МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра КИТ

**Звіт  
по навчальній практиці**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Дніпро, 2019

**Постановка задачі**

Розробити та реалізувати структуру даних, що відображає роботу з розрідженою матрицею (не менше 70% елементів дорівнюють нулю). Необхідно реалізувати наступні операції:

* додавання елемента матриці (первинне розміщення елемента при створенні матриці, додавання елемента у вже створену матрицю – заміна нуля на інше);
* заміна значення *a*i,j на інше;
* видалення елемента матриці за заданими індексами та по значенню;
* пошук елемента матриці за заданими індексами;
* друк матриці у звичному форматі та форматі розрідженої матриці – індекси та значення. Наприклад, форма для одного елементу (i,j) *a*i,j;
* індивідуальне завдання (може бути оформлено у вигляді декількох функцій, для поліпшення структурованості програми).

|  |  |  |
| --- | --- | --- |
| рівень | завдання | структура |
| C | рівень A | 2 |

**Індивідуальне завдання:**

18. Сформувати одновимірний масив з середньо арифметичних елементів рядків матриці.

**Зовнішні специфікації**

Формат вхідних даних

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Кількість стовпців | arr.rows | Ціле позитивне число | 5 |
| 2 | Кількість рядків | arr.cols | Ціле позитивне число | 4 |
| 3 | Номер меню | number | Ціле позитивне число в діапазоні від 1 до 6 | 1 |
| 5 | Номер стовпця | coordinate\_i | Ціле позитивне число |  |
| 6 | Номер рядка | coordinate\_j | Ціле позитивне число | 100 |

Формат вихідних даних

|  |  |  |
| --- | --- | --- |
| № | Вхідні дані | Тестове повідомлення |
| 1 | Одновимірний масив з середньо геометричного ненульових елементів стовбців матриці | Масив з середньо арифметичних елементів рядків матриці:  56 89 1 23 0 |
| 2 | Пошук елемента за заданими індексами | Є такий елемент: <2;1> 23 |

**Функціональні вимоги до програми**

Програма повинна реалізувати наступні дії:

* додавання елемента матриці (первинне розміщення елемента при створенні матриці, додавання елемента у вже створену матрицю – заміна нуля на інше);
* заміна значення *a*i,j на інше;
* видалення елемента матриці за заданими індексами та по значенню;
* пошук елемента матриці за заданими індексами;
* друк матриці у звичному форматі та форматі розрідженої матриці – індекси та значення. Наприклад, форма для одного елементу (i,j) *a*i,j;
* індивідуальне завдання (може бути оформлено у вигляді декількох функцій, для поліпшення структурованості програми).

**Вибір методу рішення завдання**

Алгоритм розробки розрідженої матриці:

1. Створюємо двовимірний масив, яка задана користувачем.

2. Заповнюємо її нулями.

3. Далі вираховуємо 30% не нульових елементів.

4. Створюємо динамічний одновимірний масив для індексів по рядках.

5. Заповнюємо його випадковими числами на проміжку [0; кількість рядків - 1], та сортуємо його.

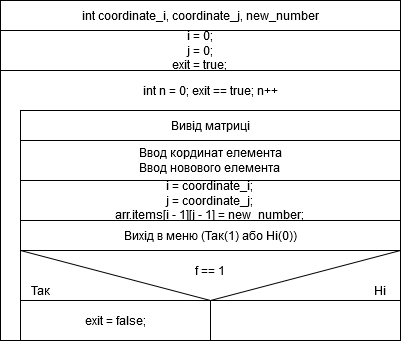
6. Так само створюємо масив для індексів по стовпцям

7. Заповнюємо його випадковими числами на проміжку [0; кількість стовпців - 1]

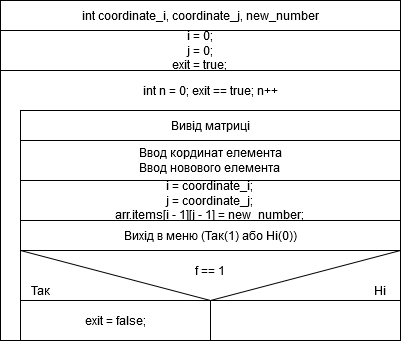
8. Беремо елементи з масиву по рядку та по стовпцях (беремо елементи в обох масивах одинакові по індексах) і по цим індексам в двовимірний масив ми заповняємо випадкове число.

