C++
ARRAY-2

Lecture-12

Raghav Garg

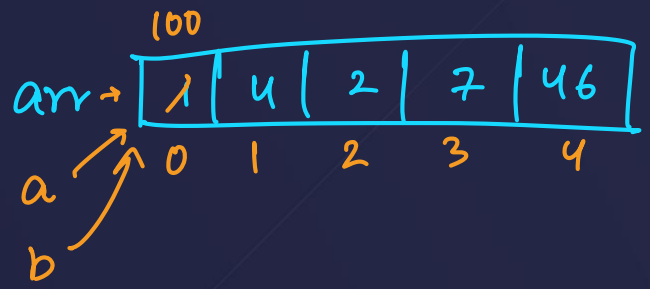
Today's checklist

1) Passing array to functions
2) ~~Dynamic allocation~~ → will be covered in a seperate video
3) Vectors in C++
4) Operations on Vector
5) Problem on arrays and Two pointers

Passing Array to Functions

→ reference

```
void display(int a[]){
    for(int i=0;i<=4;i++){
        cout<<a[i]<<" ";
    }
    cout<<endl;
    return;
}
void change(int b[]){
    b[0] = 100;
}
int main(){
    int arr[5] = {1,4,2,7,46};
    // accessing the elements o
    // updation, pass by value
    display(arr);
    change(arr);
    display(arr);
}
```



1 4 2 7 46

MCQ : When you pass an array as an argument to a function, what actually gets passed?

1. address of the array
2. values of the elements of the array
3. address of the first element of the array
4. number of elements of the array

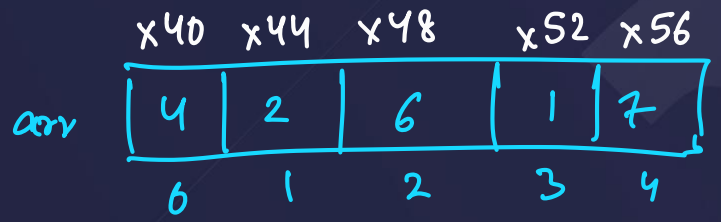
Arrays and Pointers

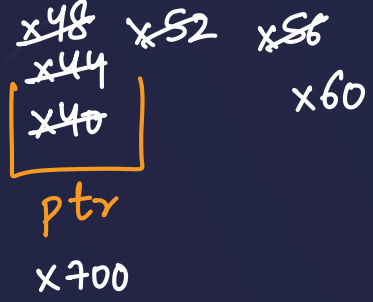
```
char arr[3] = {'a', '2', '$'};

int arr[] = {1, 5, 2, 3, 4};
int* ptr = arr;        ✓
int* ptr = &arr;       X
int* ptr = &arr[0];    ✓
int* ptr = arr[0];     X
```

```
int x = 4;
int* ptr = &x;         ✓
int* ptr = x;          X
```

Arrays and Pointers

```
int arr[] = {4,2,6,1,7};
int* ptr = arr; // giving address
for(int i=0;i<=4;i++){
    cout<<*ptr<<" ";
    ptr++;
}
```





Output

4 2 6 1 7

Vector in C++ → Dynamic Array

↓

array ki replacement

↓

problem → fixed size

```
int arr[5] = {1,2,3,4,5};
```

```
{1,2,3,4,5,6};
```

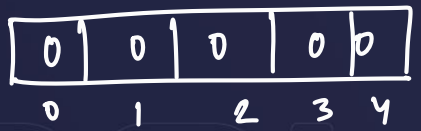
```
vector<int> arr;
```

Basic Operations on Vectors

- Syntax
- push_back, pop_back, size, capacity, at, sort

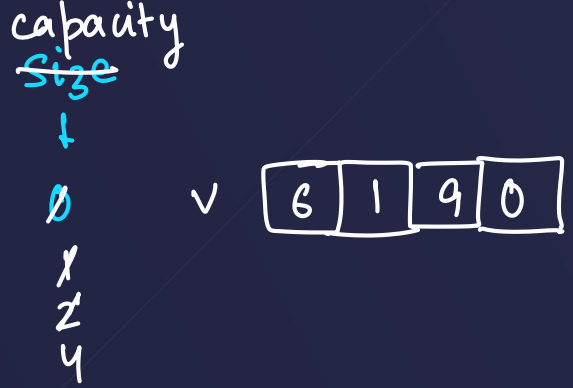
```
vector<int> V(5);
```

V



Basic Operations on Vectors

```
vector<int> v; //
v.push_back(6);
v.push_back(1);
v.push_back(9);
v.push_back(0);
```



Basic Operations on Vectors

```
vector<int> v; // y
// inserting / inpu
v.push_back(6);
// v.push_back(1);
v[1] = 1;
v.push_back(9);
v.push_back(10);

// if you want to u
cout<<v[0]<<" ";
cout<<v[1]<<" ";
cout<<v[2]<<" ";
cout<<v[3]<<" ";
```

Output

6 1 9 10 α

6 9 10 0

Garbage

Passing vectors to Functions : Diff. to Arrays

Vectors are passed by value . Each time you pass , new vector is created.

