```
1)_Search_Element.cpp X
           #include <iostream>
     2
           using namespace std;
      3
     4
           int search(int arr[], int n, int x)
     5
                                                    Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete_Data-Structure_&_Algorithms\1). Arr... —
      6
                for(int i = 0; i < n; i++)
                                                    Searched Index = 1
     8
                     if(arr[i] == x)
                                                    Process returned 0 (0x0)
                                                                                 execution time : 0.166 s
     9
                           return i;
                                                    Press any key to continue.
    10
    11
    12
                return -1;
    13
    14
    15
           int main()
    16
                int arr[] = \{20, 5, 7, 25\}, x = 5;
    17
                cout<<"Searched Index = "<<search(arr, 4, x)<<endl;;</pre>
    18
    19
    20
```

```
2) Insert_Element.cpp X
         #include <iostream>
                                                                ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete_Data-Structure_&_Algorithms\1...
         #include <cmath>
                                                                Before Insertion
         using namespace std;
                                                                5 10 20
         int insert(int arr[], int n, int x, int cap, int pos
                                                                After Insertion
   6
       ={
   7
             if(n == cap)
                                                                5 7 10 20
   8
                 return n;
   9
             int idx = pos - 1;
                                                                Process returned 0 (0x0)
                                                                                                          execution time: 0.286 s
  10
             for (int i = n - 1; i >= idx; i--)
                                                                Press any key to continue.
  11
  12
                 arr[i + 1] = arr[i];
  13
  14
             arr[idx] = x;
  15
             return n + 1;
  16
  17
  18
         int main()
  19
       ={
  20
                int arr[5], cap = 5, n = 3;
  21
                arr[0] = 5; arr[1] = 10; arr[2] = 20;
  22
                cout << "Before Insertion" << endl;
                for(int i=0; i < n; i++)
  23
  24
  25
                 cout << arr[i] << " ";
  26
  27
                cout << endl;
  28
                int x = 7, pos = 2;
  29
                n = insert(arr, n, x, cap, pos);
  30
                cout << "After Insertion" << endl;
  31
                for(int i=0; i < n; i++)
  32
  33
                     cout << arr[i] << " ";
  34
  35
```

```
3) Delete Element.cpp X
         #include <iostream>
         #include <cmath>
   3
         using namespace std;
                                                               Before Deletion
                                                               3 8 12 5 6
   5
         int deleteEle(int arr[], int n, int x)
                                                               After Deletion
             int i = 0;
   8
             for(i = 0; i < n; i++)
                                                               3 8 5 6
   9
  10
                 if(arr[i] == x)
  11
                     break;
  12
  13
             if(i == n)
  14
                 return n;
  15
             for(int j = i; j < n - 1; j++)
  16
  17
                 arr[j] = arr[j + 1];
  18
  19
             return n-1;
  20
  21
  22
         int main()
  23
                int arr[] = \{3, 8, 12, 5, 6\}, x = 12, n = 5;
  24
  25
                cout<<"Before Deletion"<<endl;
  26
                for(int i=0; i < n; i++)
  27
                 cout<<arr[i]<<" ";
  29
  30
                cout << endl;
  31
                n = deleteEle(arr, n, x);
  32
                cout<<"After Deletion"<<endl;
  33
                for(int i=0; i < n; i++)
  34
  35
                     cout<<arr[i]<<" ";
  36
  37
```

```
■ Select "C\Users\Akash Singh\Documents\Coding\" CHALLENGE \Complete_Data-Structure_&_Algorithms\1). Array\_3\_... - □ ×

Before Deletion
3 8 12 5 6

After Deletion
3 8 5 6

Process returned 0 (0x0) execution time : 0.130 s

Press any key to continue.
```

```
4) Reverse Array.cpp X
    1
         #include <iostream>
                                                  Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "Complete Data-Structure & Algorith...
    2
          #include <cmath>
                                                 Before Reverse
    3
         using namespace std;
    4
                                                 10 5 7 30
    5
           void reverse(int arr[], int n)
                                                 After Reverse
    6
       - {
                                                 30 7 5 10
    7
              int low = 0, high = n - 1;
              while (low < high)
    8
                                                                                       execution time : 0.231 s
                                                 Process returned 0 (0x0)
    9
                                                 Press any key to continue.
   10
                  int temp = arr[low];
                  arr[low] = arr[high];
   11
   12
                  arr[high] = temp;
   13
   14
                  low++;
   15
                  high--;
   16
   17
   18
         int main()
   19
       ={
   20
                 int arr[] = \{10, 5, 7, 30\}, n = 4;
   21
   22
                 cout << "Before Reverse" << endl;
                 for (int i = 0; i < n; i++)
   23
   24
                      cout << arr[i] << " ";
   25
   26
   27
                 cout << endl;
   28
                 reverse (arr, n);
                 cout << "After Reverse" << endl;
   29
                 for(int i = 0; i < n; i++)
```

cout << arr[i] << " ";

```
_5)_Left_Rotate_Array_by_1.cpp X
         #include <iostream>
                                                       ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete_Data-Structure_&_Algorithms\1). Array\_5)_Left_Rotate_Array_by_1.e...
         #include <cmath>
                                                       Before Left Rotation
         using namespace std;
                                                       1 2 3 4 5
         void lRotateOne(int arr[], int n)
    6
                                                       After '1' Left Rotation
             int temp = arr[0];
    8
             for(int i = 1; i < n; i++)
                                                       2 3 4 5 1
    9
                                                       Process returned 0 (0x0)
                                                                                                    execution time : 0.130 s
   10
                 arr[i - 1] = arr[i];
   11
                                                       Press any key to continue.
   12
             arr[n-1] = temp;
   13
   14
   15
         int main()
   16
       ={
   17
                int arr[] = \{1, 2, 3, 4, 5\}, n = 5;
   18
                cout<<"Before Left Rotation"<<endl;</pre>
   19
                for(int i = 0; i < n; i++)
   20
                     cout<<arr[i]<<" ";
   21
   22
   23
                cout<<endl;
   24
                lRotateOne(arr, n);
   25
                cout<<"After !!! Left Rotation"<<endl;</pre>
```

