



CodeClub

<Week 1, Session> Date: 04/09/2023

Questions to try for Discussion Session

→ Merge two Sorted arrays without using extra space.

- ◆ <https://practice.geeksforgeeks.org/problems/merge-two-sorted-arrays-1587115620/1>

→ Minimum Number of jumps to reach the end of the array.

- ◆ <https://leetcode.com/problems/jump-game-ii/>
- ◆ <https://practice.geeksforgeeks.org/problems/minimum-number-of-jumps-1587115620/1>

→ Also, try the questions done in the Coding session with Vector Data Structures instead of arrays (Even in your local IDEs).

- ◆ [Sum of Array Elements](#)
- ◆ [Max Min](#)
- ◆ [Reverse the Array](#)

Questions to try for self-practice

- Left Rotate an array by one place
- Linear Search
- Remove duplicates from the Sorted array
- Merge Intervals
- Trapping Rainwater problem

Alternatively, you can follow the links below

[A2Z Striver's DSA sheet \[Look for Array Questions\]](#)

<https://takeuforward.org/strivers-a2z-dsa-course/strivers-a2z-dsa-course-sheet-2/>

Preferred Playlist to follow:

https://youtube.com/playlist?list=PLgUwDviBIf0oF6QL8m22w1hIDC1vJ_BHz&si=uT38ZbtmJYoxHfwh


Important Announcements

- Everyone is requested to be on time for the Coding sessions, i.e. at 6 PM every Monday (Coding Session) and Thursday (Discussion).
- Due to the festival of Janmashtami, the Discussion session for this Thursday is rescheduled for Friday, i.e., September 8, 2023.

Thanks and Regards, Team CodeClub!



Lecture - 01

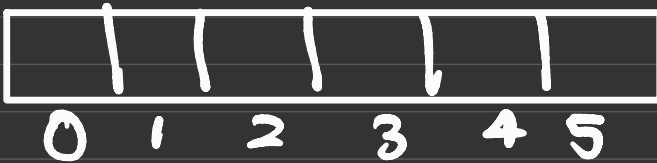


Arrays.

→ Data Structure



int arr[] ⇒ float arr[]
↓
arr[6]



* Access

```
for (0 - n) {  
    cout << arr[i] << endl;  
    cin >> arr[i];  
}
```

* Vectors

→ Dynamically Size
→ allocate
→ Operations

```
int arr [ ]
```

```
vector <int> temp (6, 0)
```



```
vector <int> vect;
```

```
vect.push_back (5);  
" " " " (7);
```



```
(9)  
(10) ←
```



vect.size() - 1
↔ 4

vect.pop-back()

①

4	3	2	1
---	---	---	---

Sum
for (0 - n) {

Sum += arr[i];

}

return Sum;

* Man Min

-2	5	1	3	-1	2
----	---	---	---	----	---

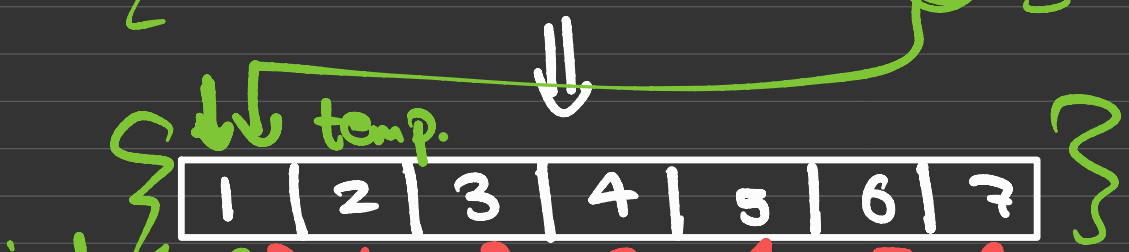
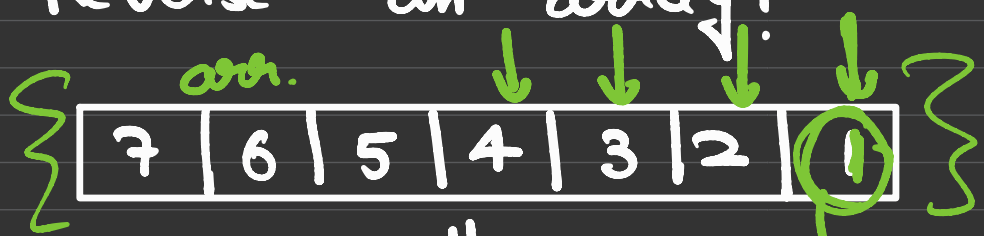
maximum = INT_MIN

