

# **НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

## **ЛАБОРАТОРНА РОБОТА №6**

### **Масиви. Визначення масивів**

Виконала:  
студентка групи ІПЗ-210076  
Соколовська Софія

Київ – 2021

## **ВАРІАНТ – 18**

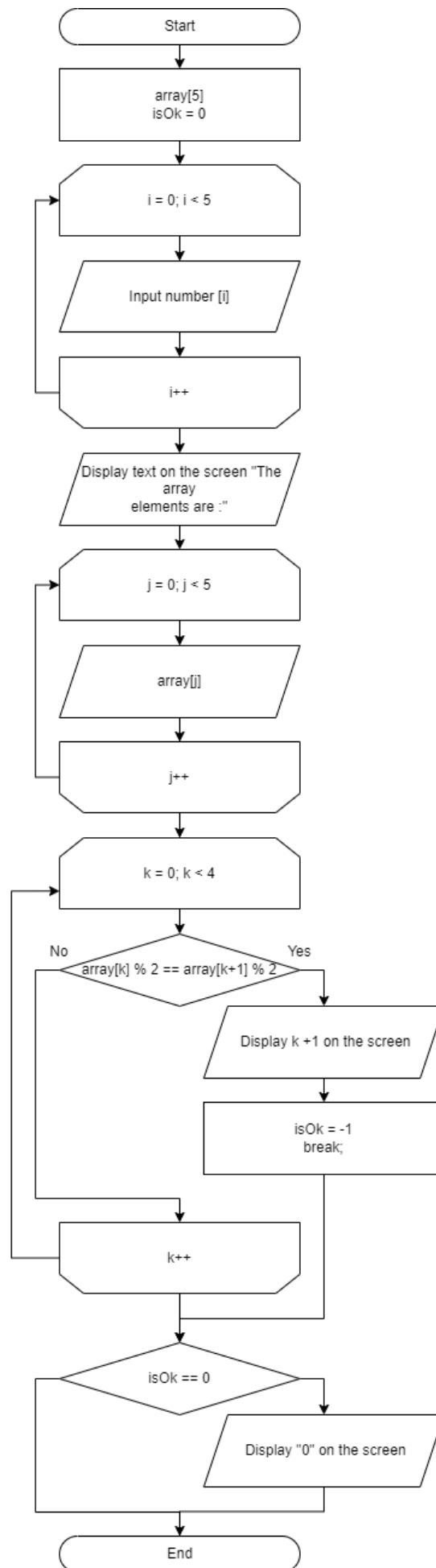
**Тема:** Масиви. Визначення масивів.

**Мета:** Опанування практичними навичками розробки програм із використанням масивів. Розробка стандартних алгоритмів із використанням масивів.

### **Завдання 1**

Використати одномірний масив цілих чисел, який заповнити довільними цілими числами. Перевірити, чи чергуються в ньому парні і непарні числа. Якщо чергуються, то вивести 0, якщо ні, то вивести порядковий номер першого елемента, що порушує закономірність.

- 1) На рис. 1.1 наведена блок-схема алгоритму програми, яка перевіряє чергування парних та непарних чисел.



## 2) Код програми:

```
#include <stdio.h>

#include <stdlib.h>

int main()

{

    int array [5] = { };

    int isOk = 0;

    for(int i = 0; i < 5; i++) {

        printf("Input number[%i]\n", i);

        scanf("%i", &array[i]);

    }

    printf("The array elements are : \n");

    for(int j = 0; j < 5; j++) {

        printf("%i\t", array[j]);

    }

    for(int k = 0; k < 4; k++){

        if(array[k] % 2 == array[k+1] % 2) {

            printf("\n%i", k + 1);

            isOk = -1;

            break;

        }

    }

    if (isOk == 0) {

        printf("\n0");

    }

}
```

```
}  
  
}
```

3) На рис 1.2 наведений результат програми.