27

29 30 31 for(int i = 0; i < n; i++)

cout<<arr[i]<<" ";

```
_6)_Left_Rotate_Array_by_d_(Time_dN).cpp X
          #include <iostream>
          #include <cmath>
          using namespace std;
          void lRotateOne(int arr[], int n)
              int temp = arr[0];
              for(int i = 1; i < n; i++)
  10
                  arr[i - 1] = arr[i];
  11
  12
              arr[n-1] = temp;
  13
  14
          void leftRotate(int arr[], int d, int n)
  15
              for(int i = 0; i < d; i++)
  16
  17
  18
                  1RotateOne(arr, n);
  19
  20
  21
          int main()
  22
  23
                 int arr[] = {1, 2, 3, 4, 5}, n = 5, d = 2;
  24
                 cout<<"Before Left Rotation"<<endl;
                 for(int i = 0; i < n; i++)
  25
  26
  27
                      cout<<arr[i]<<" ";
  28
  29
                 cout << endl:
  30
                 leftRotate(arr, d, n);
                 cout<<"After 121 Left Rotation"<<endl;
  31
  32
                 for(int i = 0; i < n; i++)
  33
                      cout<<arr[i]<<" ";
  34
  35
  36
  37
```

Before Left Rotation

1 2 3 4 5

After '2' Left Rotation

3 4 5 1 2

Process returned 0 (0x0) execution time: 0.207 s

Press any key to continue.

```
_7)_Left_Rotate_Array_by_d_(Time_N).cpp X
        #include <iostream>
                                               ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete_Data-Structure_&_Algorithms\1). Array\_7)_Left_Rotate_Array_by_d_(Time_N).exe"
                                                                                                                                                       #include <cmath>
   3
        using namespace std;
                                              Before Left Rotation
   4
   5
        void leftRotate(int arr[], int d, int n)
                                              1 2 3 4 5
   6
           int temp[d];
                                              After '2' Left Rotation
           for(int i = 0; i < d; i++)
   8
   9
                                              3 4 5 1 2
               temp[i] = arr[i];
  10
  11
                                              Process returned 0 (0x0) execution time : 0.094 s
  12
           for(int i = d; i < n; i++)
  13
  14
               arr[i - d] = arr[i];
                                              Press any key to continue.
  15
  16
           for(int i = 0; i < d; i++)
  17
  18
               arr[n - d + i] = temp[i];
  19
  20
  21
  22
        int main()
  23
  24
             int arr[] = \{1, 2, 3, 4, 5\}, n = 5, d = 2;
  25
             cout<<"Before Left Rotation"<<endl;
  26
              for(int i = 0; i < n; i++)
  27
  28
                  cout<<arr[i]<<" ";
```

31

32

34 35

36

37

cout << endl;

}

leftRotate(arr, d, n);

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

cout<<arr[i]<<" ";

cout<<"After "2" Left Rotation"<<endl;

```
_8)_Left_Rotate_Array_by_d_(Time_N).cpp X
          #include <iostream>
   2
          using namespace std;
   3
          void reverse (int arr[], int low, int high)
   5
              while (low < high)
                  swap(arr[high], arr[low]);
   9
                  low++;
  10
                  high--;
  11
  12
  13
  14
          void leftRotate(int arr[], int d, int n)
  15
  16
              reverse (arr, 0, d - 1);
  17
              reverse (arr, d, n - 1);
  18
              reverse (arr, 0, n - 1);
  19
  20
          int main()
  21
  22
                int arr[] = \{1, 2, 3, 4, 5\}, n = 5, d = 4;
  23
  24
                cout<<"Before Left Rotation"<<endl;
  25
                 for(int i = 0; i < n; i++)
  26
  27
                      cout<<arr[i]<<" ";
  28
  29
                 cout << endl;
  30
                 leftRotate(arr, d, n);
  31
                 cout<<"After '4' Left Rotation"<<endl;
  32
                 for(int i = 0; i < n; i++)
  33
  34
                      cout << arr[i] << " ";
  35
  36
  37
```

Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE \Complete_Data-Structure_&_Algorithms\1). Array_8_Left_Rotate_Array_by_d_... - \Rightarrow \times \text{Process returned 0 (0x0) execution time : 0.150 s Press any key to continue.

```
9) Remove Duplicates from a sorted array.cpp X
        #include<iostream>
                                            Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1), Array\ 9) Remove Duplicat...
        using namespace std;
   3
                                           Before Removal Duplicates: 10 20 20 30 30 30
                                                                                                                                        30
        int remDups(int arr[], int n)
                                           Size = 7
            int temp[n];
            temp[0] = arr[0];
                                           After Removal Duplicates: 10 20 30
            int res = 1:
            for(int i=0; i<n; i++)
                                           Size = 3
  10
  11
               if(temp[res-1] != arr[i])
  12
  13
                   temp[res] = arr[i];
                                           Process returned 0 (0x0) execution time: 0.170 s
                   res++:
  15
                                           Press any key to continue.
  16
  17
            for(int i=0; i<n; i++)
  19
               arr[i] = temp[i];
  20
  21
            return res;
  23
  24
        int main()
  25
       E (
            int arr[] = {10, 20, 20, 30, 30, 30, 30}, n=7;
  27
            cout << "Before Removal Duplicates: ";
            for(int i=0; i<n; i++)
  29
               cout << arr[i] << " ";
  31
            cout << endl << "Size = " << n << endl;
  33
            int Res = remDups(arr, n);
  35
            cout << "After Removal Duplicates: ";
  36
            for(int i=0; i<Res; i++)
  37
               cout << arr[i] << ";
  39
  40
            cout << endl << "Size = " << Res << endl;
  41
```

```
_10)_Remove_Duplicates_from_a_sorted_array.cpp X
         #include<iostream>
                                                Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1). Array\ 1...
         using namespace std;
                                               Before Removal Duplicates: 10 20 20 30 30
                                                                                                                               30
                                                                                                                        30
         int remDups(int arr[], int n)
                                               Size = 7
       ={
    6
             int res = 1;
                                               After Removal Duplicates: 10 20 30
             for (int i=1; i<n; i++)
   8
                                               Size = 3
   9
                 if(arr[i] != arr[res-1])
  10
  11
                     arr[res] = arr[i];
                                               Process returned 0 (0x0) execution time: 0.123 s
  12
                     res++;
  13
                                               Press any key to continue.
  14
  15
             return res;
  16
  17
  18
         int main()
  19
       = {
             int arr[] = {10, 20, 20, 30, 30, 30, 30}, n=7;
   20
   21
             cout << "Before Removal Duplicates: ";
             for(int i=0; i<n; i++)
   22
   23
                 cout << arr[i] << ";
   24
   25
   26
             cout << endl << "Size = " << n << endl;
   27
             int Res = remDups(arr, n);
   28
  29
             cout << "After Removal Duplicates: ";
             for (int i=0; i<Res; i++)
   30
   31
   32
                 cout << arr[i] << ";
   33
   34
             cout << endl << "Size = " << Res << endl;
   35
```

```
11) Largest Number.cpp X
          #include<iostream>
                                                   ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete-Data-Structure-and-Algorithm\code\1). Array\_11)_Largest_Numb...
          using namespace std;
                                                  Index = 2
          int getLargest(int arr[], int n)
        _ {
     6
               for(int i=0; i<n; i++)
                                                  Process returned 0 (0x0) execution time: 0.205 s
                                                  Press any key to continue.
     8
                   bool flag = true;
                   for(int j=1; j<n; j++)
    10
    11
                        if(arr[j] > arr[i])
    12
    13
                            flag = false;
    14
                            break;
    15
    16
    17
                   if (flag==true)
    18
                        return i;
    19
    20
               return -1;
    21
    22
    23
          int main()
    24
    25
               int arr[] = \{10, 5, 20, 8\}, n=4;
    26
               cout<<"Index = "<<qetLargest(arr, n)<<endl;</pre>
```

```
_12)_Largest Number.cpp X
            #include<iostream>
                                                         Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \Complete-Data-Structure-and-Alg...
            using namespace std;
                                                         Index = 2
                                                        Process returned 0 (0x0) execution time : 0.207 s
            int getLargest(int arr[], int n)
Press any key to continue.
      5
      6
                  int res=0;
                  for(int i=1; i<n; i++)
                       if(arr[i] > arr[res])
       9
                             res = i;
     10
                  return res;
     11
     12
     13
            int main()
     14
     15
                  int arr[] = \{10, 5, 20, 8\}, n=4;
                  cout<<"Index = "<<qetLargest(arr, n)<<endl;</pre>
     16
     17
```

```
13) Second Largest Number.cpp X
          #include < iostream>
