше значення їх елементів.
4. Задані цілочисельні матриці *A (n* х *m)* та *B (m* х *р)*. Обчислити значення *А*⋅*В* , де , *i* = 1, …, *n*; *j* = 1, …, *p*; *k* = 1, …, *m*.
5. Дані дійсні числа *a1, ..., an*. Одержати квадратну матрицю порядка *n*, яка має наступний вигляд:

.



1. По заданій квадратній матриці *А* побудувати матрицю *В* тієї ж розмірності, кожен елемент *bij*якої дорівнює мінімальному елементу трикутника в матриці *А*, що визначається головною діагоналлю матриці та вершиною *аij* .
2. Нехай задані координати *n* точок на площині: *x1, y1, x2, y2, ..., xn, yn*. Знайти номера двох точок, відстань між якими є найбільшою (вважати, що така пара точок одна).
3. По заданій квадратній матриці *А* побудувати нову матрицю *В* тієї ж розмірності, кожен елемент *bij*якої одержується як середнє арифметичне сусідів елемента *aij*по рядкам та стовпцям (сусідами для елемента *aij*є елементи *ai-1,j , ai+1,*j, *a i,j-1 , ai,j+1*).
4. Побудувати квадратну матрицю з елементами 1, 2, ..., n, розміщеними по спіралі, починаючи з лівого верхнього кута. Знайти мінімальне значення кутових елементів отриманої матриці.
5. Задана дійсна квадратна матриця розмірності *n х n*. Переставити її стовпці таким чином, щоб упорядкувати за збільшенням значення елементів головної діагоналі.
6. Із матриці *А(n х m)* побудувати вектор *Х (n)* так, що компоненти вектора *Х* - впорядковані по зменшенню середні арифметичні значення елементів рядків матриці *А*.
7. Задані три дійсні квадратні матриці розмірності *n*, що містять від'ємні елементи. З’ясувати, яка з цих матриць має найменшу суму найбільших по модулю елементів її рядків. Замінити знайденим значенням елементи головної діагоналі визначеної матриці.
8. Задані дійсні матриці *С(n x m)* та *В(n x m)*. Oдержати нові матриці *Y(n x m)* та *Z(n x m)* шляхом обміну місцями рядків, що містять найбільше і найменше значення їх елементів.
9. Обчислити

 ,

де *А, В* - цілочисельні квадратні матриці розмірності *n*.

1. Задана дійсна квадратна матриця. Переставити її рядки таким чином, щоб упорядкувати за зменшенням значення елементів головної діагоналі.
2. Знайти максимальне значення серед елементів матриці *А (n х n)*, розміщених над її головною діагоналлю. Видалити стовпець матриці *А*, що містить знайдений елемент.
3. Із матриці *Х(n х m)* побудувати матрицю *Y(m х n),* помінявши місцями рядки та стовпці матриці *Х*.
4. У двох заданих дійсних квадратних матрицях розмірності *n* поміняти місцями їх головні діагоналі, попередньо упорядкувавши за збільшенням елементи цих діагоналей.
5. Задані цілочисельні матриці *A (m x n)* та *M (m x n)*, що містять від'ємні елементи. Нехай *В (n)* і *T (n)* - середні арифметичні елементів стовпців заданих матриць. Обчислити .
6. Задані цілочисельні матриці *A (n х n)* та *B (n х n)*. У кожній із них поміняти місцями мінімальні елементи рядка і відповідні елементи головної діагоналі. Побудувати матрицю *Q (n х n)*, елементи якої обчислюються за формулою .
7. Для заданої матриці *В(n х n)* вивести усі сідлові точки та їх індекси. Сідлова точка *bij* визначається як мінімальне значення в *i*-му рядку і максимальне значення в *j*-му стовпці матриці. Якщо сідлових точок у цій матриці немає, то вивести відповідне повідомлення.
8. Визначити, чи є задана ціла квадратна матриця магічним квадратом, тобто такою, у якої суми елементів по кожному стовпцю, рядку та кожній з двох діагоналей рівні між собою.
   1. Контрольні питання
9. Яким є принцип зображення двовимірного масиву в оперативній пам’яті?
10. Як здійснюється доступ до елементів двовимірного масиву?
11. Означити типові операції над двовимірними масивами.
12. Чи можна у С/С++ передавати у функцію параметр-масив, який може мати різну розмірність?
13. Чи можна у С/С++ передавати у функцію рядок матриці, як одновимірний масив?
14. У чому полягає алгоритм множення матриць?
    1. Структура звіту з лабораторної роботи

Звіт оформлюється у вигляді документу та складається із наступних частин:

1. Номер роботи та автор
2. Умова задачі
3. Блок-схема алгоритму вирішення задачі, виконана у Visio
4. Текст усіх файлів проекту
5. Копії екранних форм результатів роботи
6. Висновок, що містить обґрунтування коректності розробленого проекту