Report No: 1
Report Name: Write a program to print all unique elements in the array.
Code:

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

int main(){
   int i, size, count;
   cout << "Enter array size: ";
   cin >> size;
   int myArray[size];

   for(i = 0 ; i < size; i++){
      cout << "Enter " << (i+1) << " Element: ";
      cin >> myArray[i];
   }

   cout << "Unique element array: ";
   for(i = 0; i < size ; i++){
      count = 0;

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

      for(int k = i+1; k < size ; k++){
         if(myArray[i] == myArray[k]){
            count ++;
         }
      }

      if(count == 0){
         cout << myArray[i];
      }
   }
   cout <<endl;
}
```

Output:

```
Enter array size: 4
Enter 1 Element: 2
Enter 2 Element: 2
Enter 3 Element: 3
Enter 4 Element: 5
Unique element array: 3, 5
```

Report No: 2
Report Name: Write a program to print all negative elements in an array.
Code:

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

int main(){
   int i, size, count;
   cout << "Enter array size: ";
   cin >> size;
   int myArray[size];

   for(i = 0 ; i < size; i++){
      cout << "Enter " << (i+1) << " Element: ";
      cin >> myArray[i];
   }
   cout << "Negative element array: ";
   for(i = 0; i < size; i++){
      if(myArray[i] < 0){
         cout << myArray[i] << ",  ";
      }
   }
}
```

Output:

```
Enter array size: 4
Enter 1 Element: 10
Enter 2 Element: -87
Enter 3 Element: 90
Enter 4 Element: -23
Negative element array: -87 , -23
```

Report No: 3
Report Name: Write a program to count the number of duplicate elements.
Code:

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

int main()
{
   int size;
   cout << "Enter array size: ";
   cin >> size;

   int myArray[size];
   for (int i = 0; i < size; i++)
   {
```

```
    cout << "Enter " << (i+1) << " element: ";
    cin >> myArray[i];
  }

  int count = 0;

  for (int i = 0; i < size - 1; i++)
  {
    bool isDuplicate = false;
    for (int j = i + 1; j < size; j++)
    {
      if (myArray[i] == myArray[j])
      {
        isDuplicate = true;
        break;
      }
    }
    if (isDuplicate)
    {
      count++;
      while (i < size - 1 && myArray[i] == myArray[i + 1])
      {
        i++;
      }
    }
  }

  cout << "Total Duplicate number: " << count << endl;
}
```

Output:

```
Enter array size: 4
Enter 1 element: 1
Enter 2 element: 2
Enter 3 element: 2
Enter 4 element: 3
Total Duplicate number: 1
```

Report No: 4

Report Name: Write a program to count the number of unique elements.

Code:

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

int main(){
  int i, size, duplicate, count = 0;
```

```
   cout << "Enter array size: ";
   cin >> size;
   int myArray[size];

   for(i = 0 ; i < size; i++){
      cout << "Enter " << (i+1) << " Element: ";
      cin >> myArray[i];
   }

   cout << "Unique element array: ";
   for(i = 0; i < size ; i++){
      duplicate = 0;

      for(int j = 0; j < i; j++){
         if(myArray[i] == myArray[j]){
            duplicate ++;
         }
      }

      for(int k = i+1; k < size ; k++){
         if(myArray[i] == myArray[k]){
            duplicate ++;
         }
      }

      if(duplicate == 0){
         count++;
      }
   }
   cout << "Total unique element: " <<count;
}
```

Output:

```
Enter array size: 4
Enter 1 Element: 1
Enter 2 Element: 5
Enter 3 Element: 5
Enter 4 Element: 6
Unique element array: Total unique element: 2
```

Report No: 5

Report Name: Write a program to sort array elements in descending order.