-5	-4	-3	-2	-1	-6
----	----	----	----	----	----

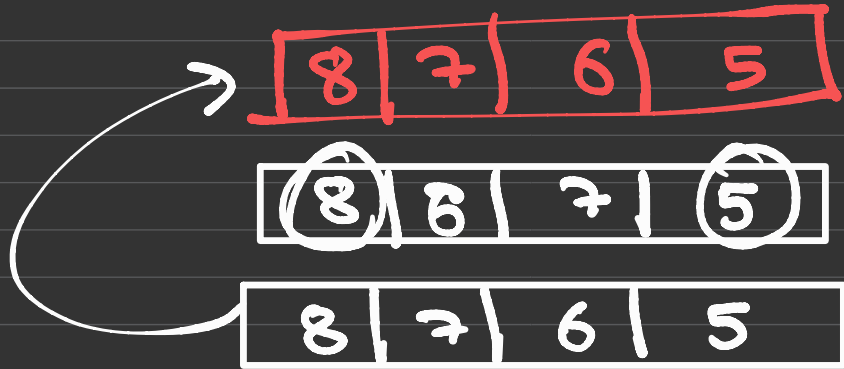
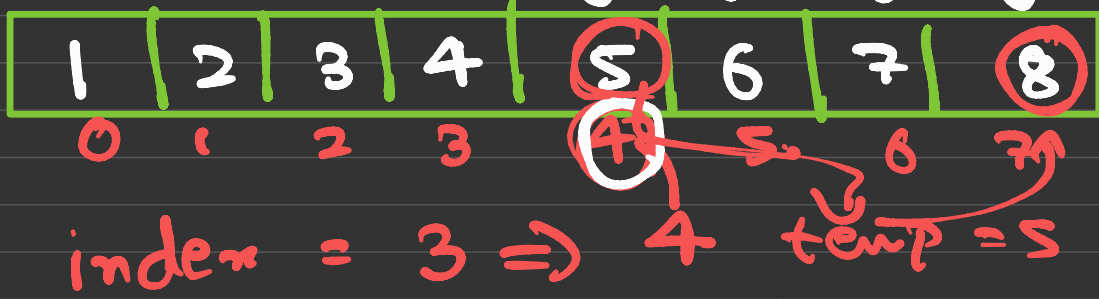
limits. ← ⊖ 214 .. ←

minimum = INT_MAX
↓
214 ..

* Reverse an array!



```
index = 0 1 2 3 4 5 6  
for (i = n-1; i >= 0; i--)  
    temp[index] = arr[i];  
    index++;  
}
```

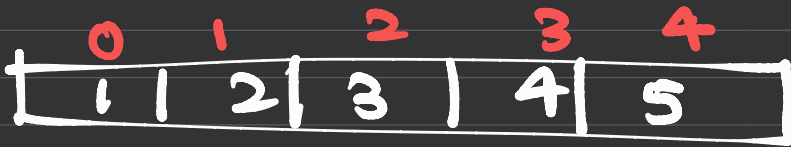
$$\text{low} = \text{mid} + 1$$

$$\text{high} = n - 1$$

↑
Size

while (low <= high) {

}



$$\text{vect.size() - 1} \Rightarrow 5 - 1$$

↓
4

↔ 4

