Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра «Комп’ютеризовані системи автоматики»



ЗВІТ

про виконання лабораторної роботи №5

«ДВОВИМІРНІ ТА ДИНАМІЧНІ МАСИВИ»

із навчальної дисципліни: «Алгоритми і структури даних»

Варіант №1

Виконав:

студент групи ІР-24

Кіндрат Віктор Романович

Прийняв:

к. ф.-м. н., доцент.

Сиротюк С. В.

Львів 2024

**Мета роботи** – вдосконалення навичок роботи з різними масивами (двовимірними, покажчиків, динамічними) та покажчиками на масиви.

**Програма 5\_1.** Програма обчислює суму значень елементів двовимірного масиву в рядках та стовпцях.

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

int arr[4][3] = {{2, 3, 4}, {7}, {1, 9}, {8, 4, 10}}, s1[4], s2[3];

int main() {

int i, j;

puts("Array arr[4][3]");

for (i = 0; i < 4; i++) {

for (j = 0; j < 3; j++)

printf("%5d", arr[i][j]);

printf("\n");

}

printf("\n Sum elements strings\n");

for (i = 0; i < 4; i++) {

for (j = 0; j < 3; j++)

s1[i] += arr[i][j];

printf("s1[%d]=%d\n", i, s1[i]);

}

printf("\n Sum elements columns\n");

for (j = 0; j < 3; j++) {

for (i = 0; i < 4; i++)

s2[j] += arr[i][j];

printf("s2[%d]=%d\n", j, s2[j]);

}

system("pause");

return 0;

}

**Завдання до програми 5\_1.** Розробіть програму, в якій вводиться із клавіатури і виводиться на екран двовимірний масив цілих чисел заданого розміру та виконуються задані обчислення.

1. Обчисліть суму додатних непарних елементів в стовбцях масиву. Розмір масиву 5 на 4.

**Програмний код:**

#include <stdio.h>  
  
  
int main() {  
 int arr[5][4];  
  
 for (int i = 0; i < 5; i++) {  
 printf("Заповнення рядка %d:\n", i + 1);  
  
 for (int j = 0; j < 4; j++) {  
 printf("Заповніть елемент матриці %d.%d = ", i + 1, j + 1);  
 scanf("%d", &arr[i][j]);  
 }  
 }  
  
 printf("Введана матриця: \n");  
 for (int i = 0; i < 5; i++) {  
 for (int j = 0; j < 4; j++) {  
 printf("%5d ", arr[i][j]);  
 }  
 printf("\n");  
 }  
  
 for (int j = 0; j < 4; j++) {  
 int sum = 0;  
  
 for (int i = 0; i < 5; i++) {  
 if (arr[i][j] > 0 && arr[i][j] % 2 != 0) {  
 sum += arr[i][j];  
 }  
 }  
  
 printf("Сума %d: %d \n", j + 1, sum);  
 }  
  
  
 return 0;  
}

**Приклад виконання:**

viktorkindrat@MacBook-Air-Viktor lab5 % ./1.out

Заповнення рядка 1:

Заповніть елемент матриці 1.1 = 1

Заповніть елемент матриці 1.2 = 2

Заповніть елемент матриці 1.3 = 3

Заповніть елемент матриці 1.4 = 4

Заповнення рядка 2:

Заповніть елемент матриці 2.1 = 5

Заповніть елемент матриці 2.2 = 6

Заповніть елемент матриці 2.3 = 7

Заповніть елемент матриці 2.4 = 8

Заповнення рядка 3:

Заповніть елемент матриці 3.1 = 9

Заповніть елемент матриці 3.2 = 0

Заповніть елемент матриці 3.3 = 11

Заповніть елемент матриці 3.4 = 22

Заповнення рядка 4:

Заповніть елемент матриці 4.1 = 33

Заповніть елемент матриці 4.2 = 44

Заповніть елемент матриці 4.3 = 55

Заповніть елемент матриці 4.4 = 66

Заповнення рядка 5:

Заповніть елемент матриці 5.1 = 77

Заповніть елемент матриці 5.2 = 88

Заповніть елемент матриці 5.3 = 99

Заповніть елемент матриці 5.4 = 100

Введана матриця:

1 2 3 4

5 6 7 8

9 0 11 22

33 44 55 66

77 88 99 100

Сума 1: 125

Сума 2: 0

Сума 3: 175

Сума 4: 0

**Програма 5\_2.** В програмі розглядається можливість доступу до елементів двовимірного масиву різними способами.

#include <stdio.h>

#include <time.h> //для запуску генератора випадкових чисел

#include <stdlib.h> //для підключення генератора випадкових чисел

#include <iostream>

const int N = 4, M = 3;

float A[N][M], sum;

int main() {

srand((unsigned)time(NULL)); // запуск генератора випадк. чисел

int i, j;

puts("Array A:");

for (i = 0; i < N; i++) {

for (j = 0; j < M; j++) {

\*(\*(A + i) + j) = rand() % 1001 \* 0.1f - 50.f; //від –50.0 до 50.0

printf("%8.1f", \*(A[i] + j));

}

printf("\n");

}

for (i = 0; i < N; i++)

for (j = 0; j < M; j++)

if (A[i][j] > 0 && A[i][j] <= 20)

sum += (\*(A + i))[j];

printf("Sum elements from 0 to 20 =%6.1f\n", sum);

system("pause");

return 0;

}

**Завдання до програми 5\_2.** Розробіть програму, в якій двовимірний масив заданого розміру заповнюється випадковими числами у заданому діапазоні і виводиться на екран. Потім виконуються задані обчислення, використовуючи різні способи доступу до елементів масиву.

1. Обчисліть кількість чисел в масиві, що більші за 15.5. Розмір масиву 5 на 4. Діапазон випадкових чисел від 10.50 до 30.50.

**Програмний код:**

#include <stdio.h> #include <stdlib.h> #include <time.h>  
  
#define ROWS 5 #define COLS 4  
  
int main() {  
 float arr[ROWS][COLS];  
 int count = 0;  
 srand((unsigned int)time(NULL));  
  
 for (int i = 0; i < ROWS; i++) {  
 for (int j = 0; j < COLS; j++) {  
 arr[i][j] = 10.50f + (rand() / (float)RAND\_MAX) \* 20.0f;  
 }  
 }  
  
 puts("Array:");  
 for (int i = 0; i < ROWS; i++) {  
 for (int j = 0; j < COLS; j++) {  
 printf("%8.2f ", arr[i][j]);  
 }  
 printf("\n");  
 }  
  
 for (int i = 0; i < ROWS; i++) {  
 for (int j = 0; j < COLS; j++) {  
 if (arr[i][j] > 15.5f) {  
 count++;  
 }  
 }  
 }  
  
 printf("Number of elements greater than 15.5: %d\n", count);  
  
 return 0;  
}

**Приклад виконання:**

viktorkindrat@MacBook-Air-Viktor lab5 % ./2.out

Array:

18.77 26.37 19.75 22.73

13.71 16.59 21.50 29.16

12.48 23.92 30.14 16.42

30.14 25.95 20.09 17.30

28.85 22.50 18.20 22.54

Number of elements greater than 15.5: 18

**Програма 5\_3.** В програмі використовується масив покажчиків.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <iostream>

int main() {

char \*fi[] = {"September", "October", "November"};

printf(" %s\n %s\n %s\n", fi[0], fi[1], fi[2]);

printf("Size massiv pointer=%d\n", sizeof(fi));

printf("Size pointer 1 string=%d\n", sizeof(fi[0]));

printf("Size pointer 2 string=%d\n", sizeof(fi[1]));

printf("Size pointer 3 string=%d\n", sizeof(fi[2]));

puts("\n Massiv pointer string");

for (int i = 0; i < 3; i++)

printf("%s %d\n", fi[i], strlen(fi[i]));

system("pause");

return 0;

}

**Завдання до програми 5\_3.** Розробіть програму, в якій визначається масив покажчиків на три задані рядки. Ці рядки виводяться на екран, визначається їх розмір та розмір масиву покажчиків. Потім виконуються задані обчислення.

1. Визначте масив з 3-х покажчиків на рядки з прізвищем, ім’ям та по батькові. У першому рядку підрахуйте кількість букв 'о'.

**Програмний код:**

#include <stdio.h> #include <string.h>  
  
int main() {  
 char \*fi[] = {"Ivanneko", "Petro", "Iakovych"};  
  
 printf(" %s\n %s\n %s\n", fi[0], fi[1], fi[2]);  
  
 printf("Size of the array of pointers = %lu bytes\n", sizeof(fi));  
 printf("Size of pointer to the 1st string = %lu bytes\n", sizeof(fi[0]));  
 printf("Size of pointer to the 2nd string = %lu bytes\n", sizeof(fi[1]));  
 printf("Size of pointer to the 3rd string = %lu bytes\n", sizeof(fi[2]));  
  
 int count = 0;  
  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < strlen(fi[i]); j++) {  
 if (fi[i][j] == 'o' || fi[i][j] == 'O') {  
 count++;  
 }  
 }  
 }  
  
