

```
1  #include<iostream>
2  using namespace std;
3
4  int leaders(int arr[], int n)
5  {
6      for(int i=0; i<n; i++)
7      {
8          bool flag=false;
9          for(int j=i+1; j<n; j++)
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 {
24     int arr[] = {7, 10, 4, 3, 6, 5, 2}, n=7;
25     cout<<"Leaders = ";
26     leaders(arr, n);
27     cout<<endl;
28 }
```

Select "C:\Users\Akash Singh\Documents\Coding\CHALLENGE\Complete-Data-Structure-and-Algorithm\code\1). Array_017)_Leaders_in_an_Arra...

Leaders = 10 6 5 2

Process returned 0 (0x0) execution time : 0.101 s
Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int leaders(int arr[], int n)
5  {
6      int cur_leader = arr[n-1];
7      cout<<cur_leader<<" ";
8
9      for(int i=n-2; i>=0; i--)
10     {
11         if(cur_leader < arr[i])
12         {
13             cur_leader = arr[i];
14             cout<<cur_leader<<" ";
15         }
16     }
17 }
18
19 int main()
20 {
21     int arr[] = {7, 10, 4, 3, 6, 5, 2}, n=7;
22     cout<<"Leaders in Reverse Order = ";
23     leaders(arr, n);
24     cout<<endl;
25 }
```

Leaders in Reverse Order = 2 5 6 10

Process returned 0 (0x0) execution time : 0.284 s
Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int maxDiff(int arr[], int n)
5  {
6      int res = arr[1] - arr[0];
7      for(int i=0; i<n; i++)
8          for(int j=i+1; j<n; j++)
9              res = max(res, arr[j]-arr[i]);
10     return res;
11 }
12
13 int main()
14 {
15     int arr[] = {2, 3, 10, 6, 4, 8, 1}, n=7;
16     cout<<"Maximum Difference = "<<maxDiff(arr,n)<<endl;
17 }
```

Select "C:\Users\Akash Singh\Documents\Coding\CHALLENGE\Complete-Data-Structure-and-Algorithm\...

Maximum Difference = 8

Process returned 0 (0x0) execution time : 0.190 s

Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int maxDiff(int arr[], int n)
5  {
6      int res = arr[1]-arr[0], minVal = arr[0];
7      for(int i=1; i<n; i++)
8      {
9          res = max(res, arr[i]-minVal);
10         minVal = min(minVal, arr[i]);
11     }
12     return res;
13 }
14
15 int main()
16 {
17     int arr[] = {2, 3, 10, 6, 4, 8, 1}, n=7;
18     cout<<"Maximum Difference = "<<maxDiff(arr,n)<<endl;
19 }
```

Select "C:\Users\Akash Singh\Documents\Coding\CHALLENGE\Complete-Data-Structure-and-Algorithm\co...

Maximum Difference = 8

Process returned 0 (0x0) execution time : 0.201 s
Press any key to continue.


```
1  #include<iostream>
2  using namespace std;
3
4  int maxProfit(int price[], int start, int end)
5  {
6      if(end <= start )
7          return 0;
8      int profit = 0;
9      for(int i=start; i<end; i++)
10     {
11         for(int j=i+1; j<=end; j++)
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 {
27     int price[] = {1, 5, 3, 8, 12}, start=0, end = 5;
28     cout<<"Maximum Profit in Stock Market = "<<maxProfit(price, start, end)<<endl;
29 }
```

Maximum Profit in Stock Market = 13

Process returned 0 (0x0) execution time : 0.122 s
Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int maxProfit(int price[], int n)
5  {
6      int profit = 0;
7      for(int i=1; i<n; i++)
8          if(price[i] > price[i-1])
9              profit += (price[i]-price[i-1]);
10     return profit;
11 }
12
13 int main()
14 {
15     int price[] = {1, 5, 3, 8, 12}, n=5;
16     cout<<"Maximum Profit in Stock Market = "<<maxProfit(price, n)<<endl;
17 }
```

Select "C:\Users\Akash Singh\Documents\Coding\ CHALLENGE \Complete-Data-Structure-and-Algorith...

Maximum Profit in Stock Market = 13

Process returned 0 (0x0) execution time : 0.096 s
Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int getWater(int arr[], int n)
5  {
6      int res = 0;
7      for(int i=1; i<n-1; i++)
8      {
9          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++)
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)<<endl;
26 }
```

Select "C:\Users\Akash Singh\Documents\Coding\ CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1). Array_023)_Trap...

Maximum fill Boxes = 6

Process returned 0 (0x0) execution time : 0.076 s
Press any key to continue.


```
1  #include<iostream>
2  using namespace std;
3
4  int getWater(int arr[], int n)
5  {
6      int res=0;
7      int lmax[n], rmax[n];
8
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 = "<<getWater(arr, n)<<endl;
26 }
```

Select "C:\Users\Akash Singh\Documents\Coding\CHALLENGE\Complete-Data-Structure-and-Algorithm\code\1). Array...

Maximum fill Boxes = 6

Process returned 0 (0x0) execution time : 0.087 s
Press any key to continue.