**Алгоритм програми**

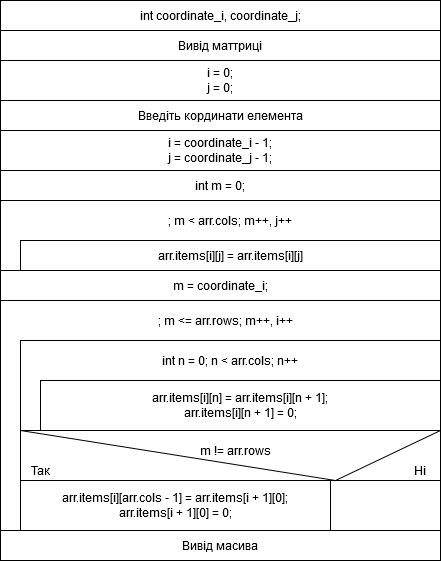
1. Додавання елемента в матрицю



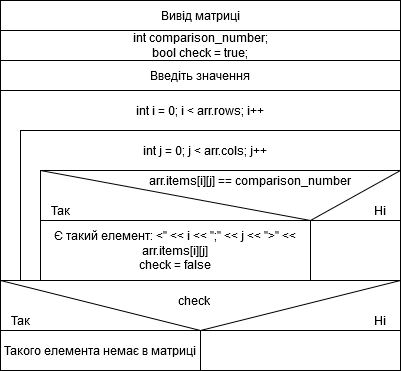
2. Заміна елемента в матрицю



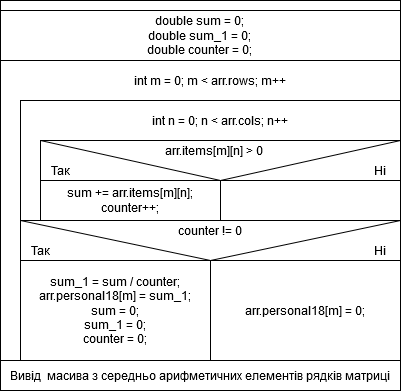
3. Видалення елемента в матриці



4. Пошук елемента матриці



5. Масив з середньо арифметичних елементів рядків матриці



**Текст програми**

#include <iostream>

#include "Windows.h"

#include <ctime>

void main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

struct matrix

{

int rows, cols;

int N;

int\*\* items;

double\* personal18;

};

matrix arr;

std::cout << "Введіть кількість рядків матриці: ";

while (!(std::cin >> arr.rows) || (std::cin.peek() != '\n') || !(arr.rows > 1)) // цикл для перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

system("cls");

std::cerr << "Помилка вводу! Повторіть ввод. Кількість рядків повино бути цілим числом та більше одиниці." << std::endl;

std::cout << std::endl;

std::cout << "Введіть кількість рядків матриці: ";

}

std::cout << "Введіть кількість стовпців матриці: ";

while (!(std::cin >> arr.cols) || (std::cin.peek() != '\n') || !(arr.cols > 1)) // цикл для перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

system("cls");

std::cerr << "Помилка вводу! Повторіть ввод. Кількість стовпців повино бути цілим числом та більше одиниці." << std::endl;

std::cout << std::endl;

std::cout << "Введіть кількість стовпців матриці: ";

}

arr.personal18 = new double [arr.rows];

arr.items = new int \*[arr.rows];

for (int i = 0; i < arr.rows; i++)

{

arr.items[i] = new int[arr.cols];

}

int min\_1 = 0;

int max\_1 = arr.rows;

int min\_2 = 0;

int max\_2 = arr.cols;

int x, size;

size = arr.cols \* arr.rows;

x = (size \* 30) / 100;

srand((unsigned int)time(NULL));

for (int i = 0; i < arr.rows; ++i)