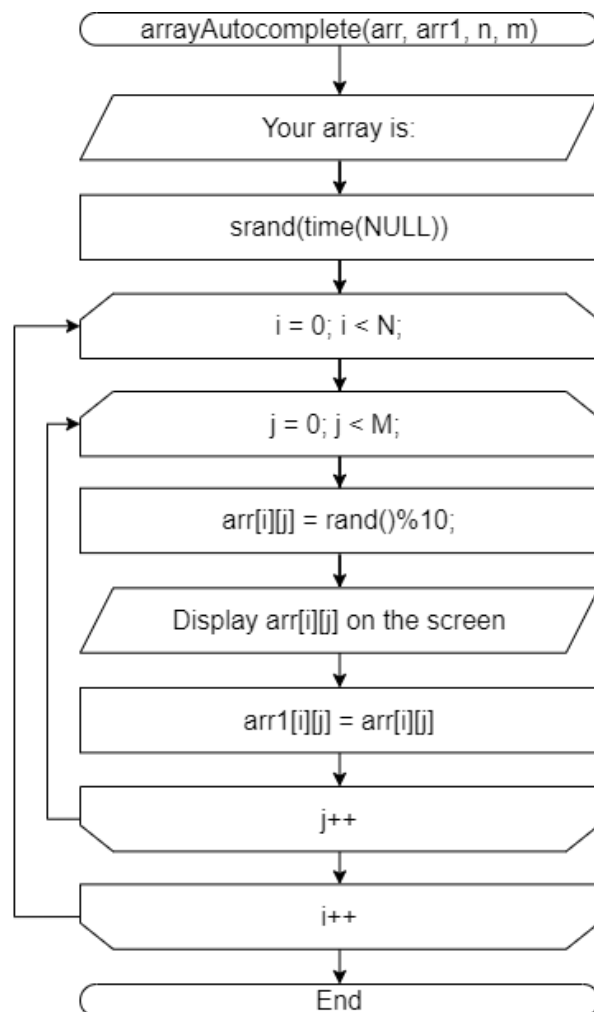
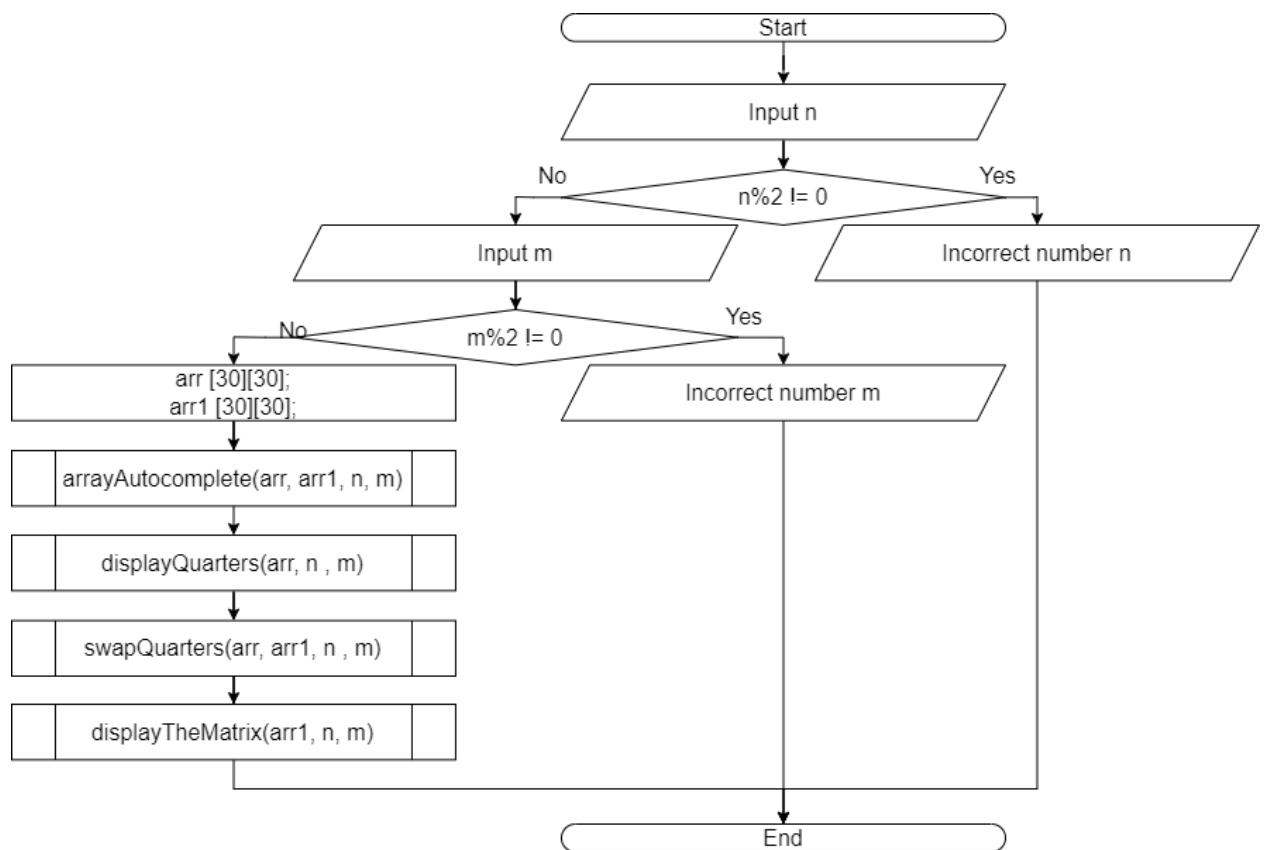
```
Input number[0]  
2  
Input number[1]  
1  
Input number[2]  
4  
Input number[3]  
3  
Input number[4]  
6  
The array elements are :  
2      1      4      3      6  
0
```

