ALGORITHMS LABORATORY
[CS-2098]

Individual Work

Lab. No.- 1 Topic- Random Numbers Date.- 24/07/2023

Roll Number:	**21051577**	**Branch/Section:**	**CSE-3**
Name in Capital:	**MANAV MALHOTRA**		

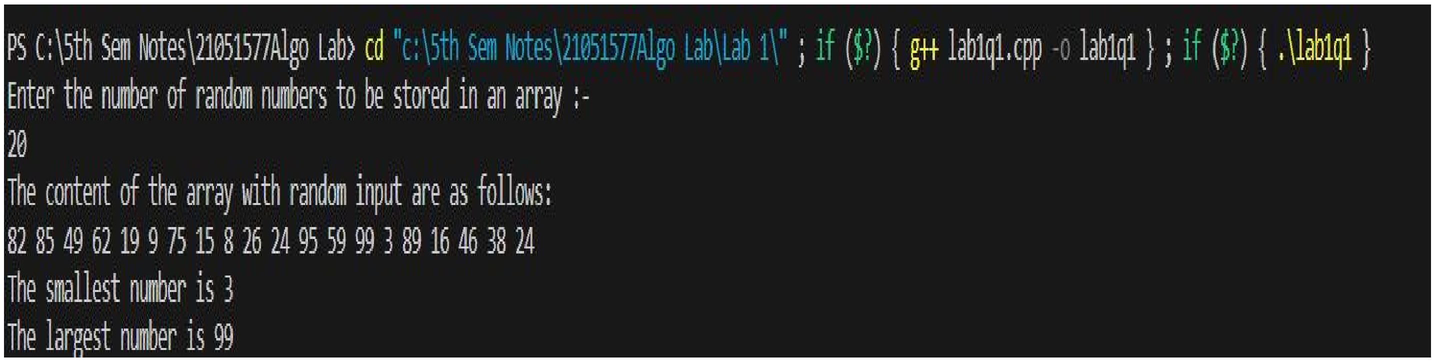
Program No: 1.1

Program Title:
Write a program to store random numbers into an array of n integers and then find out the smallest and largest number stored in it. n is the user input.

Sample Input	Sample Output
Enter number of random numbers to be stored in an array: 10	The content of the array with random input are as follows: 10 40 35 47 68 22 40 46 98 10 The smallest number is 10 The largest number is 98

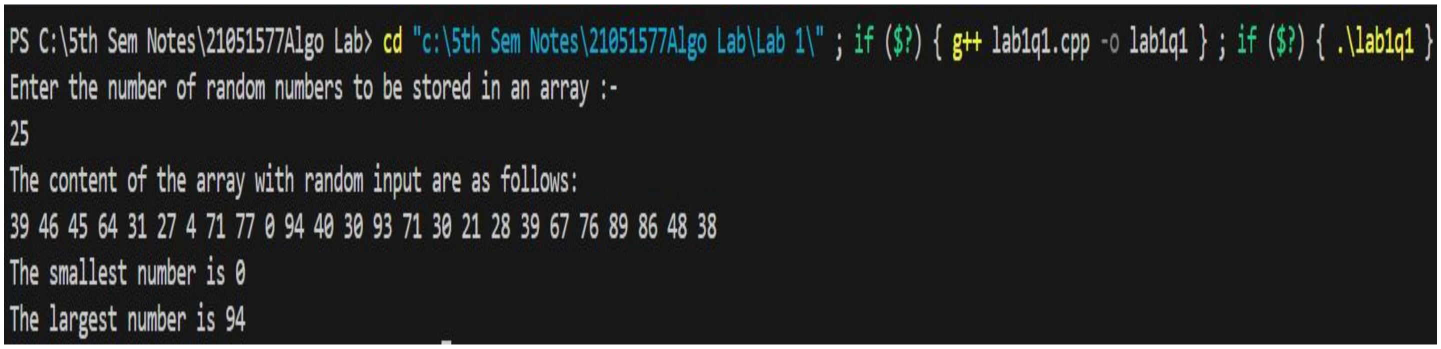
Input/Output Screenshots:

RUN-1:



```
PS C:\5th Sem Notes\21051577Algo Lab> cd "c:\5th Sem Notes\21051577Algo Lab\Lab 1\" ; if ($?) { g++ lab1q1.cpp -o lab1q1 } ; if ($?) { .\lab1q1 }
Enter the number of random numbers to be stored in an array :-
20
The content of the array with random input are as follows:
82 85 49 62 19 9 75 15 8 26 24 95 59 99 3 89 16 46 38 24
The smallest number is 3
The largest number is 99
```