{

for (int j = 0; j < arr.cols; ++j)

{

arr.items[i][j] = 0;

}

}

int\* R;

R = new int [x];

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

{

R[i] = static\_cast <int>((rand()) % static\_cast <int>((max\_1 - min\_1) \* 1)) + min\_1;

}

int y = 0;

int tmp;

for (int i = x - 1; i >= y; i) //сортировка

{

for (int j = x - 1; j >= y; j--)

{

if (R[j] < R[j - 1])

{

tmp = R[j];

R[j] = R[j - 1];

R[j - 1] = tmp;

}

}

y++;

}

int\* C;

C = new int[x];

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

{

C[i] = static\_cast <int>((rand()) % static\_cast <int>((max\_2 - min\_2) \* 1)) + min\_2;

}

int i, j;

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

{

i = R[n];

j = C[n];

arr.items[i][j] = 1 + rand() % 100;

}

int number;

bool exit = true;

int f; //зміна дял вихода з пункта меню

do

{

system("cls");

std::cout << "Розріджена матриця:" << std::endl;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

std::cout << arr.items[i][j] << "\t";

}

std::cout << std::endl;

}

std::cout << std::endl;

std::cout << "1.Додавання елемента матриці" << std::endl;

std::cout << "2.Заміна значення ai,j на інше" << std::endl;

std::cout << "3.Видалення елемента матриці за заданими індексами та по значенню" << std::endl;

std::cout << "4.Пошук елемента матриці за заданими індексами" << std::endl;

std::cout << "5.Одновимірний масив з середньо геометричного ненульових елементів стовбців матриці" << std::endl;

std::cout << "6.Вихід" << std::endl;

std::cin >> number;

std::cin.get();

switch(number)

{

case 1:

{

system("cls");

int coordinate\_i, coordinate\_j, new\_number;

i = 0;

j = 0;

exit = true;

for (int n = 0; exit == true; n++)

{

system("cls");

//std::cout << "Розріджена матриця:" << std::endl;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

std::cout << arr.items[i][j] << "\t";

}

std::cout << std::endl;

}

std::cout << "Введіть кординати елемента: ";

std::cin >> coordinate\_i >> coordinate\_j;

std::cout << "Введіть новий елемент: ";

std::cin >> new\_number;

i = coordinate\_i;

j = coordinate\_j;

arr.items[i - 1][j - 1] = new\_number;

std::cout << std::endl;

std::cout << "Вихід в меню (Так(1) або Ні(0)): ";

std::cin >> f;

if (f == 1)

{

exit = false;

system("cls");

}

}

}

break;

case 2:

{

system("cls");

int coordinate\_i, coordinate\_j, new\_number;

i = 0;

j = 0;

exit = true;

for (int n = 0; exit == true; n++)

{

system("cls");

//std::cout << "Розріджена матриця:" << std::endl;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

std::cout << arr.items[i][j] << "\t";

}

std::cout << std::endl;

}

std::cout << "Введіть кординати елемента: ";

std::cin >> coordinate\_i >> coordinate\_j;

std::cout << "Елемент масиво по цім координатам: " << arr.items[coordinate\_i - 1][coordinate\_j - 1] << std::endl;

std::cout << "Введіть новий елемент: ";

std::cin >> new\_number;

i = coordinate\_i;

j = coordinate\_j;

arr.items[i - 1][j - 1] = new\_number;

//std::cout << std::endl;

std::cout << "Вихід в меню (Так(1) або Ні(0)): ";

std::cin >> f;

if (f == 1)

{

exit = false;

}

}

}

break;

case 3:

{

system("cls");

int coordinate\_i, coordinate\_j;

std::cout << "Масив: " << std::endl;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

std::cout << arr.items[i][j] << "\t";

}

std::cout << std::endl;