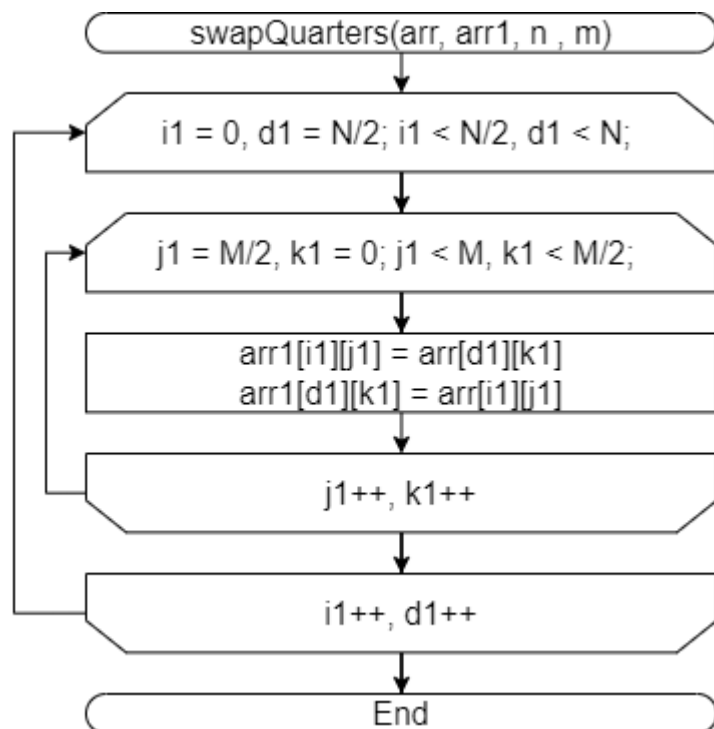
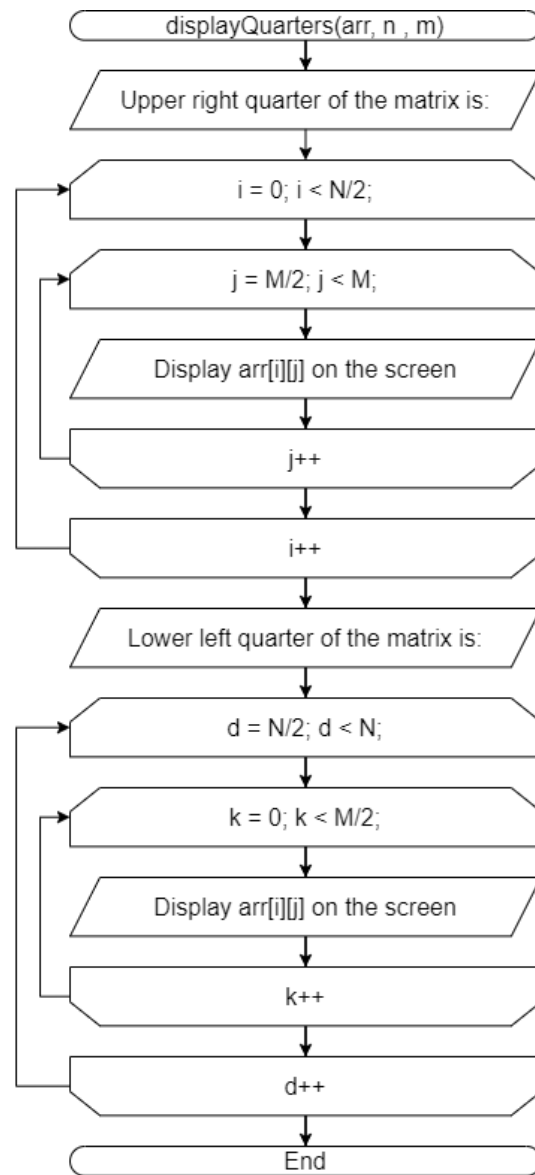
```
Input number[0]  
3  
Input number[1]  
6  
Input number[2]  
5  
Input number[3]  
5  
Input number[4]  
2  
The array elements are :  
3      6      5      5      2  
3
```