                                                  ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete-Data-Structure-and-Algorithm\code\1). Array\_13)_Second_Largest_Number.e... —
         using namespace std;
    3
                                                  Index = 2
         int getLargest(int arr[], int n)
       ⊟{
             int largest=0;
                                                 Process returned 0 (0x0)
                                                                                                     execution time : 0.119 s
             for (int i=1; i<n; i++)
                 if(arr[i] > arr[largest])
                                                 Press any key to continue.
    9
                     largest = i;
   10
             return largest;
   11
   12
   13
         int getSecondLargest(int arr[], int n)
   14
       F (
   15
             int largest = getLargest(arr, n);
   16
             int res = -1;
   17
             for(int i=0; i<n; i++)
   18
   19
                 if(arr[i] != arr[largest])
   20
   21
                     if (res == -1)
   22
                          res = i;
   23
                      else if (arr[i] > arr[res])
   24
                          res = i;
   25
   26
   27
             return res;
   28
   29
   30
         int main()
   31
       - {
   32
             int arr[] = \{10, 5, 18, 20\}, n=4;
   33
             cout<<"Index = "<<getSecondLargest(arr, n) <<endl;;</pre>
```

```
14) Second Largest Number.cpp X
          #include<iostream>
                                                     Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete-Data-Structure-and-Algorithm\code\1). Array\_14)_Second_La... —
          using namespace std;
                                                     Index = 2
          int getSecondLargest(int arr[], int n)
                                                     Process returned 0 (0x0) execution time : 0.099 s
               int res = -1, largest = 0;
               for(int i=1; i<n; i++)
                                                     Press any key to continue.
     8
     9
                   if(arr[i] > arr[largest])
   10
   11
                        res=largest;
                        largest=i;
   13
   14
                   else if(arr[i] != arr[largest])
   15
   16
                        if(res == -1 ||arr[i] > arr[res])
   17
                            res = i;
   18
   19
   20
               return res;
   22
   23
          int main()
   24
   25
              int arr[] = \{10, 5, 18, 20\}, n=4;
   26
               cout<<"Index = "<<qetSecondLargest(arr, n)<<endl;;</pre>
```

```
15) Move All ZEROs to End.cpp X
        #include<iostream>
                                            Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1). Array\_15) Move_All_ZEROs...
                                                                                                                                                   using namespace std;
                                           Before Move Zeros: 0 8 5 0 0
                                                                                                               10
                                                                                                                              20
        void moveToEnd(int arr[], int n)
      E1
                                           Size = 8
           for(int i=0; i<n; i++)
              if(arr[i] == 0)
                                           After Move Zeros: 20 10
  10
                  for(int j=i+1; j<n; j++)
                                           Size = 8
  12
                     if (arr[j] != 0)
  13
  14
                         int temp = arr[i];
  15
                         arr[i] = arr[i];
                                           Process returned 0 (0x0) execution time : 0.076 s
                         arr[j] = temp;
  17
                                           Press any key to continue.
  18
  19
  20
  22
        int main()
           int arr[] = {0, 8, 5, 0, 0, 10, 0, 20}, n=8;
  26
           cout << "Before Move Zeros: ";
           for (int i=0; i<n; i++)
  28
               cout<<arr[i]<<" ";
  30
  31
           cout << endl << "Size = " << n << endl;
  32
  33
           moveToEnd(arr, n);
           cout << "After Move Zeros: ";
  35
            for(int i=0; i<n; i++)
  36
  37
              cout << arr[i] << ";
  39
           cout << endl << "Size = " << n << endl;
```

```
#include<iostream>
                                       Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1). Array\ 16) Move... —
      using namespace std;
                                      Before Move Zeros: 0 8 5 0 0
                                                                                                 10
                                                                                                              20
      void moveToEnd(int arr[], int n)
                                      Size = 8
 6
         int count = 0;
         for(int i=0; i<n; i++)
                                      After Move Zeros: 8 5 10
                                                                                       20
9
             if(arr[i] != 0)
                                      Size = 8
10
11
                int temp = arr[i];
12
                arr[i] = arr[count];
13
                arr[count] = temp;
                                      Process returned 0 (0x0) execution time: 0.139 s
14
                count++;
15
                                      Press any key to continue.
16
17
18
19
      int main()
20
    - 1
21
         int arr[] = {0, 8, 5, 0, 0, 10, 0, 20}, n=8;
22
         cout << "Before Move Zeros: ";
23
         for(int i=0; i<n; i++)
24
             cout<<arr[i]<<" ";
25
26
         cout<<endl<<"Size = "<<n<<endl;
27
28
29
         moveToEnd(arr, n);
         cout<< "After Move Zeros: ";
30
31
         for(int i=0; i<n; i++)
32
             cout << arr[i] << " ";
33
34
35
         cout<<endl<<"Size = "<<n<<endl;
```

