HOMEWORK DAY: 08

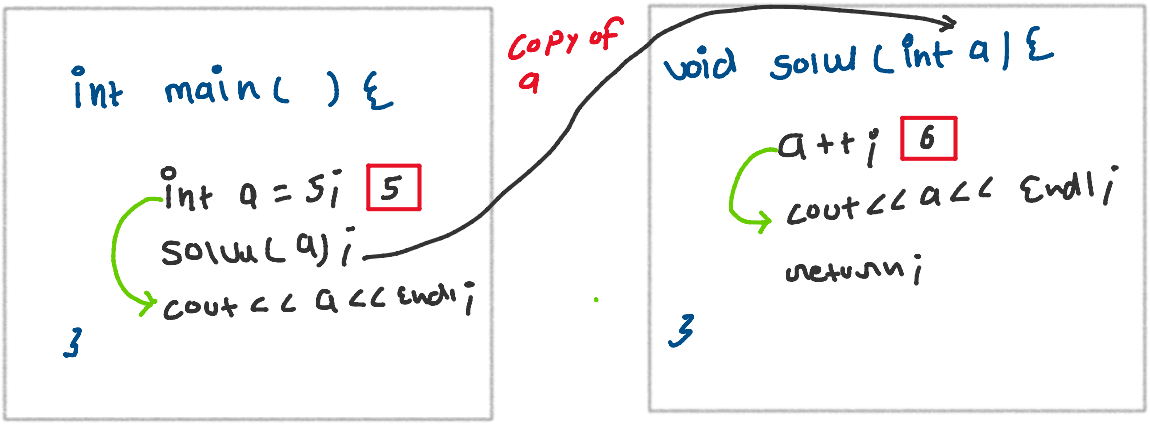
09 September 2023 14:04

class 08 — 08/09/2023

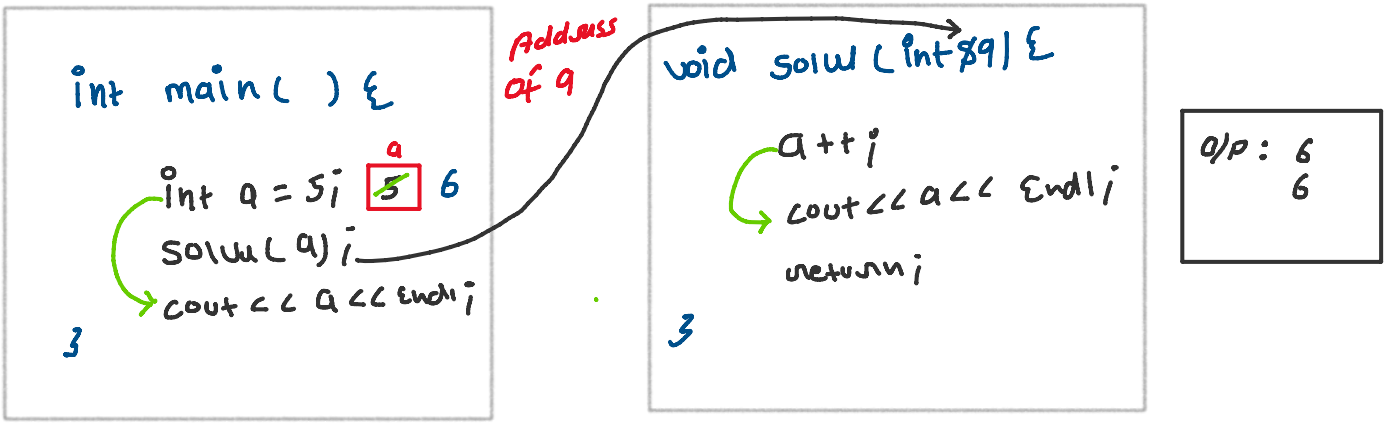
https://www.linkedin.com/in/manojoffcialmj/