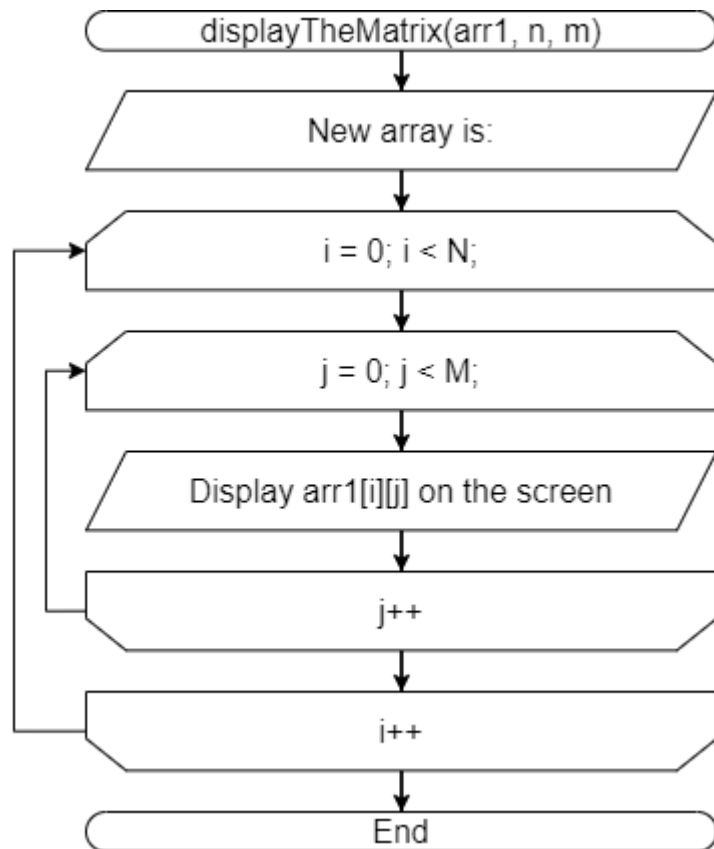
## Завдання 2

Реалізувати алгоритм перетворення матриці розміру  $n \times m$ , змінюючи місцями ліву нижню і праву верхню **чверті** матриці. Значення  $n$  і  $m$  визначаються користувачем ( $n$  – парне число).

- 1) На рис. 2.1 наведена блок-схема алгоритму програми, яка змінює місцями ліву нижню і праву верхню чверті матриці.







2) Код програми:

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

    //Enter the size of the matrix

    int n;

    int m;

    printf("Input n\n");

    scanf("%i", &n);

    if(n%2 != 0) {

        printf("Incorrect number n");
```



```

    } else {

        printf("Input m\n");

        scanf("%i", &m);

        if(m%2 != 0) {

            printf("Incorrect number m");

        } else {

            int arr [30][30];

            int arr1 [30][30];

            arrayAutocomplete(arr, arr1, n, m);

            displayQuarters(arr, n , m);

            swapQuarters(arr, arr1, n , m);

            displayTheMatrix(arr1, n, m);

        }

    }

}

```

```

void arrayAutocomplete(int arr[30][30], int arr1[30][30], int N, int M)
{
    //Array autocomplete

    printf("Your array is:\n");

    srand(time(NULL));