16) Move All ZEROs to End.cpp X

```
_017)_Leaders_in_an_Array.cpp X
          #include<iostream>
                                                    Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete-Data-Structure-and-Algorithm\code\1). Array\_017)_Leaders_in_an_Arra...
          using namespace std;
                                                   Leasers = 10 \ 6 \ 5 \ 2
          int leaders(int arr[], int n)
     5
     6
              for(int i=0; i<n; i++)
                                                                                                execution time : 0.101 s
                                                  Process returned 0 (0x0)
                                                   Press any key to continue.
                   bool flag=false;
     9
                   for(int j=i+1; j<n; j++)</pre>
   10
   11
                       if(arr[i] <= arr[j])
   12
   13
                            flag=true;
   14
                            break;
   15
   16
   17
                   if(flag==false)
   18
                       cout << arr[i] << ";
   19
   20
   21
   22
          int main()
   23
              int arr[] = {7, 10, 4, 3, 6, 5, 2}, n=7;
   24
              cout<<"Leasers = ";
   26
               leaders (arr, n);
               cout << endl;
```

```
_018)_Leaders_in_an_Array.cpp X
           #include<iostream>
                                                   Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1). Array\_018)_Leaders_in... —
          using namespace std;
                                                   Leasers in Reverse Order = 2 5 6 10
           int leaders(int arr[], int n)
                                                  Process returned 0 (0x0)
                                                                                        execution time : 0.284 s
               int cur leader = arr[n-1];
                                                  Press any key to continue.
               cout << cur leader << ";
     8
     9
               for(int i=n-2; i>=0; i--)
    10
    11
                    if(cur leader < arr[i])</pre>
    12
    13
                         cur leader = arr[i];
    14
                         cout << cur leader << " ";
    15
    16
    17
    18
    19
           int main()
    20
               int arr[] = \{7, 10, 4, 3, 6, 5, 2\}, n=7;
               cout<<"Leasers in Reverse Order = ";</pre>
    22
    23
               leaders (arr, n);
    24
               cout << endl;
    25
```

```
019) Maximum Difference.cpp X
            #include<iostream>
                                                    Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE \Complete-Data-Structure-and-Algorithm\...
                                                    Maximum Difference = 8
            using namespace std;
                                                    Process returned 0 (0x0) execution time : 0.190 s
            int maxDiff(int arr[], int n) Press any key to continue.
                 int res = arr[1] - arr[0];
                 for(int i=0; i<n; i++)
                       for(int j=i+1; j<n; j++)
      9
                            res = max(res, arr[j]-arr[i]);
     10
                 return res;
     12
     13
            int main()
     14
                 int arr[] = \{2, 3, 10, 6, 4, 8, 1\}, n=7;
     15
                 cout<<"Maximum Difference = "<<maxDiff(arr,n)<<end1;</pre>
     16
```

```
_020)_Maximum_Difference.cpp X
           #include<iostream>
           using namespace std;
           int maxDiff(int arr[], int n)
      5
     6
                int res = arr[1]-arr[0], minVal = arr[0];
                for(int i=1; i<n; i++)
                                                            Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \Complete-Data-Structure-and-Algorithm\co... —
     8
                                                            Maximum Difference = 8
     9
                     res = max(res, arr[i]-minVal);
    10
                     minVal = min(minVal, arr[i]);
                                                           Process returned 0 (0x0)
                                                                                        execution time : 0.201 s
    11
                                                           Press any key to continue.
    12
                return res;
    13
    14
    15
           int main()
    16
    17
                int arr[] = \{2, 3, 10, 6, 4, 8, 1\}, n=7;
                cout<<"Maximum Difference = "<<maxDiff(arr,n)<<endl;</pre>
    18
    19
```

```
_021)_Stock_Buy_and Shell.cpp X
         #include<iostream>>
                                                                 ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\Complete-Data-Structure-and-Algorithm\code\1). Array... —
         using namespace std;
                                                                Maximum Profit in Stock Market = 13
         int maxProfit(int price[], int start, int end)
                                                                Process returned 0 (0x0) execution time: 0.122 s
              if (end <= start )
                  return 0:
                                                                Press any key to continue.
              int profit = 0;
              for(int i=start; i<end; i++)</pre>
   10
   11
                  for(int j=i+1; j<=end; j++)</pre>
   12
   13
                      if (price[j] > price[i])
   14
   15
                          int cur profit = price[j] - price[i] +
   16
                                           maxProfit(price, start, i-1) +
   17
                                           maxProfit (price, j+1, end);
   18
                          profit = max (profit, cur profit);
   19
   20
   21
   22
              return profit;
   23
   24
   25
         int main()
   26
       -1
   27
              int price[] = {1, 5, 3, 8, 12}, start=0, end = 5;
              cout<<"Maximum Profit in Stock Market = "<<maxProfit(price, start, end)<<end1;</pre>
   28
   29
```

```
022) Stock Buy and Shell.cpp X
            #include<iostream>
                                                         Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \Complete-Data-Structure-and-Algorith...
            using namespace std;
                                                         Maximum Profit in Stock Market = 13
                                                         Process returned 0 (0x0) execution time : 0.096 s
            int maxProfit(int price[], int n)
                                                         Press any key to continue.
                 int profit = 0;
      6
                 for(int i=1; i<n; i++)
                      if(price[i] > price[i-1])
                       profit += (price[i]-price[i-1]);
     10
                 return profit;
     11
     12
     13
            int main()
     14
                 int price[] = \{1, 5, 3, 8, 12\}, n=5;
     15
                 cout<<"Maximum Profit in Stock Market = "<<maxProfit(price, n) <<end1;</pre>
     16
     17
```

```
_023)_Trapping_Rain_Water.cpp X
          #include<iostream>
                                                 Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1), Array\ 023) Trap...
          using namespace std;
                                                Maximum fill Boxes = 6
          int getWater(int arr[], int n)
                                                Process returned 0 (0x0)
                                                                                     execution time : 0.076 s
     6
               int res = 0:
                                                Press any key to continue.
               for(int i=1; i<n-1; i++)
     8
                    int lmax = arr[i];
    10
                    for(int j=0; j<i; j++)
    11
                        lmax = max(lmax, arr[j]);
    12
    13
                    int rmax = arr[i];
    14
                    for(int j=i+1; j<n; j++)</pre>
    15
                        rmax = max(rmax, arr[j]);
    16
    17
                    res += (min(lmax, rmax) - arr[i]);
    18
    19
               return res;
    20
    21
    22
          int main()
    23
    24
               int arr[] = \{3, 0, 1, 2, 5\}, n=5;
    25
               cout<<"Maximum fill Boxes = "<<getWater(arr, n) <<end1;</pre>
    26
```

```
_024)_Trapping_Rain_Water.cpp X
          #include<iostream>