Code:

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

```
int main()
{
   int n, i;
   cout << "Enter size of array: ";
   cin >> n;
   int myArray[n];
   for (i = 0; i < n; i++)
   {
      cout << "Enter " << (i + 1) << " element: ";
      cin >> myArray[i];
   }
   for (i = 0; i < n; i++)
   {
      for (int j = 0; j < n - 1; j++)
      {
         if (myArray[j] < myArray[j + 1])
         {
            int temp = myArray[j + 1];
            myArray[j + 1] = myArray[j];
            myArray[j] = temp;
         }
      }
   }
   cout << "Bubble Sort new descending: ";
   for (i = 0; i < n; i++)
   {
      cout << myArray[i] << " ";
   }
}
```

Output:

```
Enter size of array: 4
Enter 1 element: 65
Enter 2 element: 10
Enter 3 element: 45
Enter 4 element: 98
Bubble Sort new descending order: 98 65 45 10
```

Report No: 6

Report Name: Write a program push element in stack and print in descending order.

Code:

```
#include <bits/stdc++.h>
using namespace std;

int top = -1;
#define size 5
```

```
int myStack [size];
void pushFunction();
void showFunction();
void mainFunction();

int main()
{
   mainFunction();
}

void pushFunction()
{
   system("cls");
   int value;
   if (top == size - 1)
   {
      cout << "Stack is overflow";
      cout << "\nPress any key to continue";
      fflush(stdin);
      getchar();
      main();
   }
   else
   {
      top = top + 1;
      cout << "\nEnter " << top << " index element: ";
      cin >> value;
      myStack [top] = value;
      cout << "\nPress any key to continue";
      fflush(stdin);
      getchar();
      main();
   }
}

void showFunction()
{
   system("cls");
   if (top == -1)
   {
      cout << "Stack is underflow";
      cout << "\nPress any key to continue";
      fflush(stdin);
      getchar();
      main();
   }
   cout <<"Reverse Order: ";
   while (top >= 0)
```

```
    {
        cout << myStack [top] << " ";
        top--;
    }   cout << "\nPress any key to continue";
    fflush(stdin);
    getchar();
    main();
}

void mainFunction()
{
    system("cls");
    int choice;
    cout << "1. Insert element (Push)" << endl;
    cout << "2. Show Function" << endl;
    cout << "Choice your option: ";
    cin >> choice;

    switch (choice)
    {
    case 1:
        pushFunction();
        break;

    case 2:
        showFunction();
        break;

    default:
        cout << "Something went wrong";
        cout << "\nPress any key to continue";
        fflush(stdin);
        getchar();
        main();
    }
}
```

Output:

```
1. Insert element (Push)
2. Show Function
Choice your option:     1

Enter 0 index element: 10
Enter 1 index element: 20
Enter 2 index element: 30
Enter 3 index element: 40
Enter 4 index element: 50
```

Reverse Order: 50 40 30 20 10

Report No: 7
Report Name: Write a program to sorted order with using quick sort.
Code:

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

void quickSort(int number[20], int first, int last)
{
   int i, j, pivot, temp;

   if (first < last)
   {
      pivot = first;
      i = first;
      j = last;
      while (i < j)
      {
         while (number[i] <= number[pivot] && i < last)
            i++;
         while (number[j] > number[pivot])
            j--;
         if (i < j)
         {
            temp = number[i];
            number[i] = number[j];
            number[j] = temp;
         }
      }
      temp = number[pivot];
      number[pivot] = number[j];
      number[j] = temp;
      quickSort(number,first,j-1);
      quickSort(number, j + 1, last);
   }
}

int main()
{
   int i, count, number[20];
   cout << "Enter elements (<= 20): ";
   cin >> count;
   for (i = 0; i < count; i++)
   {
      cout << "Enter " << (i + 1) << " element: ";
```

```
      cin >> number[i];
   }
   quickSort(number, 0, count - 1);
   cout << "The Sorted Order is: ";
   for (i = 0; i < count; i++){
      cout << number[i] << " ";
   }
}
```

Output:

```
Enter elements (<= 20): 4
Enter 1 element: 65
Enter 2 element: 12
Enter 3 element: 36
Enter 4 element: 69
The Sorted Order is: 12 36 65 69
```