 printf("Number of 'o's in the strings = %d\n", count);  
  
 return 0;  
}

**Приклад виконання:**

viktorkindrat@MacBook-Air-Viktor lab5 % ./3.out

Ivanneko

Petro

Iakovych

Size of the array of pointers = 24 bytes

Size of pointer to the 1st string = 8 bytes

Size of pointer to the 2nd string = 8 bytes

Size of pointer to the 3rd string = 8 bytes

Number of 'o's in the strings = 3

**Програма 5\_4.** В програмі створюється динамічний масив для змінних типу int. Розмір масиву та значення елементів масиву вводяться з клавіатури. По закінченні роботи з масивом пам’ять звільняється.

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

int main() {

int n, i;

int \*ptr; //визначення покажчика на масив чисел типу int

puts("Input size array");

scanf("%d", &n);

//запит пам’яті

ptr = new int[n]; // ptr = (int\*)malloc(n

\* sizeof(int));

puts("Input elements array");

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

scanf("%d", ptr + i);

puts("Output elements array");

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

printf("%3d", \*(ptr + i));

printf("\n");

delete[] ptr; //вивільнення пам’яті

system("pause");

return 0;

}

**Завдання до програми 5\_4.** Розробіть програму, в якій виконується введення з клавіатури динамічного масиву. Після введення даних виконується сортування масиву за зростанням та виводиться на екран.

1. Виконайте динамічне виділення пам’яті для масиву чисел типу float. Розмір масиву вводиться з клавіатури.
2. Виконайте динамічне виділення пам’яті для масиву символів. Розмір масиву вводиться з клавіатури.
3. Виконайте динамічне виділення пам’яті для масиву рядків. Розмір масиву вводиться з клавіатури.
4. Виконайте динамічне виділення пам’яті для масиву об’єктів класу. Розмір масиву вводиться з клавіатури.
5. Виконайте динамічне виділення пам’яті для масиву структур. Розмір масиву вводиться з клавіатури.

**Програмний код:**

#include <stdio.h> #include <stdlib.h> #include <time.h>  
  
int main() {  
 int n;  
 long \*arr;  
  
 printf("Input size of the array: ");  
 scanf("%d", &n);  
  
 arr = (long\*)malloc(n \* sizeof(long));  
  
 if (arr == NULL) {  
 printf("Error! Memory not allocated.\n");  
 return -1;  
 }  
  
 srand((unsigned)time(NULL));  
  
 for (int i = 0; i < n; i++) {  
 arr[i] = rand() % 10001 - 5000; // від -5000 до +5000  
 }  
  
 printf("\nGenerated array:\n");  
 for (int i = 0; i < n; i++) {  
 printf("%ld ", arr[i]);  
 }  
 printf("\n");  
  
 free(arr);  
  
 return 0;

}

**Приклад виконання:**

viktorkindrat@MacBook-Air-Viktor lab5 % ./4.out

Input size of the array: 3

Generated array:

2258 4620 118

**Програма 5\_5**

В програмі створюються двовимірні динамічні квадратні матриці. Розмір матриць вводиться з клавіатури. Значення елементів першої матриці вводяться з клавіатури, а другої – заповнюються випадковими числами від -50 до +50. Потім в другу матрицю заносяться суми елементів обох матриць.

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main() {

int i, j, N;

printf("Input the size of the square matrix: ");

scanf("%u", &N);

int \*\*matr; // покажчик на масив покажчиків

matr = new int\*[N]; // запит пам’яті для масиву покажчиків

if (matr == NULL) {

puts("Error!");

return -1;

}

for (i = 0; i < N; i++) {

matr[i] = new int[N]; // запит пам’яті для елементів рядків

if (matr[i] == NULL) {

puts("Error!");

return -2;

}

}

puts("\nInput the elements of the matrix matr:");

for (i = 0; i < N; i++)

for (j = 0; j < N; j++)

scanf("%d", &matr[i][j]);

puts("\n\tMatrix matr:");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("%8d", matr[i][j]);

}

printf("\n");

}

int \*\*mas; // Покажчик на масив покажчиків

mas = new int\*[N]; // Запит пам’яті для масиву покажчиків

if (mas == NULL) {

puts("Error!");

return -1;

}

for (i = 0; i < N; i++) {

mas[i] = new int[N]; // Запит пам’яті для елементів рядків

if (mas[i] == NULL) {

puts("Error!");

return -2;