}

i = 0;

j = 0;

std::cout << "Введіть кординати елемента: ";

std::cin >> coordinate\_i >> coordinate\_j;

i = coordinate\_i - 1;

j = coordinate\_j - 1;

std::cout << "Елемент масиво по цім координатам: " << arr.items[i][j] << std::endl;

int m = 0;

for (; m < arr.cols; m++, j++)

{

arr.items[i][j] = arr.items[i][j];

}

m = coordinate\_i;

for (; m <= arr.rows; m++, i++)

{

for (int n = 0; n < arr.cols; n++)

{

arr.items[i][n] = arr.items[i][n + 1];

arr.items[i][n + 1] = 0;

}

if (m != arr.rows)

{

arr.items[i][arr.cols - 1] = arr.items[i + 1][0];

arr.items[i + 1][0] = 0;

}

}

arr.items[arr.rows - 1][arr.cols - 1] = 0;

std::cout << "Масив: " << std::endl;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

std::cout << arr.items[i][j] << "\t";

}

std::cout << std::endl;

}

system("pause");

}

break;

case 4:

{

system("cls");

std::cout << "Масив: " << std::endl;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

std::cout << arr.items[i][j] << "\t";

}

std::cout << std::endl;

}

int comparison\_number;

bool check = true;

std::cout << "Введіть значення: ";

std::cin >> comparison\_number;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

if (arr.items[i][j] == comparison\_number)

{

std::cout << "Є такий елемент: <" << i << ";" << j << ">" << arr.items[i][j] << std::endl;

check = false;

}

}

}

if (check)

{

std::cout << "Такого елемента немає в матриці" << std::endl;

}

system("pause");

}

break;

case 5:

{

system("cls");

double sum = 0;

double sum\_1 = 0;

double counter = 0;

for (int m = 0; m < arr.rows; m++)

{

for (int n = 0; n < arr.cols; n++)

{

if (arr.items[m][n] > 0)

{

sum += arr.items[m][n];

counter++;

}

}

if (counter != 0)

{

sum\_1 = sum / counter;

arr.personal18[m] = sum\_1;

sum = 0;

sum\_1 = 0;

counter = 0;

}

else

{

arr.personal18[m] = 0;

}

}

std::cout << "Масив з середньо арифметичних елементів рядків матриці:" << std::endl;

for (int m = 0; m < arr.rows; m++)

{

std::cout << arr.personal18[m] << " ";

}

std::cout << std::endl << std::endl;

system("pause");

}

break;

case 6:

return;

break;

default:

break;

}

}while (number != 6);

return;

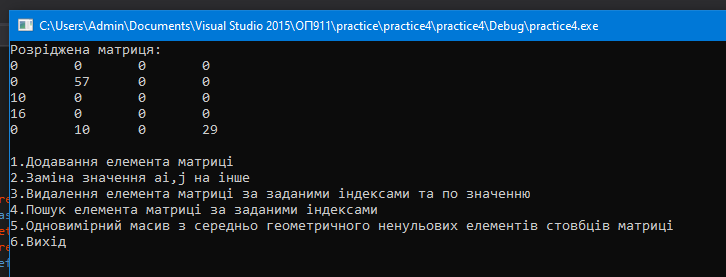
}

**Розробка тестів**

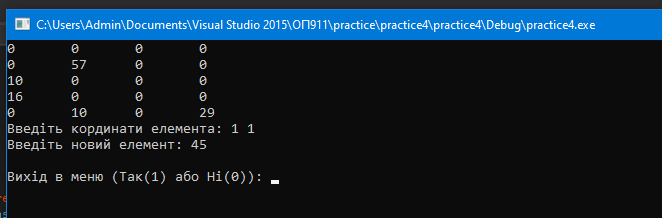
Вхідні дані для тестів:

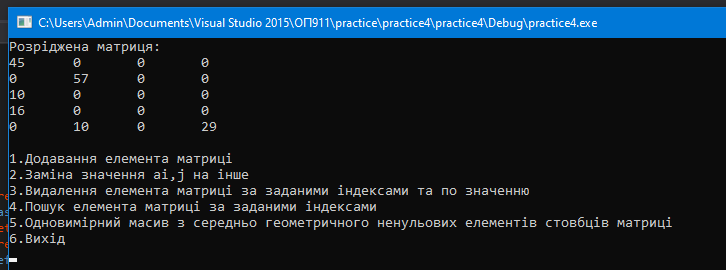
Розмір матриці: 54

Меню

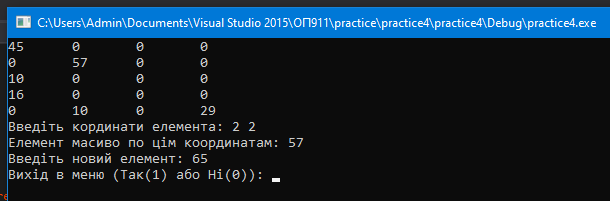


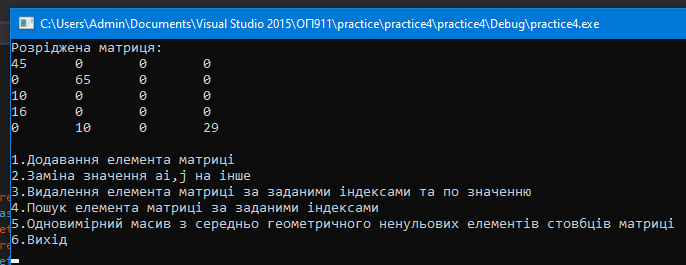
Додавання елемента в матрицю



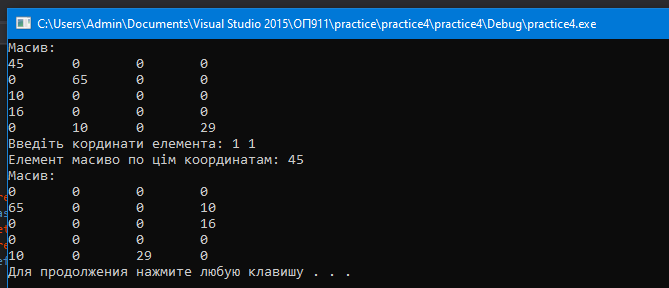


Заміна елемента в матриці

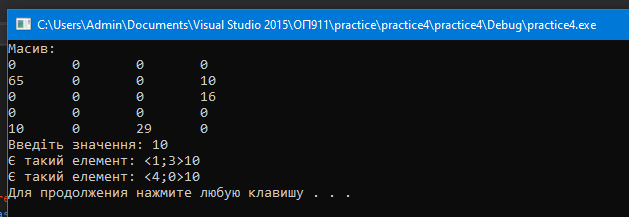




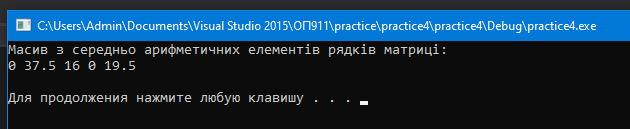
Видалення елемента в матриці



Пошук елемента матриці



Масив з середньо арифметичних елементів рядків матриці



Висновок: Розріджена матриця ­ [матриця](https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), більша частина елементів якої є нулі. Немає єдиного визначення, яка кількість ненульових елементів має бути в матриці, щоб вона була розрідженою.

При зберіганні і перетворенні розріджених матриць в [комп'ютері](http://znaimo.com.ua/Компютер) буває корисно, а часто і необхідно, використовувати спеціальні [алгоритми](http://znaimo.com.ua/Алгоритм) та [структури даних](http://znaimo.com.ua/Структура_даних), які враховують розріджену структуру матриці. Операції та алгоритми, що застосовуються для роботи зі звичайними, щільними матрицями, працюють відносно повільно і вимагають значних обсягів пам'яті, коли застосовуються до великих розріджених матриць. Розріджені дані по природі своїй легко [стискуються](http://znaimo.com.ua/Стиснення_даних), а облік розрідженості часто призводить до зменшення вимог до [комп'ютерної пам'яті](http://znaimo.com.ua/Компютерна%20пам'ять).