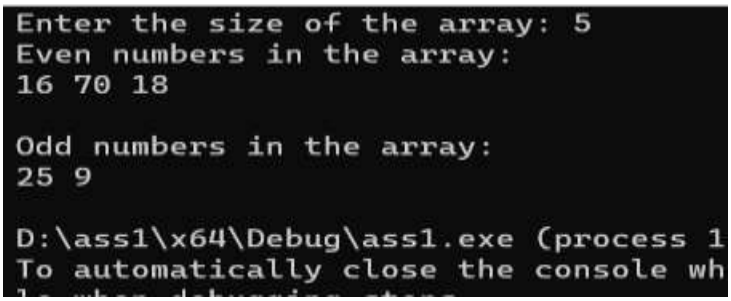
➢ Ques 1:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

void fillArray(int* arr, int size) {
    for (int i = 0; i < size; i++) {
        arr[i] = rand() % 100 + 1;
    }
}
void findEvenOdd(int* arr, int size, int** evenPtr, int** oddPtr) {
    int evenCount = 0;
    int oddCount = 0;

    for (int i = 0; i < size; i++) {
        if (arr[i] % 2 == 0) {
            evenPtr[evenCount++] = &arr[i];
        }
        else {
            oddPtr[oddCount++] = &arr[i];
        }
    }
     Null-terminate the arrays of pointers
    evenPtr[evenCount] = nullptr;
    oddPtr[oddCount] = nullptr;
}

void display(int** x, int size) {
    for (int i = 0; x[i] != nullptr; i++) {
        cout << *x[i] << " ";
    }
    cout << endl;
}
int main() {
    // Seed the random number generator
    srand(time(nullptr));

    int size;
    cout << "Enter the size of the array: ";
    cin >> size;

    int* arr = new int[size];
    int** evenPtr = new int* [size];
    int** oddPtr = new int* [size];

     Fill the array with random numbers
    fillArray(arr, size);

    findEvenOdd(arr, size, evenPtr, oddPtr);

    cout << "Even numbers in the array:\n";
    display(evenPtr, size);

    cout << "\nOdd numbers in the array:\n";
    display(oddPtr, size);
```

```
    delete[] arr;
    delete[] evenPtr;
    delete[] oddPtr;

    return 0;
}
```



➢ Ques 2:

```
#include <iostream>
#include <cstring>
using namespace std;

int numOfOccurences(const char* sentence, const char* find) {

    int count = 0;

    for (int i = 0; sentence[i] != '0'; i++) {

        char ch = find[i];

        if (sentence[i] == find[0]) {
            bool cheak = true;

            // Loop through each character in the word
            for (int j = 0; find[j] != '\0'; j++) {
                // If the characters don't match
                if (sentence[i + j] != find[j]) {
                    cheak = false;
                    break;
                }
            }
            if (cheak) {
                count++;
            }
        }
    }

    return count;
}
int main() {
    const int MAX_SIZE = 100;
    char sentence[MAX_SIZE];
    char find[MAX_SIZE];
```

```
    cout << "Enter the sentence: ";
    cin.getline(sentence, MAX_SIZE);

    cout << "Enter the word to find: ";
    cin.getline(find, MAX_SIZE);

    int occurrences = numOfOccurences(sentence, find);
    cout << "Number of occurrences: " << occurrences << endl;
    system("pause")
```

```
Enter the sentence: “Posing a possible post of possibilities”
Enter the word to find: pos
Number of occurrences: 4
Press any key to continue . . .
```

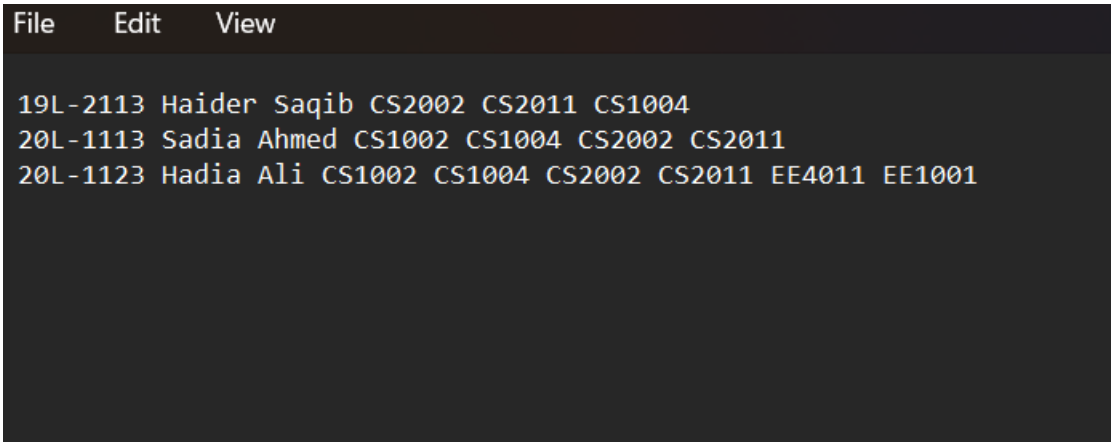
- ## Ques 3

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

struct Student {
    string rollNumber;
    string name;
    string courseCodes[5];
    int numCourses;
};
void loadDataFromFile(ifstream & inputFile, Student& student) {

    if (inputFile.is_open()) {
        inputFile >> student.rollNumber;

        getline(inputFile, student.name);

        student.numCourses = 0;
        string courseCode;

        while (inputFile >> courseCode && student.numCourses < 5) {
            student.courseCodes[student.numCourses++] = courseCode;
        }

        inputFile.close();
    }
    else {
        cout << "Unable to open RollNumber file ";
    }
}
```

```
bool hasStudiedCourse( Student& student,  string& courseCode) {
    for (int i = 0; i < student.numCourses; i++) {
        if (student.courseCodes[i] == courseCode) {
            return true;
            break;
        }
    }
    return false;
}
int main() {
    Student student;
    string rollNumber,courseCode;

    cout << "Enter student's roll number: ";
    cin >> rollNumber;

    cout << "Enter course code: ";
    cin >> courseCode;
    ifstream   inFile;
    inFile.open("rollnumber.txt");

    loadDataFromFile(inFile, student);

    if (hasStudiedCourse(student, courseCode)) {
        cout << "Student with roll number " << student.rollNumber << " has studied
course " << courseCode << "." << endl;
    }
    else {
        cout << "Student with roll number " << student.rollNumber << " has not
studied course " << courseCode << "." << endl;
    }

}
```

```
Enter student's roll number: 19L-2113
Enter course code: CS2002
Student with roll number 19L-2113 has studied course CS2002.

D:\Project1\x64\Debug\Project1.exe (process 14008) exited with code 0.
Press any key to close this window . . .
```

```
Enter student's roll number: 20L-1113
Enter course code: CS1234
Student with roll number 19L-2113 has not studied course CS1234.

D:\Project1\x64\Debug\Project1.exe (process 1156) exited with code 0.
Press any key to close this window . . .
```



In this last code problem in has studied course function , but we make mahaool

Ques -4 is Rotate Array