                                                  Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1). Array...
          using namespace std;
                                                  Maximum fill Boxes = 6
          int getWater(int arr[], int n)
                                                  Process returned 0 (0x0) execution time: 0.087 s
              int res=0;
                                                  Press any key to continue.
              int lmax[n], rmax[n];
     9
              lmax[0] = arr[0];
   10
              for(int i=1; i<n; i++)
   11
                   lmax[i] = max(arr[i], lmax[i-1]);
   12
   13
              rmax[n-1] = arr[n-1];
   14
              for(int i=n-2; i>=0; i--)
   15
                   rmax[i] = max(arr[i], rmax[i+1]);
   16
   17
              for(int i=1; i<n-1; i++)
   18
                   res += (min(lmax[i], rmax[i]) - arr[i]);
   19
              return res;
   20
   21
   22
          int main()
   23 - {
   24
              int arr[] = \{3, 0, 1, 2, 5\}, n=5;
   25
              cout<<"Maximum fill Boxes = "<<qetWater(arr, n)<<end1;</pre>
   26
```

```
025) Maximum Consecutive ONEs in binary Array.cpp X
          #include<iostream>
                                                              ■ Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE "\365-Days-of-Code\174). Complete DSA (Array[25-35])\_025)_Maxi...
          using namespace std;
                                                             Maximum 1's in binary Array = 3
          int maxConsecutiveOnes(bool arr[], int n)
    5
                                                             Process returned 0 (0x0) execution time: 0.311 s
              int res = 0;
              for(int i=0; i<n; i++)</pre>
                                                             Press any key to continue.
    8
    9
                   int curr = 0;
   10
                   for(int j=i; j<n; j++)</pre>
   12
                       if(arr[j] == 1)
   13
                            curr++;
   14
                       else
   15
                            break;
   16
   17
                   res = max(res, curr);
   18
   19
              return res;
   20
          int main()
   23
        - {
   24
              bool arr[] = \{0, 1, 1, 0, 1, 1, 1\};
              int n=7;
              cout<<"Maximum 1's in binary Array = "<<maxConsecutiveOnes(arr, n)<<endl;</pre>
   26
```

```
026) Maximum Consecutive ONEs in binary Array.cpp X
           #include<iostream>
                                                             Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \365-Days-of-Code\174), Complete DSA (Array[25-35]... —
          using namespace std;
                                                             Maximum 1's in binary Array = 3
          int maxConsecutiveOnes(bool arr[], int n)
                                                             Process returned 0 (0x0) execution time : 0.212 s
     6
               int res = 0, curr = 0;
                                                            Press any key to continue.
               for(int i=0; i<n; i++)
     8
                    if(arr[i] == 0)
    10
                        curr = 0;
    11
                    else
    12
    13
                        curr++;
    14
                        res = max (res, curr);
    15
    16
    17
               return res;
    18
    19
    20
          int main()
    21
    22
               bool arr[] = {0, 1, 1, 0, 1, 1, 1};
    23
               int n=7;
    24
               cout<<"Maximum 1's in binary Array = "<<maxConsecutiveOnes(arr, n)<<end1;</pre>
    25
```

```
027) Maximum Subarray Sum.cpp X
           #include<iostream>
                                                         ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\365-Days-of-Code\174). Complete DSA (Array[25-35])\_... —
           using namespace std;
                                                         Maximum SubArray Sum = 11
     4
           int maxSubarraySum(int arr[], int n)
                                                         Process returned 0 (0x0)
                                                                                         execution time : 0.183 s
     5
                                                         Press any key to continue.
     6
                int res = arr[0];
                for(int i=0; i<n; i++)
     8
     9
                     int curr = 0;
    10
                     for(int j=i; j<n; j++)</pre>
    11
    12
                          curr += arr[j];
    13
                          res = max(res, curr);
    14
    15
    16
                return res;
    17
    18
    19
           int main()
    20
    21
                int arr[] = \{2, 3, -8, 7, -1, 2, 3\}, n=7;
    22
                cout<<"Maximum SubArray Sum = "<<maxSubarraySum(arr,n)<<end1;</pre>
    23
```

```
028) Maximum_Subarray_Sum.cpp X
           #include<iostream>
                                                         Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\365-Days-of-Code\174). Complete DSA (Array[25-35]...
           using namespace std;
                                                        Maximum SubArray Sum = 11
           int maxSubarraySum(int arr[], int n)
                                                        Process returned 0 (0x0) execution time: 0.595 s
     5
                                                        Press any key to continue.
     6
                int res = arr[0];
                int maxEnding = arr[0];
                for(int i=1; i<n; i++)</pre>
     8
     9
    10
                    maxEnding = max(maxEnding + arr[i], arr[i]);
    11
                     res = max (res, maxEnding);
    12
    13
                return res;
    14
    15
    16
           int main()
    17
    18
                int arr[] = \{2, 3, -8, 7, -1, 2, 3\}, n=7;
                cout<<"Maximum SubArray Sum = "<<maxSubarraySum(arr,n)<<end1;</pre>
    19
    20
```

```
029) Maximum_Circular_Subarray_Sum.cpp X
          #include<iostream>
          using namespace std;
          int maxCircularSubarraySum(int arr[], int n)
                                                          Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \'365-Days-of-Code\174). Complete DSA (Array[25-35])\_02... —
     6
               int res = arr[0];
                                                          Maximum Circular SubArray Sum = 12
               for(int i=0; i<n; i++)</pre>
     8
                                                          Process returned 0 (0x0) execution time: 0.231 s
     9
                    int curr max = arr[i];
    10
                    int curr sum = arr[i];
                                                          Press any key to continue.
    11
                    for(int j=1; j<n; j++)
    12
    13
                        int index = (i+j)%n;
    14
                        curr sum += arr[index];
   15
                        curr max = max (curr max, curr sum);
   16
    17
                   res = max (res, curr max);
   18
    19
               return res;
    20
    21
   22
          int main()
    23
    24
               int arr[] = \{5, -2, 3, 4\}, n=4;
    25
               cout<<"Maximum Circular SubArray Sum = "<<maxCircularSubarraySum(arr,n)<<end1;</pre>
    26
```

```
030) Maximum Circular Subarray Sum.cpp X
          #include<iostream>
         using namespace std;
         int normalMaxSum(int arr[], int n)
       -1
              int res = arr[0];
              int maxEnding = arr[0];
              for(int i=1; i<n; i++)
   9
  10
                 maxEnding = max(maxEnding + arr[i], arr[i]);
  11
                 res = max (res, maxEnding);
  12
  13
              return res;
  14
  15
  16
          int maxCircularSubarraySum(int arr[], int n)
       = {
  17
             int max normal = normalMaxSum(arr,n);
  18
              if (max normal < 0)
  19
  20
                  return 0;
  21
              int arr sum = 0;
              for(int i=0; i<n; i++)
  23
  24
                 arr sum += arr[i];
  26
                 arr[i] = -arr[i];
  27
  28
              int max circular = arr sum + normalMaxSum(arr, n);
  29
              return max (max normal, max circular);
  30
  31
         int main()
       □{
  33
  34
              int arr[] = \{5, -2, 3, 4\}, n=4;
              cout<<"Maximum Circular SubArray Sum = "<<maxCircularSubarraySum(arr,n)<<end1;
  36
```

Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE \365-Days-of-Code\174). Complete DSA (Array[25-35])_030)_Maximum_Circular_S... — Maximum Circular SubArray Sum = 12

Process returned 0 (0x0) execution time : 0.113 s

Press any key to continue.

```
_031)_Maximum_length_Even_Odd_Subarray.cpp X
          #include<iostream>
                                                  Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE '\365-Days-of-Code\174). Complete DSA (Array[25-35])\_031)_Maximu...
          using namespace std;
                                                 Maximum Length of Even-Odd = 3
          int maxEvenOdd(int arr[], int n)
                                                 Process returned 0 (0x0) execution time : 0.179 s
               int res = 1;
                                                 Press any key to continue.
               for(int i=0; i<n; i++)
     8
     9
                   int curr = 1;
    10
                   for (int j=i+1; j<n; j++)
   11
   12
                        if((arr[j]%2 == 0 && arr[j-1]%2 != 0) ||
   13
                            (arr[j] %2 != 0 && arr[j-1] %2 == 0))
   14
                                 curr++;
   15
                        else
   16
                            break;
   17
   18
                   res = max (res, curr);
   19
   20
               return res;
    21
   22
   23
          int main()
   24
   25
               int arr[] = \{10, 12, 14, 7, 8\}, n=5;
               cout<<"Maximum Length of Even-Odd = "<<maxEvenOdd(arr, n)<<end1;</pre>
   26
```

```
032) Maximum length Even Odd Subarray.cpp X
          #include<iostream>
                                                   Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\365-Days-of-Code\174). Complete DSA (Array[2...
          using namespace std;
                                                   Maximum Length of Even-Odd = 3
          int maxEvenOdd(int arr[], int n)
                                                   Process returned 0 (0x0) execution time: 0.431 s
     6
               int res = 1, curr = 1;
                                                   Press any key to continue.
               for(int i=1; i<n; i++)
     8
     9
                   if((arr[i]%2 == 0 && arr[i-1]%2 != 0) ||
    10
                            (arr[i]%2 != 0 && arr[i-1]%2 == 0))
    11
    12
                        curr++;
    13
                        res = max (res, curr);
    14
    15
                    else
    16
                        curr = 1;
    17
    18
               return res;
    19
    20
    21
          int main()
    22
    23
               int arr[] = \{10, 12, 14, 7, 8\}, n=5;
    24
               cout<<"Maximum Length of Even-Odd = "<<maxEvenOdd(arr, n)<<end1;</pre>
    25
```

```
_033)_Majority_Element.cpp X
           #include<iostream>
                                                     Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \'365-Days-of-Code\174\). Complete DSA (Array[25-35])\_033\_...
           using namespace std;
                                                     Majority Element Index = 3
           int findMajority(int arr[], int n)
                                                     Process returned 0 (0x0) execution time: 0.268 s
     5
     6
               for(int i=0; i<n; i++)
                                                     Press any key to continue.
     8
                    int count = 1;
     9
                    for(int j=i+1; j<n; j++)
    10
                         if(arr[i] == arr[j])
    11
                              count++;
    12
                    if(count > n/2)
    13
                         return i;
    14
    15
               return -1;
    16
    17
    18
           int main()
    19
               int arr[] = \{2, 3, 4, 5, 5, 5, 5\}, n=7;
    20
               cout<<"Majority Element Index = "<<findMajority(arr, n)<<endl;</pre>
    21
    22
```

```
034) Majority Element.cpp X
         #include < iostream>
                                              Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \365-Days-of-Code\174). Complete DSA (Array[25-35])\_034)_Majority_Element.exe"
         using namespace std;
    3
                                             Majority Element Index = 3
         int findMajority(int arr[], int n)
       □ {
             int res = 0, count = 1;
             for(int i=1; i<n; i++)
                                             Process returned 0 (0x0) execution time: 0.262 s
    9
                 if(arr[res] == arr[i])
                                             Press any key to continue.
   10
                     count++;
  11
                 else
  12
                     count --:
  13
                 if(count == 0)
  14
  15
                     res=i;
  16
                    count = 1;
  17
  18
  19
   20
             count = 0;
  21
             for(int i=0; i<n; i++)
                 if(arr[res] == arr[i])
   22
   23
                    count++:
   24
             if(count \ll n/2)
   25
                 res = -1;
   26
  27
             return res;
   28
  29
   30
         int main()
  31
       = {
  32
             int arr[] = \{2, 3, 4, 5, 5, 5, 5\}, n=7;
             cout << "Majority Element Index = "<< find Majority (arr, n) << endl;
   33
   34
```

```
_035)_Minimum_Group_flips_to_make_sane.cpp X
           #include<iostream>
           using namespace std;
                                                              ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE '\365-Days-of-Code\174). Complete DSA (Array[25-35])\_...
                                                             From 2 to 4
           int minGroupFlipSame(int arr[], int n)
     6
                for(int i=1; i<n; i++)</pre>
                                                             Process returned 0 (0x0)
                                                                                                 execution time : 0.285 s
                                                             Press any key to continue.
     8
                    if(arr[i] != arr[i-1])
     9
    10
                         if(arr[i] != arr[0])
    11
                              cout<<"From "<<i<<" to ";
    12
                         else
    13
                              cout<<i-1<<endl;
    14
    15
    16
               if(arr[n-1] != arr[0])
    17
                    cout<<n-1<<endl;
    18
    19
    20
           int main()
    21
    22
                int arr[] = \{1, 1, 0, 0, 0, 1\}, n=6;
    23
               minGroupFlipSame(arr, n);
    24
```

```
#include<iostream>
      using namespace std;
      int maxSumOf K Element(int arr[], int k, int n)
 5
                                                  Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE "\365-Days-of-Code\175). Complete DSA (Array[36-...
 6
           int max sum=0;
                                                 Max Sum of 3 Consecutive Element = 45
           for(int i=0; i+k-1<n; i++)
 8
                                                 Process returned 0 (0x0) execution time: 0.297 s
 9
                int sum = 0;
                                                 Press any key to continue.
10
                for(int j=0; j<k; j++)
11
                     sum += arr[i+j];
12
                \max sum = \max(sum, \max sum);
13
14
           return max sum;
15
16
17
      int main()
18
           int arr[] = \{1, 8, 30, -5, 20, 7\}, k=3, n=6;
19
           cout<<"Max Sum of 3 Consecutive Element = "<<maxSumOf K Element(arr, k, n)<<end1;</pre>
20
```