}

}

srand((unsigned)time(NULL)); // запуск генератора випадк. чисел

puts("\n\tMatrix mas:");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

mas[i][j] = rand() % 101 - 50;

printf("%8d", mas[i][j]);

}

printf("\n");

}

puts("\nNew values of elements of matrix mas:");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

mas[i][j] = matr[i][j] + mas[i][j];

printf("%8d", mas[i][j]);

}

printf("\n");

}

// Звільнення пам’яті

for (i = 0; i < N; i++) {

delete[] matr[i]; // звільнення пам’яті з-під масивів елементів

delete[] mas[i]; // звільнення пам’яті з-під масивів елементів

}

delete[] matr; // звільнення пам’яті з-під масиву покажчиків

delete[] mas; // звільнення пам’яті з-під масиву покажчиків

system("pause");

return 0;

}

**Завдання до програми 5\_5**

Доповніть програму, створивши третю динамічну матрицю. Виконайте задані відповідно варіанту дії. Результат занесіть в третю матрицю та виведіть на екран.

1. Порівняйте значення елементів перших двох матриць, найбільше значення занесіть в третю матрицю.

**Програмний код:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main() {

int i, j, N;

// Введення розміру матриці

printf("Input the size of the square matrix: ");

scanf("%d", &N);

// Виділення пам'яті для матриці matr

int \*\*matr = (int\*\*)malloc(N \* sizeof(int\*));

if (matr == NULL) {

puts("Error! Memory not allocated.");

return -1;

}

for (i = 0; i < N; i++) {

matr[i] = (int\*)malloc(N \* sizeof(int));

if (matr[i] == NULL) {

puts("Error! Memory not allocated.");

return -2;

}

}

// Введення елементів матриці matr

puts("\nInput the elements of the matrix matr:");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

scanf("%d", &matr[i][j]);

}

}

// Виведення матриці matr

puts("\n\tMatrix matr:");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("%8d", matr[i][j]);

}

printf("\n");

}

// Виділення пам'яті для матриці mas

int \*\*mas = (int\*\*)malloc(N \* sizeof(int\*));

if (mas == NULL) {

puts("Error! Memory not allocated.");

return -1;

}

for (i = 0; i < N; i++) {

mas[i] = (int\*)malloc(N \* sizeof(int));

if (mas[i] == NULL) {

puts("Error! Memory not allocated.");

return -2;

}

}

// Ініціалізація генератора випадкових чисел

srand((unsigned)time(NULL));

// Заповнення масиву mas випадковими числами від -50 до 50

puts("\n\tMatrix mas:");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

mas[i][j] = rand() % 101 - 50; // випадкове число від -50 до 50

printf("%8d", mas[i][j]);

}

printf("\n");

}

// Виділення пам'яті для третьої матриці maxMat

int \*\*maxMat = (int\*\*)malloc(N \* sizeof(int\*));

if (maxMat == NULL) {

puts("Error! Memory not allocated.");

return -1;

}

for (i = 0; i < N; i++) {

maxMat[i] = (int\*)malloc(N \* sizeof(int));

if (maxMat[i] == NULL) {

puts("Error! Memory not allocated.");

return -2;

}

}

// Порівняння елементів перших двох матриць і занесення найбільшого у maxMat

puts("\nNew matrix maxMat (largest values from matr and mas):");

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

if (matr[i][j] > mas[i][j]) {

maxMat[i][j] = matr[i][j];

} else {

maxMat[i][j] = mas[i][j];

}

printf("%8d", maxMat[i][j]);

}

printf("\n");

}

// Звільнення пам'яті

for (i = 0; i < N; i++) {

free(matr[i]); // звільнення пам'яті з-під рядків matr

free(mas[i]); // звільнення пам'яті з-під рядків mas

free(maxMat[i]); // звільнення пам'яті з-під рядків maxMat

}

free(matr); // звільнення пам'яті з-під масиву покажчиків matr

free(mas); // звільнення пам'яті з-під масиву покажчиків mas

free(maxMat); // звільнення пам'яті з-під масиву покажчиків maxMat

return 0;

}

**Приклад виконання:**

Input the size of the square matrix: 3

Input the elements of the matrix matr:

1

2

3

4

5

6

7

8

9

Matrix matr:

1 2 3

4 5 6

7 8 9

Matrix mas:

43 -14 17

0 34 17

32 12 -17

New matrix maxMat (largest values from matr and mas):

43 2 17

4 34 17

32 12 9

**Висновок:**

Під час виконання лабораторної роботи, я вдосконалив навички роботи з різними масивами (двовимірними, покажчиків, динамічними) та покажчиками на масиви.