ARRAY LEVEL-2

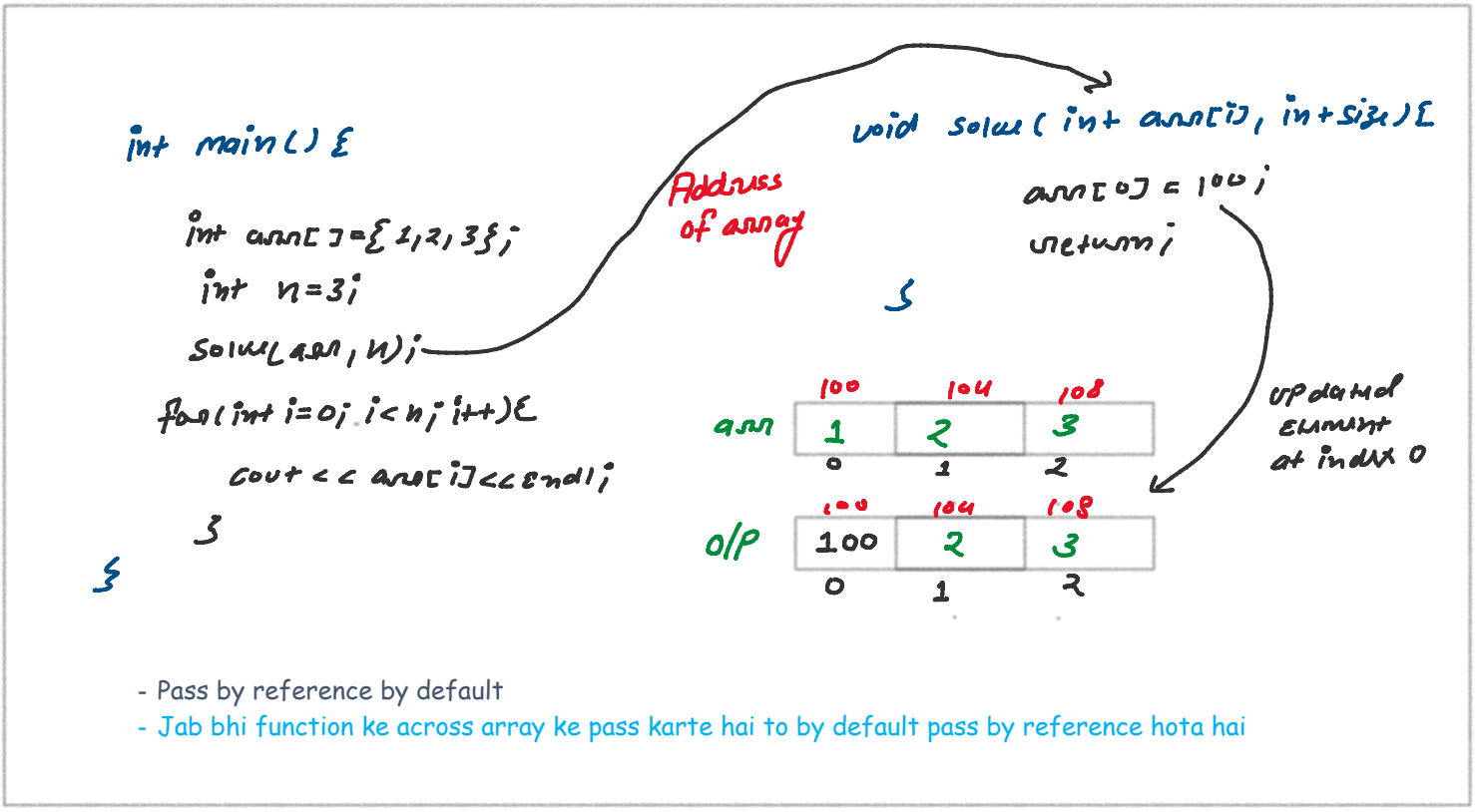
Function call by value:



Function call by reference:

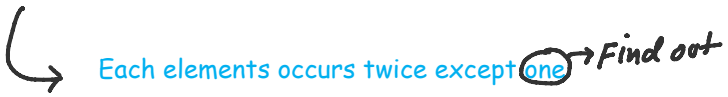


Array and function call by reference:

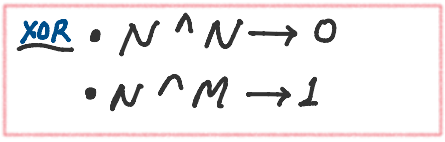


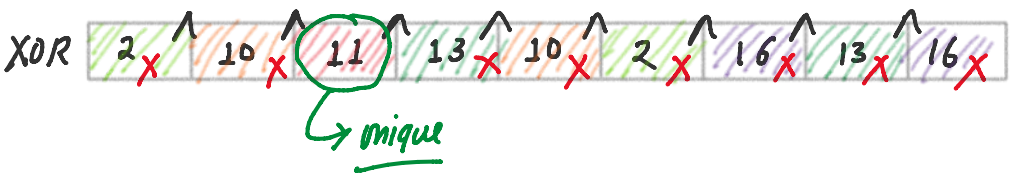
- Pass by reference by default
- Jab bhi function ke across array ke pass karte hai to by default pass by reference hota hai