036) Max Sum of K Consecutive Element.cpp X

```
_037)_Max_Sum_of_K_Consecutive_Element.cpp X
           #include<iostream>
           using namespace std;
           int maxSumOf K Element(int arr[], int k, int n)
     5
                                                             ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \365-Days-of-Code\175). Complete DSA (Arr...
     6
                int curr sum = 0;
                                                             Max Sum of 3 Consecutive Element = 45
                for (int i=0; i<k; i++)
     8
                    curr sum += arr[i];
                                                             Process returned 0 (0x0)
                                                                                        execution time: 0.483 s
     9
                int max sum = curr sum;
                                                             Press any key to continue.
    10
                for(int i=k; i<n; i++)
    11
    12
                    curr sum += (arr[i] - arr[i-k]);
    13
                    max sum = max (max sum, curr sum);
    14
    15
                return max sum;
    16
    17
    18
           int main()
    19
    20
                int arr[] = \{1, 8, 30, -5, 20, 7\}, k=3, n=6;
    21
                cout<<"Max Sum of 3 Consecutive Element = "<<maxSumOf K Element(arr, k, n)<<end1;</pre>
```

```
_038)_Find_SubArray_of_Given_Sum.cpp X
           #include<iostream>
                                                                     Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \365-Days-of-Code\175). Complete DSA (Array[36...
          using namespace std;
                                                                    Sum found between indexes 2 and 3
          int subArraySum(int arr[], int n, int given sum)
                                                                    Process returned 0 (0x0)
                                                                                                     execution time : 0.212 s
     6
               for(int i=0; i<n; i++)
                                                                    Press any key to continue.
     8
                    int sum = 0;
                    for(int j=i; j<n; j++)</pre>
    10
                        sum += arr[j];
    12
                        if(sum == given sum)
   13
                             cout<<"Sum found between indexes "<< i <<" and "<< j <<endl;</pre>
   14
   15
                             return true;
   16
    17
   18
    19
               cout<<"No subArray found"<<endl;</pre>
    20
    21
   22
          int main()
   23
    24
               int arr[] = {1, 4, 20, 3, 10, 5}, n=6, given sum=23;
               subArraySum(arr, n, given sum);
    26
```

```
039) Find SubArray of Given Sum.cpp X
          #include<iostream>
                                                                      ■ Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\365-Days-of-Code\175). Complete DSA (Array[36-44]... —
         using namespace std;
                                                                      Sum found between indexes 2 and 3
         int subArraySum(int arr[], int n, int given sum)
              int curr sum = arr[0], start=0;
                                                                      Process returned 0 (0x0)
                                                                                                            execution time : 0.218 s
              for (int end=1; end<n; end++)
                                                                      Press any key to continue.
    9
                  while(curr sum > given sum && start < end-1)</pre>
   10
   11
                      curr sum -= arr[start];
   12
                      start++;
   13
   14
                  if(curr sum == given sum)
   15
   16
                      cout<<"Sum found between indexes "<< start <<" and "<< end-1 <<end1;
   17
                      return true;
   18
   19
                  if (end<n)
   20
                      curr sum += arr[end];
   21
   22
              cout<<"No subArray found"<<endl;</pre>
   23
   24
   25
         int main()
   26
   27
              int arr[] = {1, 4, 20, 3, 10, 5}, n=6, given sum=23;
   28
              subArraySum(arr, n, given sum);
```

```
_040)_Print_N_bonacci_Numbers.cpp X
           #include<iostream>
           using namespace std;
                                                            Select "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE \'365-Days-of-Code\175). Complete DSA (Array[36-...
           void bonacciseries(long n, int m)
                                                           N-bonacci Numbers = 0 0 0 0 1 1 2 4 8 16 31 61 120 236 464
      5
      6
                cout<< "N-bonacci Numbers = " ;</pre>
                                                           Process returned 0 (0x0) execution time: 0.194 s
                int a[m] = { 0 };
                                                           Press any key to continue.
                a[n - 1] = 1;
      9
                a[n] = 1;
    10
                for (int i = n + 1; i < m; i++)
                     a[i] = 2 * a[i - 1] - a[i - n - 1];
    11
    12
                for (int i = 0; i < m; i++)
    13
                     cout << a[i] << " ";
    14
                cout << endl;
    15
    16
    17
           int main()
    18
    19
                int N = 5, M = 15;
    20
                bonacciseries (N, M);
    21
                return 0;
```

```
041) GetSum on Given Queries.cpp X
         #include<iostream>
                                                                "C:\Users\Akash Singh\Documents\Coding\' CHALLENGE "\365-Days-of-Code\175). Complete DSA (Array[36-44])\ 041) Get...
         using namespace std;
                                                                Sum of subarray index from 1 to 3 = 20
         int prefix sum[10000];
                                                                Sum of subarray index from 0 to 2 = 13
         void preSum(int arr[], int n)
                                                                Process returned 0 (0x0)
                                                                                                 execution time : 0.156 s
             prefix sum[0] = arr[0];
                                                                Press any key to continue.
    9
              for(int i = 1; i < n; i++)
   10
   11
                  prefix sum[i] = prefix sum[i - 1] + arr[i];
   12
   13
   14
   15
         int getSum(int prefix sum[], int 1, int r)
   16
   17
              if(1 != 0)
   18
                  return prefix sum[r] - prefix sum[l - 1];
   19
              else
                  return prefix sum[r];
   20
   21
   22
   23
   24
   25
         int main()
   26
   27
              int arr[] = \{2, 8, 3, 9, 6, 5, 4\}, n = 7;
   28
             preSum(arr, n);
   29
              cout<<"Sum of subarray index from 1 to 3 = "<<getSum(prefix sum, 1, 3)<<end1;</pre>
              cout<<"Sum of subarray index from 0 to 2 = "<<getSum(prefix sum, 0, 2)<<endl;
   30
   31
```

```
042) Find Equilibrium Point in an Array.cpp X
           #include<iostream>
          using namespace std;
          bool checkEquilibrium(int arr[], int n)
                                                             ■ Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE "\365-Days-of-Code\175). Complete DSA (Array[3...
     6
               for(int i = 0; i < n; i++)
     8
                    int 1 sum = 0, r sum = 0;
                                                            Process returned 0 (0x0)
                                                                                              execution time : 0.849 s
     9
                    for (int j = 0; j < i; j++)
                                                            Press any key to continue.
    10
                         l sum += arr[j];
                    for (int j = i + 1; j < n; j++)
    11
    12
                         r sum += arr[j];
    13
                    if(l sum == r sum)
    14
                         return true;
    15
    16
               return false;
    17
    18
    19
    20
    21
           int main()
    22
        - {
    23
               int arr[] = \{3, 4, 8, -9, 20, 6\}, n = 6;
    24
               cout<<checkEquilibrium(arr, n)? "true" : "false";</pre>
    25
               cout << endl;
    26
```

```
043) Find Equilibrium Point in an Array.cpp X
           #include<iostream>
          using namespace std;
                                                           Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE "\365-Days-of-Code\175). Complete DSA (Arra...
          bool checkEquilibrium(int arr[], int n)
               int sum = 0;
                                                          Process returned 0 (0x0) execution time: 0.456 s
               for(int i = 0; i < n; i++)
                                                          Press any key to continue.
                    sum += arr[i];
               int 1 sum = 0;
    10
               for(int i = 0; i < n; i++)
    11
    12
                    if(l sum == sum - arr[i])
    13
                         return true;
    14
                    l sum += arr[i];
    15
                    sum -= arr[i];
    16
    17
               return false;
    18
    19
    20
           int main()
    21
    22
               int arr[] = \{4, 2, 2\}, n = 3;
    23
               cout<<checkEquilibrium(arr, n)? "true" : "false";</pre>
    24
               cout << endl;
    25
```

```
044) In Two Array Find Max Occuring Element.cpp X
          #include<iostream>
                                                      Select "C:\Users\Akash Singh\Documents\Coding\" CHALLENGE "\365-Days-of-Code\175). Complete DSA (Array[36-44])\_044)_In_T...
          #include<bits/stdc++.h>
                                                     Maximum Occuring Element in both Array = 3
          using namespace std;
          int maxOcc(int L[], int R[], int n)
                                                     Process returned 0 (0x0) execution time: 0.347 s
                  int arr[1000];
                                                     Press any key to continue.
                  memset(arr, 0, sizeof(arr));
                  for(int i = 0; i < n; i++)
   10
                      arr[L[i]]++;
   12
                      arr[R[i] + 1] --;
   13
   14
                  int maxm = arr[0], res = 0;
   15
                  for(int i = 1; i < 1000; i++)
   16
   17
                      arr[i] += arr[i - 1];
   18
                      if(maxm < arr[i])</pre>
   19
   20
                           maxm = arr[i];
   21
                           res = i;
   22
   23
   24
                  return res;
   25
   26
   27
          int main()
   28
   29
              int L[] = \{1, 2, 3\}, R[] = \{3, 5, 7\}, n = 3;
              cout << "Maximum Occurring Element in both Array = "<< maxOcc(L, R, n) << endl;
   30
   31
```