**ЛАБОРАТОРНА РОБОТА № 17**

**Варіант 2**

Робота з двовимірними масивами

***Мета*** : набуття практичних навичок роботи з двовимірними масивами.

**Хід роботи:**

Завдання 1: Дана квадратна матриця.

Розмістити елементи непарних рядків у порядку зростання.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include <ctime>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i, j, z[500][500], n, fl, c;

printf("n = ");

scanf\_s("%d", &n);

printf("\n");

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

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

z[i][j] = 0 + rand() % 51;

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

}

printf("\n");

}

do

{

fl = 0;

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

if (!(i % 2))

continue;

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

if (z[i][j - 1] > z[i][j])

{

c = z[i][j];

z[i][j] = z[i][j - 1];

z[i][j - 1] = c;

fl = 1;

}

}

}

} while (fl);

printf("\n");

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

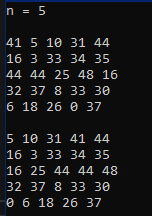
for (j = 0; j < n; j++)

printf("%i%c", z[i][j], (j != n - 1) ? ' ' : '\n');

}

return 0;

}



Завдання 2: Дана прямокутна матриця.

Циклічно зсунути парні стовпці матриці зліва направо на k позицій.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include <ctime>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i, j, z[500][500], n, m, k, x, c;

printf("n = ");

scanf\_s("%d", &n);

printf("m = ");

scanf\_s("%d", &m);

printf("k = ");

scanf\_s("%d", &k);

printf("\n");

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

for (j = 1; j <= m; j++) {

z[i][j] = 10 + rand() % 51;

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

}

printf("\n");

}

c = k;

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

for (j = 1; j <= m; j++) {

if (j % 2)

continue;

for (k = 0; c > k; k++) {

if (j + 1 + k > m)

break;

x = z[i][j + k];

z[i][j + k] = z[i][j + 1 + k];

z[i][j + 1 + k] = x;

}

}

}

printf("\n");

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

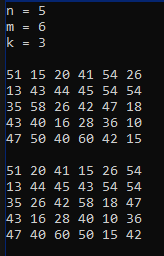
for (j = 1; j <= m; j++)

printf("%i%c", z[i][j], (j != m) ? ' ' : '\n');

}

return 0;

}



Завдання 3: Здійснити сортування двовимірного масиву з наступними параметрами:



1)Метод вибору

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include <ctime>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

unsigned int i, j, z[500][500], n, fl, c;

printf("n = ");

scanf\_s("%d", &n);

printf("\n");

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

for (j = 1; j <= n; j++) {

z[i][j] = 10 + rand() % 51;

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

}

printf("\n");

}

do

{

fl = 0;

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

for (j = 1; j <= n; j++) {

if (z[i][j - 1] > z[i][j])

{

c = z[i][j];

z[i][j] = z[i][j - 1];

z[i][j - 1] = c;

fl = 1;

}

}

}

} while (fl);

printf("\n");

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

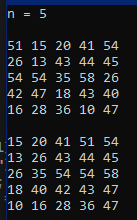
for (j = 1; j <= n; j++)

printf("%i%c", z[i][j], (j != n) ? ' ' : '\n');

}

return 0;

}



1)Метод вставкі

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include <ctime>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

unsigned int i, j, z[500][500], n, c;

printf("n = ");

scanf\_s("%d", &n);

printf("\n");

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

for (j = 1; j <= n; j++) {

z[i][j] = 10 + rand() % 51;

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

}

printf("\n");

}

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

{

for (j = 1; j <= n; j++) {

c = z[i][j];

for (unsigned int x = j - 1; x >= 0 && z[i][x] > c; x--)

{

z[i][x + 1] = z[i][x];

z[i][x] = c;

}

}

}

printf("\n");

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

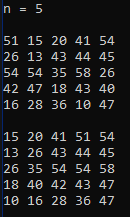
for (j = 1; j <= n; j++)

printf("%i%c", z[i][j], (j != n) ? ' ' : '\n');

}

return 0;

}



Завдання 4: Встановіть відповідність:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Масиви можуть бути: | локальні | void main ()  {  int A [5] [5]; ...  } |
| глобальні | int A [10];  void main () {  ...  } |
| 2 | Масиви бувають | двовимірні | int A [1] [5]; |
| int A [3] [5]; |
| багатовимірні | int A [3] [4] [5]; |
| int A [3] [4] [5] [6]; |
| одновимірний | int A [1]; |
| int A [5]; |
| 4 | У разі … масиви має наступний вигляд | повної ініціалізації | inta[5][3] = { {4, 7, 8}, {9, 66, -1},{5, - 5, 0},{3, -3, 30},{1, 1, 1} }; |
| int mas [ ][5] ={ { 1, 5, 3, 7, 4 },{10, 11, 13, 14, 25} }; |
| int mas [2][5] ={ 1, 5, 3, 7, 4,10, 11, ІЗ, 14, 25 }; |
| int A [4] = {1, 4, 2, 6}; |
| int Mass[3][2][4]={{{l,2,3,4},{5,6,7,8}}, {{9,10,ll,12},{13,14,15,16}}, {{17,18,19,20},{21,22,23,24}}; |
| char x[][3]={{9,8,7},{6,5,4},{3,2,1}}; |
| int A [2] [4] = {1, 4, 4, 7, 2, 1, 4, 4, 2}; |
| int A [] = {1, 4, 2, 6}; |
| int mas [ ][5] ={ 1, 5, 3, 7, 4, 10, 11, 13, 14, 25 }; |
| часткової ініціалізації | int z[6][3][2][5][8][4] |
| int A [4] = {1, 4}; |
| int A [] = {0}; |
| int A [4] [4] = {{2, 6}, {1, 4}, {6} }; |

***Висновки:*** я набув практичних навичок роботи з двовимірними масивами.