**Практична робота 6.**

**БАГАТОВИМІРНІ ТА ДИНАМІЧНІ МАСИВИ**

Мета Навчитись використовувати вказівники при роботі з багатовимірними та динамічними масивами. Навчитись розв’язувати типові задачі з використанням багатовимірних та динамічних масивів.

Приклад 1.

#include <stdio.h>

//опис цілочислових констант

const M = 3; //кількість рядків двовимірного масиву

const N = 4; //кількість стовпців двовимірного масиву

void main(void)

{

//опис двовимірного масиву цілих чисел

int arr[M][N];

//введення елементів двовимірного масиву

int i, j;

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

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

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

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

}

//порядкове виведення елементів двовимірного масиву

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

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

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

printf("\n");

}

int min; //значення мінімального

int max; //значення максимального

int i\_min, j\_min; //індекси мінімального

int i\_max, j\_max; //індекси максимального

min = max = arr[0];

i\_min = j\_min = i\_max = j\_max = 0;

//пошук значень й індексів мінімального й максимального елементів

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

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

if (arr[i][j] > max) {

max = arr[i][j];

i\_max = i;

j\_max = j;

}

if (arr[i][j] < min) {

min = arr[i][j];

i\_min = i;

j\_min = j;

}

}

//заміна місцями максимального й мінімального елементів

int d; //додаткова змінна

d = arr[i\_min][j\_min];

arr[i\_min][j\_min] = arr[i\_max][j\_max];

arr[i\_max][j\_max] = d;

//порядкове виведення елементів двовимірного масиву написати

самостійно(див.вище) ...

}

Приклад 2.

#include <iostream.h>

#include <conio.h>

int main()

{

const int n = 5, m = 5;

int a[n][m]; int i, j;

//Введення масиву a

cout << "Input an array A\n";

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

{

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

{

cout << "a[" << i << "][" << j << "]=";

cin >> a[i][j];

}

}

//Виведення масиву А у вигляді матриці

cout << "Array a\n";

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

{

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

{

cout << a[i][j] << "\t";

}

cout << "\n";

}

//Виведення заштрихованої області масиву на екран

cout << "Array in the shaded area:\n";

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

{

//вверхня половина заштрихованої області

for (j = 0; (j <= i) && (i <= n / 2); j++)

{

cout << a[i][j] << "\t";

}

//нижня половина заштрихованої області

for (j = 0; (j < m - i) && (i > n / 2); j++)

{

cout << a[i][j] << "\t";

}

cout << "\n";

}

system("pause");

return 0;

}

Приклад 3.

#include <stdio.h>

#include <stdlib.h> //тут знаходяться функції calloc(), free();

void main(void)

{

int n; //кількість елементів масиву

printf("n=");

scanf("%d", &n);

char\* data; //вказівник на тип char

//виділення динамічної пам'яті під масив символів з n комірок

//розмір кожної комірки визначається операцією sizeof (char)

data = (char\*)calloc(n, sizeof(char));

//введення елементів масиву

int i;

for (i = 0; i < 10; i++) {

flushall(); // вивільнення всіх буферів, пов'язаних з

//вивідними потоками (не обов’язково,

//але для надійності).

scanf("%c", &data[i]);

}

//заміну символів, відповідно до умови реалізувати самостійно ...

//виведення в один рядок елементів масиву реалізувати самостійно

free(data); //вивільнення динамічної пам'яті

}

Приклад 4.

#include <stdio.h>

#include <stdlib.h>

void main(void)

{

int m; //кількість рядків у масиві

int n; //кількість стовпців у масиві

//введення m й n а також перевірку коректності введених

//значень написати самостійно ...

float\*\* data; //подвійний вказівник на дійсний тип

//виділення динамічної пам'яті під двовимірний масив дійсних чисел

data = (float\*\*)calloc(m, sizeof(float\*));

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

data[i] = (float\*)calloc(n, sizeof(float));

//введення елементів масиву, пошук суми його ненегативних елементів

//і виведення результату на екран написати самостійно ...

//вивільнення пам'яті динамічного масиву

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

free(data[i]);

free(data);

}

Приклад 5.

#include <stdio.h>

void main(void)

{

//опис змінної кількості елементів масиву,

//й введення її значення написати самостійно ...

//виділення пам'яті під лінійний масив цілих чисел у стилі С++

data = new int[n];

//введення елементів масиву описати самостійно ...

//впорядкування

int i, k, d;

for (k = 0; k < n - 1; k++)

for (i = k + 1; i < n; i++)

if (data[i] > data[i + 1]) {

d = data[i];

data[i] = data[i + 1];

data[i + 1] = d;

}

//виведення результату описати самостійно ...

//звільнення пам'яті динамічного масиву в стилі С++

delete[] data;

}

Приклад 6.

ф#include <iostream>

#include <ctime>

#include <math.h>

#include <cmath>

#include <string>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int const ROWS = 5;

int const COLS = 4;

int arr[ROWS][COLS];

int indexResult = 0;

cout << "========================================================" << endl;

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

{

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

{

arr[i][j] = (rand() % 200) - 100;

cout << arr[i][j] << "\t";

if (arr[i][j] < 0)

{

indexResult += arr[i][j];

}

}

cout << endl;

}

cout << "========================================================" << endl;

cout << "Сума від'ємних елементів масиву = " << indexResult << endl;

cout << "========================================================" << endl;

}