RUN-2



```
PS C:\5th Sem Notes\21051577Algo Lab> cd "c:\5th Sem Notes\21051577Algo Lab\Lab 1\" ; if ($?) { g++ lab1q1.cpp -o lab1q1 } ; if ($?) { .\lab1q1 }
Enter the number of random numbers to be stored in an array :-
25
The content of the array with random input are as follows:
39 46 45 64 31 27 4 71 77 0 94 40 30 93 71 30 21 28 39 67 76 89 86 48 38
The smallest number is 0
The largest number is 94
```

<u>Source code</u>

```
#include <iostream>
#include <time.h>

using namespace std;

int main()
{
        int n;
        cout << "Enter the number of random numbers to be stored in an array: " << endl;
        cin >> n;

        // Dynamically allocate memory for an integer array of size n
           int *arr = new int[n];

         // Seed the random number generator using the current time
        srand(time(NULL));

        // Fill the array with random numbers between 0 and 99
        for (int i = 0; i < n; i++)
                arr[i] = rand() % 100;

        // Initialize variables to store the smallest (s) and largest (l) numbers
        int s = arr[0], l = arr[0];

        // Find the smallest and largest numbers in the array
        for (int i = 1; i < n; i++)
        {
                if (s > arr[i])
                        s = arr[i];
                if (l < arr[i])
                        l = arr[i];
        }

        // Display the content of the array with random inputs
         cout << "The content of the array with random input are as follows:" << endl;
        for (int i = 0; i < n; i++)
                cout << arr[i] << " ";
        cout << endl;

        // Display the smallest and largest numbers
        cout << "The smallest number is " << s << endl;
        cout << "The largest number is " << l;

        // Deallocate the memory for the dynamically allocated array
        delete[] arr;
        return 0;
}
```

Conclusion/Observation

The code is successfully executed with zero errors.

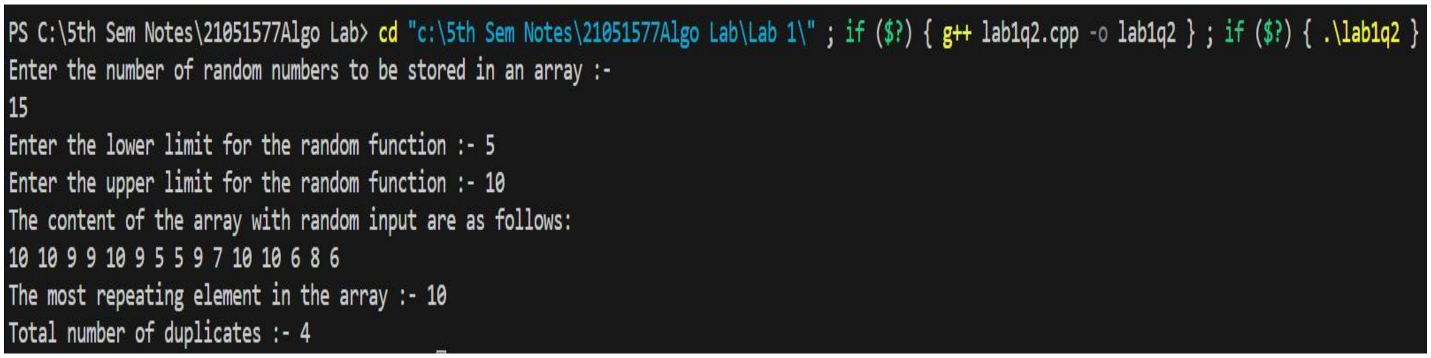
Program No: 1.2

Program Title:
Write a program to store random numbers into an array of n integers, where the array must contains some duplicates. Do the following:
a) Find out the total number of duplicate elements.
b) Find out the most repeating element in the array.
Hints: Write the random generation function in such a way that it must generate some duplicate numbers. As for example if you generate 10 random numbers within a range say 11 to 15, then atleast 5 numbers will be duplicate.

