

```
1  #include<iostream>
2  using namespace std;
3
4  int remDups(int arr[], int n)
5  {
6      int temp[n];
7      temp[0] = arr[0];
8      int res = 1;
9      for(int i=0; i<n; i++)
10     {
11         if(temp[res-1] != arr[i])
12         {
13             temp[res] = arr[i];
14             res++;
15         }
16     }
17     for(int i=0; i<n; i++)
18     {
19         arr[i] = temp[i];
20     }
21     return res;
22 }
23
24 int main()
25 {
26     int arr[] = {10, 20, 20, 30, 30, 30, 30}, n=7;
27     cout<<"Before Removal Duplicates: ";
28     for(int i=0; i<n; i++)
29     {
30         cout<<arr[i]<<" ";
31     }
32     cout<<endl<<"Size = "<<n<<endl;
33
34     int Res = remDups(arr, n);
35     cout<<"After Removal Duplicates: ";
36     for(int i=0; i<Res; i++)
37     {
38         cout<<arr[i]<<" ";
39     }
40     cout<<endl<<"Size = "<<Res<<endl;
41 }
```

Select "C:\Users\Akash Singh\Documents\Coding\ CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1). Array_9)_Remove_Duplicat...

Before Removal Duplicates: 10 20 20 30 30 30 30

Size = 7

After Removal Duplicates: 10 20 30

Size = 3

Process returned 0 (0x0) execution time : 0.170 s

Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int remDups(int arr[], int n)
5  {
6      int res = 1;
7      for(int i=1; i<n; i++)
8      {
9          if(arr[i] != arr[res-1])
10         {
11             arr[res] = arr[i];
12             res++;
13         }
14     }
15     return res;
16 }
17
18 int main()
19 {
20     int arr[] = {10, 20, 20, 30, 30, 30, 30}, n=7;
21     cout<<"Before Removal Duplicates: ";
22     for(int i=0; i<n; i++)
23     {
24         cout<<arr[i]<<" ";
25     }
26     cout<<endl<<"Size = "<<n<<endl;
27
28     int Res = remDups(arr, n);
29     cout<<"After Removal Duplicates: ";
30     for(int i=0; i<Res; i++)
31     {
32         cout<<arr[i]<<" ";
33     }
34     cout<<endl<<"Size = "<<Res<<endl;
35 }
```

Select "C:\Users\Akash Singh\Documents\Coding\ CHALLENGE \Complete-Data-Structure-and-Algorithm\code\1). Array_1...

Before Removal Duplicates: 10 20 20 30 30 30 30
Size = 7

After Removal Duplicates: 10 20 30
Size = 3

Process returned 0 (0x0) execution time : 0.123 s
Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int getLargest(int arr[], int n)
5  {
6      for(int i=0; i<n; i++)
7      {
8          bool flag = true;
9          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 = "<<getLargest(arr, n)<<endl;
27 }
```

Select "C:\Users\Akash Singh\Documents\Coding\CHALLENGE\Complete-Data-Structure-and-Algorithm\code\1). Array_11)_Largest_Numb...

Index = 2

Process returned 0 (0x0) execution time : 0.205 s
Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int getLargest(int arr[], int n)
5  {
6      int res=0;
7      for(int i=1; i<n; i++)
8          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;
16     cout<<"Index = "<<getLargest(arr, n)<<endl;
17 }
```

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

Index = 2

Process returned 0 (0x0) execution time : 0.207 s
Press any key to continue.


```
1  #include<iostream>
2  using namespace std;
3
4  int getLargest(int arr[], int n)
5  {
6      int largest=0;
7      for(int i=1; i<n; i++)
8          if(arr[i] > arr[largest])
9              largest = i;
10     return largest;
11 }
12
13 int getSecondLargest(int arr[], int n)
14 {
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;;
34 }
```

Index = 2

Process returned 0 (0x0) execution time : 0.119 s
Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  int getSecondLargest(int arr[], int n)
5  {
6      int res = -1, largest = 0;
7      for(int i=1; i<n; i++)
8      {
9          if(arr[i] > arr[largest])
10         {
11             res=largest;
12             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;
21 }
22
23 int main()
24 {
25     int arr[] = {10, 5, 18, 20}, n=4;
26     cout<<"Index = "<<getSecondLargest(arr, n)<<endl;;
27 }
```

Index = 2

Process returned 0 (0x0) execution time : 0.099 s
Press any key to continue.

```
1  #include<iostream>
2  using namespace std;
3
4  void moveToEnd(int arr[], int n)
5  {
6      for(int i=0; i<n; i++)
7      {
8          if(arr[i] == 0)
9          {
10             for(int j=i+1; j<n; j++)
11             {
12                 if(arr[j] != 0)
13                 {
14                     int temp = arr[i];
15                     arr[i] = arr[j];
16                     arr[j] = temp;
17                 }
18             }
19         }
20     }
21 }
22
23 int main()
24 {
25     int arr[] = {0, 8, 5, 0, 0, 10, 0, 20}, n=8;
26     cout<<"Before Move Zeros: ";
27     for(int i=0; i<n; i++)
28     {
29         cout<<arr[i]<<" ";
30     }
31     cout<<endl<<"Size = "<<n<<endl;
32
33     moveToEnd(arr, n);
34     cout<<"After Move Zeros: ";
35     for(int i=0; i<n; i++)
36     {
37         cout<<arr[i]<<" ";
38     }
39     cout<<endl<<"Size = "<<n<<endl;
40 }
41
```

Select "C:\Users\Akash Singh\Documents\Coding\CHALLENGE\Complete-Data-Structure-and-Algorithm\code\1). Array_15)_Move_All_ZEROs...

Before Move Zeros: 0 8 5 0 0 10 0 20
Size = 8

After Move Zeros: 20 10 5 8 0 0 0 0
Size = 8

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

```
1  #include<iostream>
2  using namespace std;
3
4  void moveToEnd(int arr[], int n)
5  {
6      int count = 0;
7      for(int i=0; i<n; i++)
8      {
9          if(arr[i] != 0)
10         {
11             int temp = arr[i];
12             arr[i] = arr[count];
13             arr[count] = temp;
14             count++;
15         }
16     }
17 }
18
19 int main()
20 {
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     {
25         cout<<arr[i]<<" ";
26     }
27     cout<<endl<<"Size = "<<n<<endl;
28
29     moveToEnd(arr, n);
30     cout<<"After Move Zeros: ";
31     for(int i=0; i<n; i++)
32     {
33         cout<<arr[i]<<" ";
34     }
35     cout<<endl<<"Size = "<<n<<endl;
36 }
```

Select "C:\Users\Akash Singh\Documents\Coding\CHALLENGE\Complete-Data-Structure-and-Algorithm\code\1). Array_16)_Move...

Before Move Zeros: 0 8 5 0 0 10 0 20
Size = 8

After Move Zeros: 8 5 10 20 0 0 0 0
Size = 8

Process returned 0 (0x0) execution time : 0.139 s
Press any key to continue.