Program 01: Find unique element



Output 11



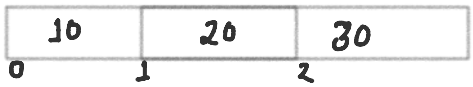


```cpp
// Program 01: Find unique element
#include<iostream>
using namespace std;

// Find unique element
int getUnique(int arr[],int size){
    int ans = 0;
    for(int i=0;i<size;i++){
        ans ^=arr[i];
    }
    return ans;
}

int main(){
    int arr[]={2,10,11,13,10,2,16,13,16};
    int n=9;

    // calling method
    int finalAns = getUnique(arr,n);
    cout<<"Unique element: "<<finalAns;

    return 0;
}

/*
Input: [2,10,11,13,10,2,16,13,16]
Output: Unique element: 11
*/
```

@manojofficialmj

Program 02: Print all pairs

Input



Output

(10,10) (20,10) (30,10)
(10,20) (20,20) (30,20)
(10,30) (20,30) (30,30)

DRY RUN

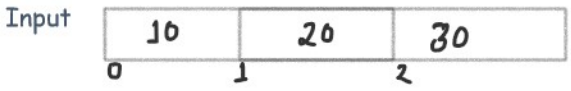
i	j	Print	arr[i], arr[j]
0	0		10 10
	1		10 20
	2		10 30
1	0		20 10
	1		20 20
	2		20 30
2	0		30 10
	1		30 20
	2		30 33
3			

END

```cpp
// Program 02: Print all pairs
#include<iostream>
using namespace std;

// Print all pairs
void PrintAllPairs(int arr[],int size){

    // outer loop
    for(int i=0;i<size;i++){
        // inner loop
        for(int j=0;j<size;j++){
            cout<<arr[i]<<", "<<arr[j]<<endl;
        }
    }
}

int main(){
    int arr[]={10,20,30};
    int n=3;

    // calling method
    PrintAllPairs(arr,n);

    return 0;
}

/*
Input: [10,20,30]
Output:
10, 10
10, 20
10, 30
20, 10
20, 20
20, 30
30, 10
30, 20
30, 30
*/
```

@manojofficialmj

Program 03: Print all triplets



Output

(10,10,10)	(20,10,10)	(30,10,10)
(10,10,20)	(20,10,20)	(30,10,20)
(10,10,30)	(20,10,30)	(30,10,30)
(10,20,10)	(20,20,10)	(30,20,10)
(10,20,20)	(20,20,20)	(30,20,20)
(10,20,30)	(20,20,30)	(30,20,30)
(10,30,10)	(20,30,10)	(30,30,10)
(10,30,20)	(20,30,20)	(30,30,20)
(10,30,30)	(20,30,30)	(30,30,30)

DRY RUN

i	j	k	arr[i], arr[j], arr[k]		
0	0	0	10	10	10
		1	10	10	20
		2	10	10	30

10, 20, 30

10, 20, 30

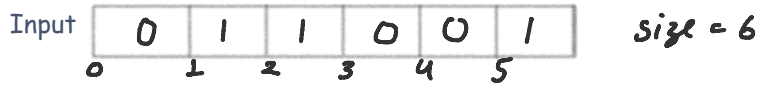
		1	10	10	20
		2	10	10	30
	1	0	10	20	10
		1	10	20	20
		2	10	20	30
	2	0	10	30	10
		1	10	30	20
		2	10	30	30
1	0	0	20	10	10
		1	20	10	20
		2	20	10	30
	1	0	20	20	10
		1	20	20	20
		2	20	20	30
	2	0	20	30	10
		1	20	30	20
		2	20	30	30
2	0	0	30	10	10
		1	30	10	20
		2	30	10	30
	1	0	30	20	10
		1	30	20	20
		2	30	20	30
	2	0	30	30	10
		1	30	30	20
		2	30	30	30

(3) → END

```cpp
// Print all triplets
void printAllTriplets(int arr[],int size){

    // outer loop
    for(int i=0;i<size;i++){
        // inner loop of i
        for(int j=0;j<size;j++){
            for(int k=0;k<size;k++){
                cout<<arr[i]<<", "<<arr[j]<<", "<<arr[k]<<endl;
            }
        }
    }
}
```

@manojofficialmj

Program 04: Print 0's and 1's



① Counting 0's and 1's Solution

```
// Counting 0's and 1's solotion
void sortZeroOne1(int arr[],int size){
    int zeroCount = 0;
    int oneCount = 0;

    // Step 01: Counting 0's and 1's
    for(int i=0;i<size;i++){
        if(arr[i]==0){
            zeroCount++;
        }
        else{
            oneCount++;
        }
    }

    // Step 02: Fill 0's
    for(int i=0;i<zeroCount;i++){
        arr[i]=0;
    }
    // Step 03: Fill 1's
    for(int i=zeroCount;i<size;i++){
        arr[i]=1;
    }
}
```