Sample Input	Sample Output
Enter number of random numbers to be stored in an array: 15	The content of the array with random input are as follows: 10 40 35 47 68 22 40 10 98 10 50 35 68 40 10 Total number of duplicates = 4 The most repeating element in the array = 10

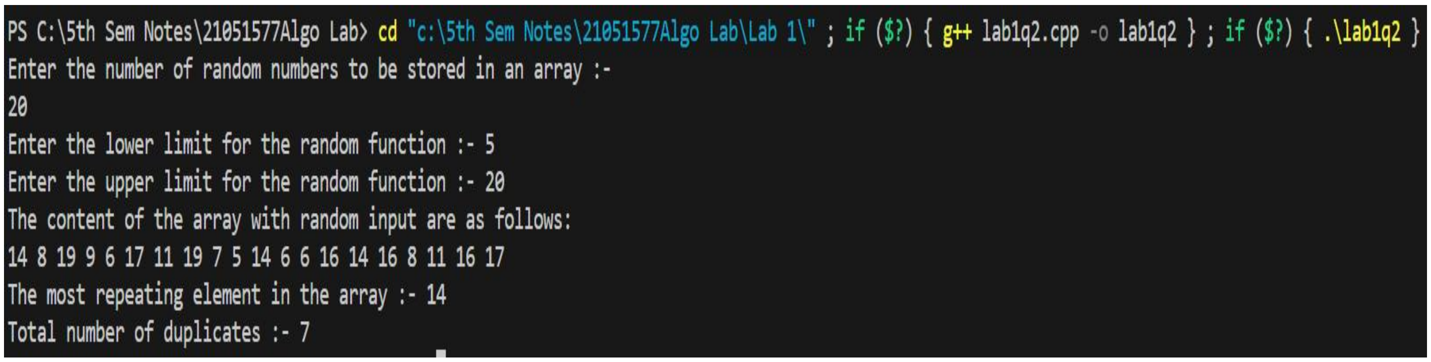
Input/Output Screenshots:

RUN-1:



```
PS C:\5th Sem Notes\21051577Algo Lab> cd "c:\5th Sem Notes\21051577Algo Lab\Lab 1\" ; if ($?) { g++ lab1q2.cpp -o lab1q2 } ; if ($?) { .\lab1q2 }
Enter the number of random numbers to be stored in an array :-
15
Enter the lower limit for the random function :- 5
Enter the upper limit for the random function :- 10
The content of the array with random input are as follows:
10 10 9 9 10 9 5 5 9 7 10 10 6 8 6
The most repeating element in the array :- 10
Total number of duplicates :- 4
```

RUN-2



```
PS C:\5th Sem Notes\21051577Algo Lab> cd "c:\5th Sem Notes\21051577Algo Lab\Lab 1\" ; if ($?) { g++ lab1q2.cpp -o lab1q2 } ; if ($?) { .\lab1q2 }
Enter the number of random numbers to be stored in an array :-
20
Enter the lower limit for the random function :- 5
Enter the upper limit for the random function :- 20
The content of the array with random input are as follows:
14 8 19 9 6 17 11 19 7 5 14 6 6 16 14 16 8 11 16 17
The most repeating element in the array :- 14
Total number of duplicates :- 7
```

Source code

```
#include <iostream>
#include <cstdlib>

using namespace std;

// Function to populate the array with random numbers in a specified range [l, u]
void createarray(int *arr, int n)
{
        int u, l;
        cout << "Enter the lower limit for the random function: ";
        cin >> l;
        cout << "Enter the upper limit for the random function: ";
        cin >> u;

        // Generate random numbers within the specified range and store them in the array
        for (int i = 0; i < n; i++)
                arr[i] = (rand() % (u - l + 1)) + l;
}

// Function to check for duplicate elements in the array
void checkduplicate(int *arr, int n)
{
        int x = 0, flag;

        // Compare each element with every other element to find duplicates
        for (int i = 0; i < n; i++)
        {
                flag = 0;
                for (int j = i + 1; j < n; j++)
                {
                        if (arr[i] == arr[j])
                        {
                                if (arr[j] == -1)
                                {
                                        continue;
                                }
                                else if (flag == 0)
                                {
                                        x++;
                                        arr[j] = -1; // Mark the duplicate element as -1
                                        flag = 1;
                                }
                                else if (flag == 1)
                                {
                                        arr[j] = -1;
                                }
                        }
                }
```

```
    }
    cout << "Total number of duplicates: " << x;
}

// Function to find the element with the highest frequency (most repeating element)
void getMaxRepeatingElement(int *arr, int n)
{
    int k, repeatcount = 0;

    for (int i = 0; i < n; i++)
    {
        int count = 1;
        for (int j = i + 1; j < n; j++)
        {
            if (arr[j] == arr[i])
            {
                count++;
            }
        }

        if (count > repeatcount)
        {
            repeatcount = count;
            k = arr[i];
        }
    }

    cout << "The most repeating element in the array: " << k << endl;
}

int main()
{
    int n;
    cout << "Enter the number of random numbers to be stored in an array: " << endl;
    cin >> n;

    // Dynamically allocate memory for an integer array of size n
    int *arr = new int[n];

    createarray(arr, n);

    cout << "The content of the array with random input are as follows:" << endl;
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;

    getMaxRepeatingElement(arr, n);
    checkduplicate(arr, n);

    // Deallocate the memory for the dynamically allocated array
    delete[] arr;

    return 0;
}
```

<u>Conclusion/Observation</u>

The code is successfully executed with zero errors.