& → ampersant

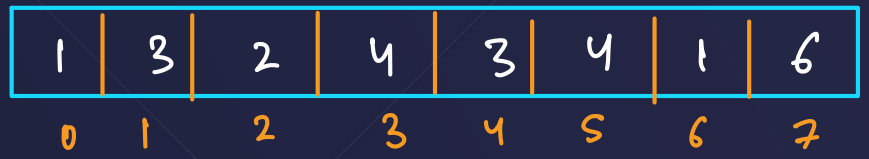
Looping in vector

index

Ques : Find the last occurrence of x in the array.

```
int x;        vector<int> v;
```

v



x = 1

```
int idx = -1;
```

Ques : Find the doublet in the Array whose sum is equal to the given value x. (LeetCode - 1) (Two Sum)

x = 7

i

target

(0,7)

(1,3)

(1,5)

(3,4)

(4,5)

(6,7)

```cpp
int x;
cout<<"Enter target : ";
cin>>x;
vector<int> v;
int n;
cout<<"Enter array size : ";
cin>>n;

cout<<"Enter array elements : ";
for(int i=0;i<n;i++){
    int q;
    cin>>q;
    v.push_back(q);
}

for(int i=0;i<=v.size()-2;i++){
    for(int j=i+1;j<=v.size()-1;j++){
        if(v[i]+v[j]==x){
            cout<<"("<<i<<","<<j<<")"<<endl;
        }
    }
}
```

v.size() = 8

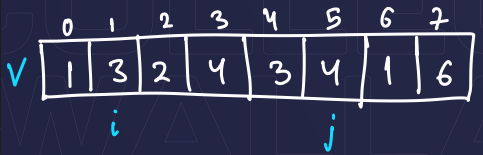
Enter target : 7

Enter array size : 8

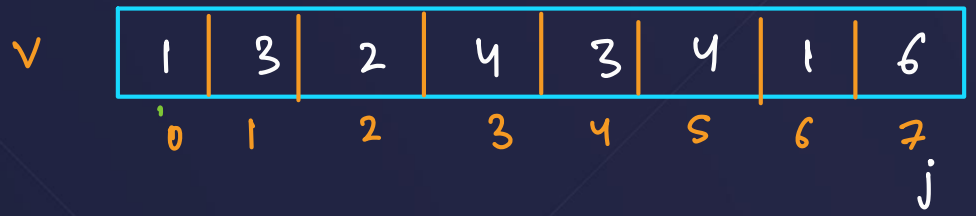
Enter array elements :

1 3 2 4 3 4 1 6

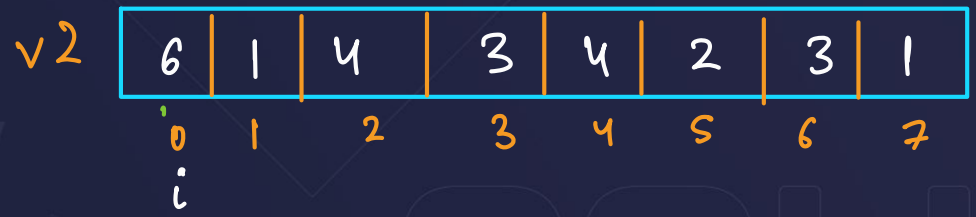
- (0, 7)
- (1, 3)
- (1, 5)
-



Ques : Write a program to copy the contents of one array into another in the reverse order.