Brute FORCE For

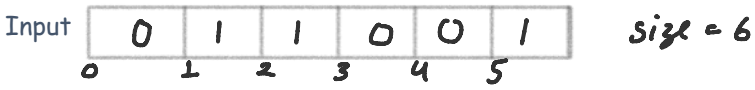
@manojofficialmj

```
// Counting 0's and 1's solotion
void sortZeroOne1(int arr[],int size){
    int zeroCount = 0;
    int oneCount = 0;

    // Step 01: Counting 0's and 1's
    for(int i=0;i<size;i++){
        if(arr[i]==0){
            zeroCount++;
        }
        else{
            oneCount++;
        }
    }

    // Easy Way
    int i=0;
    while (zeroCount--)
    {
        arr[i]=0;
        i++;
    }
    while (oneCount--)
    {
        arr[i]=1;
        i++;
    }

}
```

Brute FORCE While

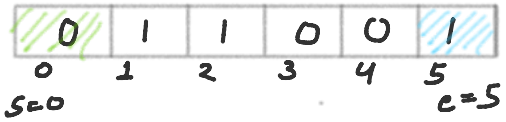
@manojofficialmj

② Two pointer approach



start → 0
End → 1

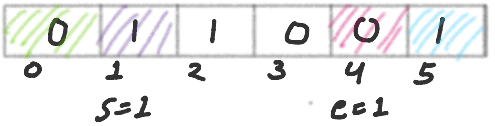
situation:01



① arr[s] = 0 s++

② arr[e] = 1 e--

situation:02



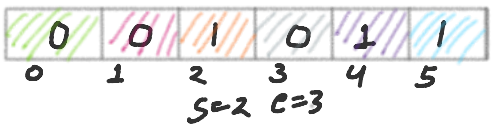
③ arr[s] = 1 && arr[e] = 0

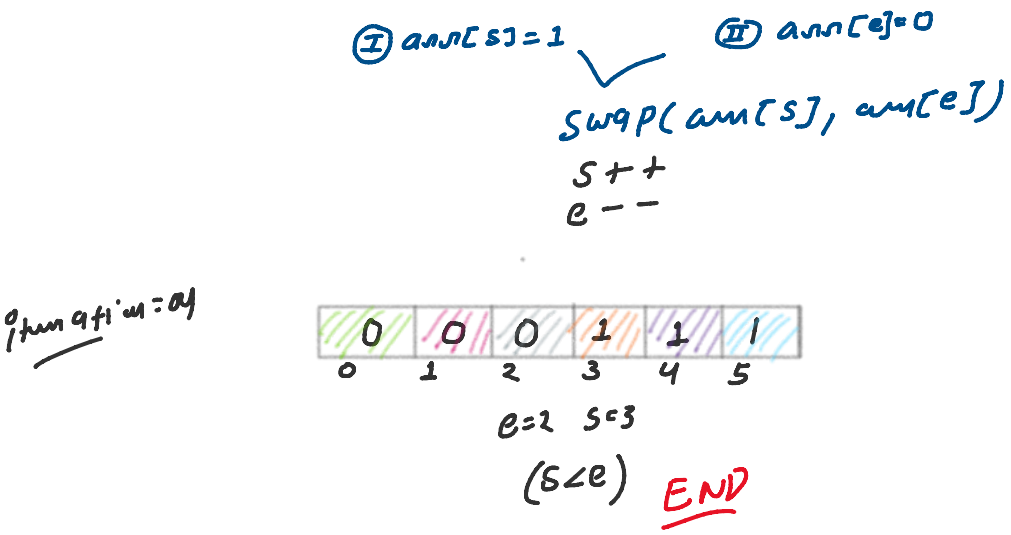
swap(arr[s], arr[e])

s++

e--

situation:03






```java
// Counting 0's and 1's solotion
void sortZeroOne1(int arr[],int size){
    int s = 0;
    int e = size-1;

    while (s<e)
    {
        // When arr[s]==0 then increase only s++
       while(s<e && arr[s]==0){
        s++;
       }
        // When arr[e]==1 then decrease only e--
       while(s<e && arr[e]==1){
        e--;
       }
        // When arr[s]==1 and arr[e]==0 then
        //increase and decrease respectively s++, e-- and swap
        if(s<e && arr[s]==1 && arr[e]==0){
            swap(arr[s],arr[e]);
            s++;
            e--;
        }
    }
}
```

@manojofficialmj

HOMEWORK
↳ next PDF
main mil jayegi
☺ MANOJ KUMAR
Thanks