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

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

            arr[i][j] = rand()%10;

```

```

        printf ("%4i",arr[i][j]);

        arr1[i][j] = arr[i][j];

    }

    printf("\n");

}

printf("\n");

}

```

```

void displayQuarters(int arr[30][30], int N, int M)

```

```

{
    //Upper right quarter of the matrix

    printf("Upper right quarter of the matrix is:\n");
    for(int i = 0; i < N/2; i++) {
        for(int j = M/2; j < M; j++) {
            printf("%4i", arr[i][j]);

        }

        printf("\n");
    }

    printf("\n");

    //Lower left quarter of the matrix

    printf("Lower left quarter of the matrix is:\n");
    for(int d = N/2; d < N; d++) {
        for(int k = 0; k < M/2; k++) {
            printf("%4i", arr[d][k]);

```

```

    }

    printf("\n");

}

}

void swapQuarters(int arr[30][30], int arr1[30][30], int N, int M)
{
    //Swap quarters of the matrix

    int i1, j1, d1, k1;

    for(i1 = 0, d1 = N/2; i1 < N/2, d1 < N; i1++, d1++) {
        for(j1 = M/2, k1 = 0; j1 < M, k1 < M/2; j1++, k1++) {
            arr1[i1][j1] = arr[d1][k1];

            arr1[d1][k1] = arr[i1][j1];
        }

        printf("\n");
    }
}

```

```

void displayTheMatrix(int arr1[30][30], int N, int M)
{
    //Display the new matrix

    printf("New array is:\n");

    for(int i = 0; i < N; i++) {
        for(int j = 0; j < M; j++) {

```

```

        printf("%4i", arr1[i][j]);

    }

    printf("\n");

}

}

```

3) На рис 2.2 наведений результат програми.

```

Input n
2
Input m
2
Your array is:
  6  9
  3  1

Upper right quarter of the matrix is:
  9

Lower left quarter of the matrix is:
  3

New array is:
  6  3
  9  1

Process returned 0 (0x0)   execution time : 1.270 s
Press any key to continue.

```

```

Input n
4
Input m
4
Your array is:
  5  0  8  5
  8  3  3  0
  8  2  3  5
  2  0  5  5

Upper right quarter of the matrix is:
  8  5
  3  0

Lower left quarter of the matrix is:
  8  2
  2  0

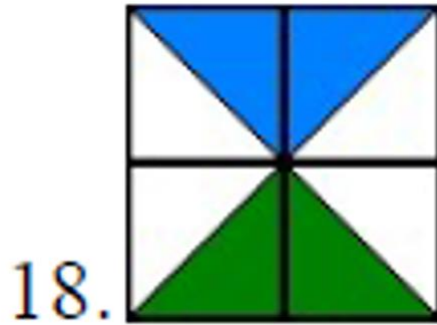
New array is:
  5  0  8  2
  8  3  2  0
  8  5  3  5
  3  0  5  5

Process returned 0 (0x0)   execution time : 0.945 s
Press any key to continue.