Program No: 1.3

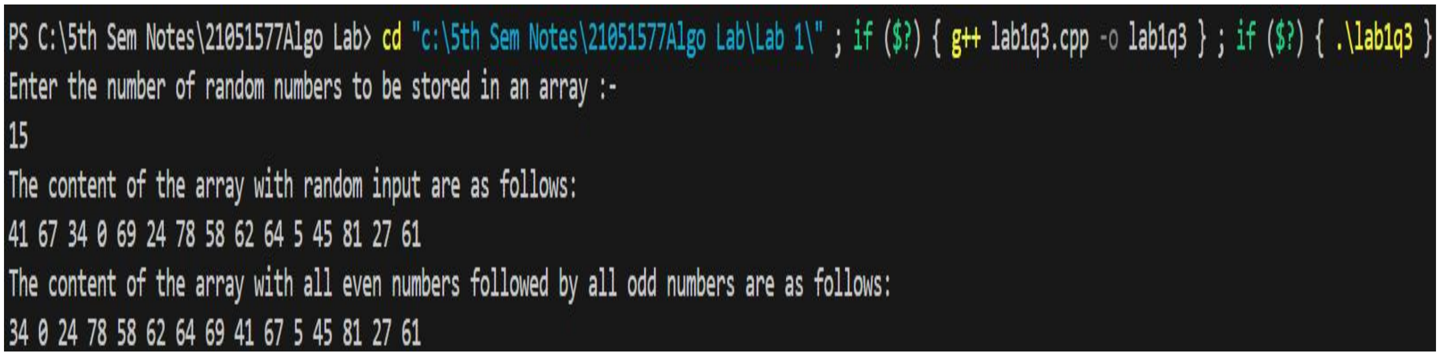
Program Title:
Write a program to rearrange the elements of an array of n integers such that all even numbers are followed by all odd numbers. How many different ways you can solve this problem. Write your approaches & strategy for solving this problem.

Sample Input	Sample Output
Enter number of random numbers to be stored in an array: 10	The content of the array with random input are as follows: 0, 1, 3, 6, 4, 9, 8, 7, 5, 2 The content of the array with all even numbers followed by all odd numbers are as follows: 1, 3, 5, 7, 9, 0, 2, 4, 6, 8 Or 5, 1, 3, 7, 9, 4 , 8, 6, 0, 2

Note: Other outputs may also posible depending on the logic used, but it must satisfy the requirements, First all odd numbers will be stored, then all even numbers, but the sequence within a group may vary.

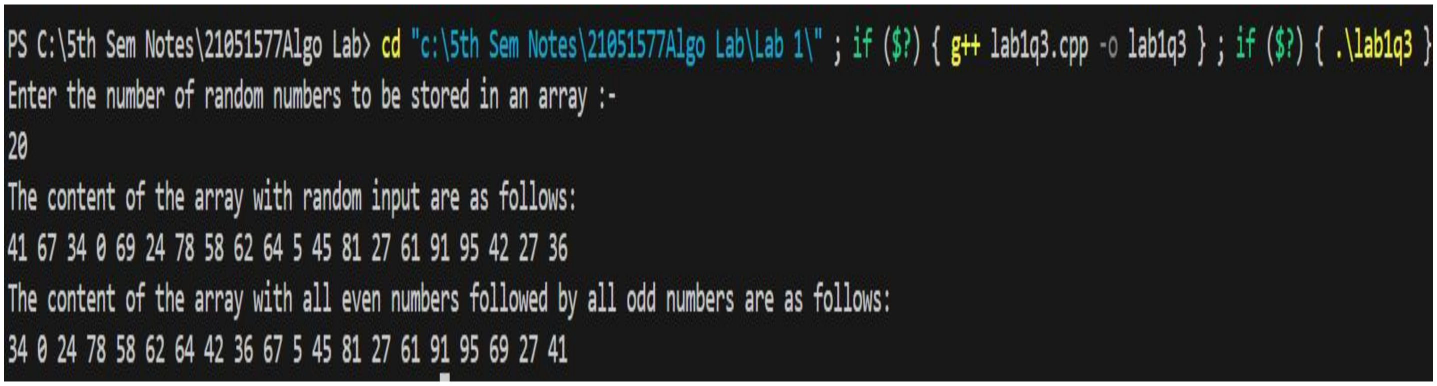
Input/Output Screenshots:

RUN-1:



```
PS C:\5th Sem Notes\21051577Algo Lab> cd "c:\5th Sem Notes\21051577Algo Lab\Lab 1\" ; if ($?) { g++ lab1q3.cpp -o lab1q3 } ; if ($?) { .\lab1q3 }
Enter the number of random numbers to be stored in an array :-
15
The content of the array with random input are as follows:
41 67 34 0 69 24 78 58 62 64 5 45 81 27 61
The content of the array with all even numbers followed by all odd numbers are as follows:
34 0 24 78 58 62 64 69 41 67 5 45 81 27 61
```

RUN-2



```
PS C:\5th Sem Notes\21051577Algo Lab> cd "c:\5th Sem Notes\21051577Algo Lab\Lab 1\" ; if ($?) { g++ lab1q3.cpp -o lab1q3 } ; if ($?) { .\lab1q3 }
Enter the number of random numbers to be stored in an array :-
20
The content of the array with random input are as follows:
41 67 34 0 69 24 78 58 62 64 5 45 81 27 61 91 95 42 27 36
The content of the array with all even numbers followed by all odd numbers are as follows:
34 0 24 78 58 62 64 42 36 67 5 45 81 27 61 91 95 69 27 41
```

Source code

```
#include <iostream>
#include <cstdlib>

using namespace std;

// Function to populate the array with random numbers in a specified range [l, u]
void createarray(int *arr, int n)
{
        int u, l;
        cout << "Enter the lower limit for the random function: ";
        cin >> l;
        cout << "Enter the upper limit for the random function: ";
        cin >> u;

         // Generate random numbers within the specified range and store them in the array
        for (int i = 0; i < n; i++)
                arr[i] = (rand() % (u - l + 1)) + l;
}

// Function to check for duplicate elements in the array
void checkduplicate(int *arr, int n)
{
        int x = 0, flag;

        // Compare each element with every other element to find duplicates
        for (int i = 0; i < n; i++)
        {
                flag = 0;
                for (int j = i + 1; j < n; j++)
                {
                        if (arr[i] == arr[j])
                        {
                                if (arr[j] == -1)
                                {
                                        continue;
                                }
                                else if (flag == 0)
                                {
                                        x++;
                                        arr[j] = -1; // Mark the duplicate element as -1
                                        flag = 1;
                                }
                                else if (flag == 1)
                                {
                                        arr[j] = -1;
                                }
                        }
                }
        }
         cout << "Total number of duplicates: " << x;
}

// Function to find the element with the highest frequency (most repeating element)
void getMaxRepeatingElement(int *arr, int n)
{
        int k, repeatcount = 0;

        for (int i = 0; i < n; i++)
        {
                int count = 1;
                for (int j = i + 1; j < n; j++)
                {
                        if (arr[j] == arr[i])
                        {
                                count++;
                        }
                }

                if (count > repeatcount)
                {
                        repeatcount = count;
                        k = arr[i];
                }
        }

        cout << "The most repeating element in the array: " << k << endl;
}

int main()
{
        int n;
        cout << "Enter the number of random numbers to be stored in an array: " << endl;
        cin >> n;

        // Dynamically allocate memory for an integer array of size n
        int *arr = new int[n];

        createarray(arr, n);

         cout << "The content of the array with random input are as follows:" << endl;
        for (int i = 0; i < n; i++)
                cout << arr[i] << " ";
        cout << endl;

        getMaxRepeatingElement(arr, n);
        checkduplicate(arr, n);
```

```
    // Deallocate the memory for the dynamically allocated array
    delete[] arr;

    return 0;
}
```

Conclusion/Observation

The code is successfully executed with zero errors.

Approach and strategy:

The rearrangeEvenAndOdd function houses the logic that drives the code. By switching all the even integers with the elements at the current indexes i and j, it does one iteration of the array. The last even integer position in the array is tracked by the variable j.

Due to the fact that this method only iterates through the array once, its time complexity is O(n), where n is the array's size. Because just the initial array is used, the space complexity is O(1).

The code demonstrates how to divide an array of numbers into even and odd numbers by rearranging their places.It is useful for testing and comprehending how the rearrangement works because it allows users the freedom to select the array's size and generates random integers for them.