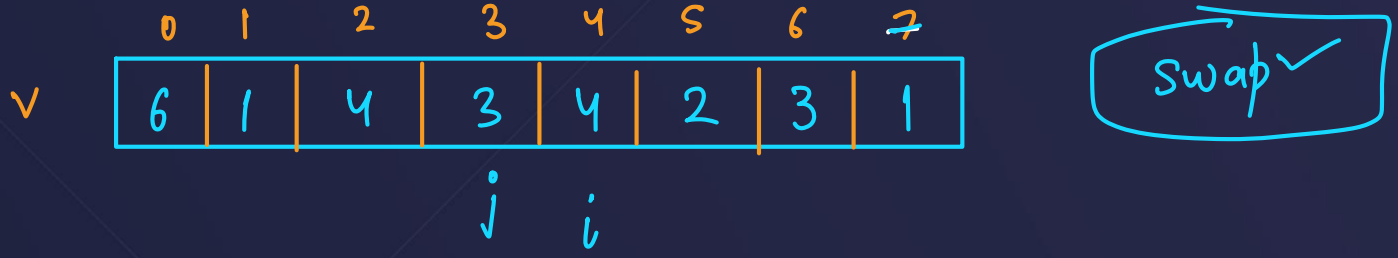
$i + j = size - 1$




```
vector<int> v2(v.size());
```

Two Pointers

Ques : Write a program to reverse the array without using any extra array.

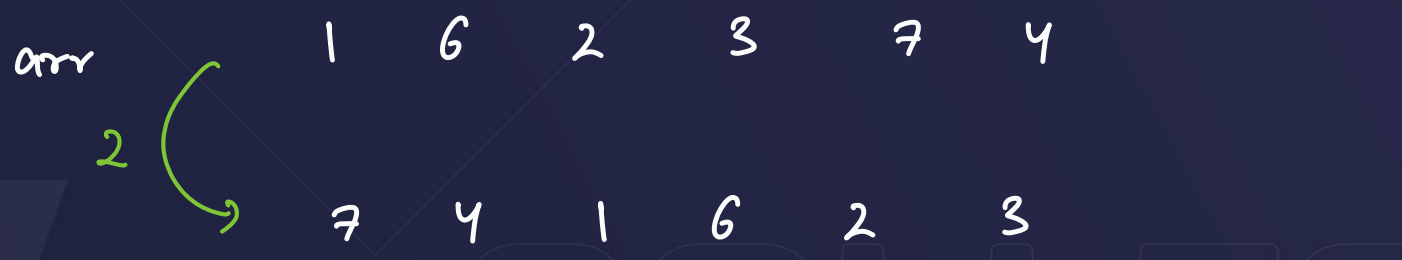


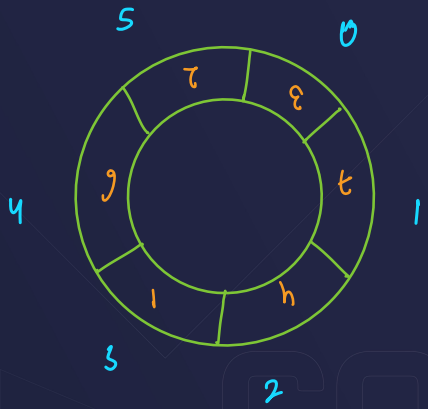

```
int i = 0;
int j = v.size() - 1;

i >= j
```

Reverse part of array

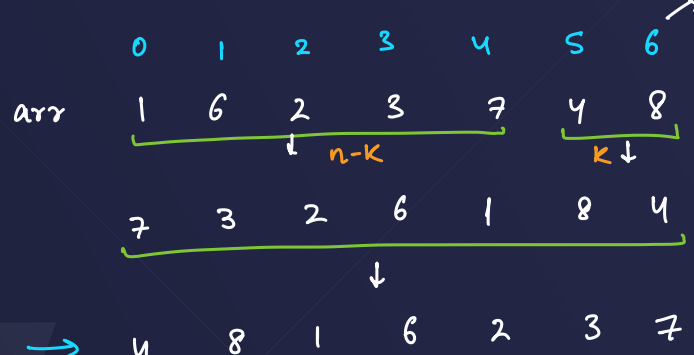
Ques : Rotate the given array ‘a’ by k steps, where k is non-negative.

Note : k can be greater than n as well where n is the size of array ‘a’.




0
1
2
3
4
5
6
7
8
9
10
11

Algorithm : Hint → reverse part of array



Step-1 : reversePart (0, n-k-1, v); int n = v.size();
reverse Part (n-k, n-1, v);
reversePart (0, n-1, v);

0	1	2	3	4	5	6
1	6	2	3	7	4	8

$n = 7$

$K = 9$

4	8	1	6	2	3	7

if $(K > n)$

$K = K \% n$

if $(K = n)$ → array is same

Homework : Leetcode → Rotate Array

THANK YOU