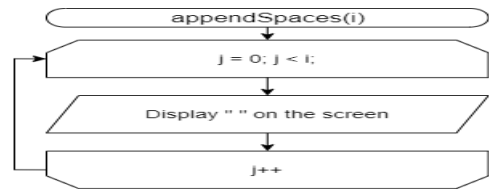
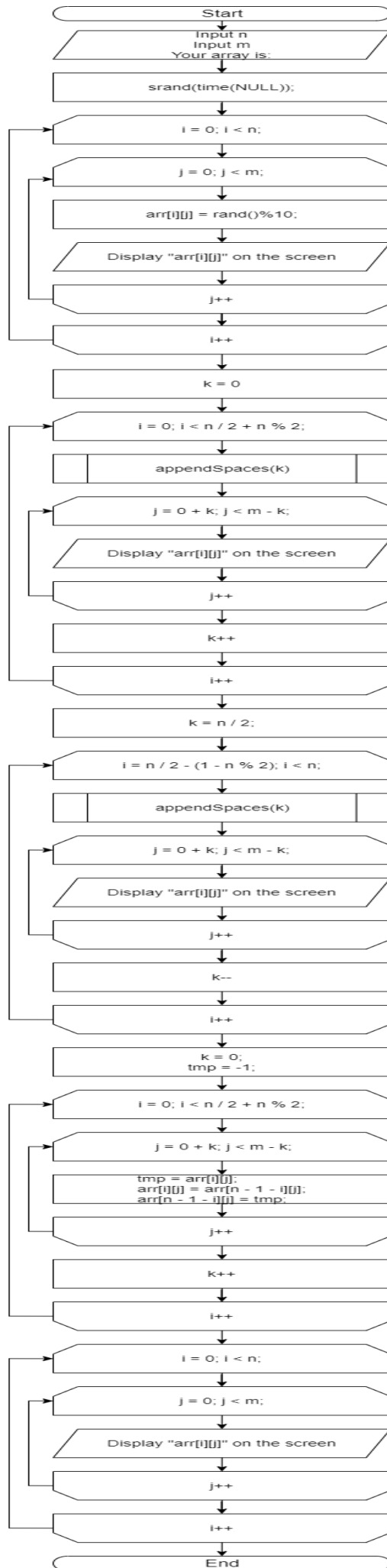
```

### Додаткове завдання

Задана матриця цілих чисел розміром  $(N;N)$ . Поміняти місцями дві зафарбовані області. Заборонено використання додаткових масивів.



- 1) На рис. 3.1 наведена блок-схема алгоритму програми, яка перевіряє чергування парних та непарних чисел.



## 2) Код програми:

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

    //Enter the size of the matrix

    int n;

    int m;

    printf("Input n\n");

    scanf("%i", &n);

    printf("Input m\n");

    scanf("%i", &m);

    int arr[n][m];


    printf("Your array is:\n");

    srand(time(NULL));

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

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

            arr[i][j] = rand()%10;

            printf ("%4i", arr[i][j]);

        }

        printf("\n");

    }
```

```

printf("\n");

//Just a simple function. Appends empty spaces to print triangles.

void appendSpaces(int i) {
    for (int j = 0; j < i; j++) {
        printf ("%4s", "");
    }
}

//Print top triangle

int k = 0;

for (int i = 0; i < n / 2 + n % 2; i++) {
    appendSpaces(k);
    for (int j = 0 + k; j < m - k; j++) {
        printf ("%4i", arr[i][j]);
    }
    printf("\n");
    k++;
}

printf("\n");

//Print bottom triangle

k = n / 2;

for (int i = n / 2 - (1 - n % 2); i < n; i++) {
    appendSpaces(k);
    for (int j = 0 + k; j < m - k; j++) {
        printf ("%4i", arr[i][j]);
    }
}

```



```

    }

    printf("\n");

    k--;

}

printf("\n");

//Swap triangles

k = 0;

int tmp = -1;

for (int i = 0; i < n / 2 + n % 2; i++) {

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

        tmp = arr[i][j];

        arr[i][j] = arr[n - 1 - i][j];

        arr[n - 1 - i][j] = tmp;

    }

    k++;

}

//Print result

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

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

        printf ("%4i", arr[i][j]);

    }

    printf("\n");

}

}

```

3) На рис 3.2 наведений результат програми.

```

Input n
3
Input m
3
Your array is:
  7  1  0
  7  0  9
  5  1  5

  7  1  0
    0

    0
  5  1  5

  5  1  5
  7  0  9
  7  1  0

Process returned 0 (0x0)   execution time : 1.704 s
Press any key to continue.

```

```

6
Input m
6
Your array is:
  7  3  8  8  3  1
  3  9  5  9  3  4
  6  2  7  2  7  6
  1  5  0  2  4  4
  6  7  9  6  6  1
  9  2  8  5  0  5

  7  3  8  8  3  1
    9  5  9  3
      7  2

      0  2
    7  9  6  6
  9  2  8  5  0  5

  9  2  8  5  0  5
  3  7  9  6  6  4
  6  2  0  2  7  6
  1  5  7  2  4  4
  6  9  5  9  3  1
  7  3  8  8  3  1

Process returned 0 (0x0)   execution time : 2.767 s
Press any key to continue.

```

**Висновок:** В ході лабораторної роботи я опанувала практичними навичками розробки програм із використанням масивів. Розробила стандартні алгоритми із використанням масивів.