Q Find unique element in an array.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

XOR → X-OR

| a | b | a∧b |
|---|---|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

a∧0 ⟹ ⓐ
0∧a =⟩ ⓐ

```
int ans = 0;
for ( int i = 0 ; i < arr.size() ; i++) {
    ans = ans ^ arr[i];
}
return ans;
```

$1 \wedge 2 \wedge 3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 6 \wedge 5 \wedge 4 \wedge 3 \wedge 2 \wedge 1$

$(1 \wedge 1) \wedge (2 \wedge 2) \wedge (3 \wedge 3) \wedge (4 \wedge 4) \wedge (5 \wedge 5) \wedge (6 \wedge 6)$
$\wedge 7$

$0 \wedge 7 \rightarrow \boxed{7}$ . unique element.

Dry Run.

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

↑
i

ans = 0

$arr[0] = 1$

$\boxed{ans} = ans \wedge arr[i]$

| 0 | 1 |

$\rightarrow ①$

$i = 2$

$\boxed{ans = 1 \wedge 2}$

$i = 3$

$ans = 1 \wedge 2 \wedge 3$

$i = 4 \quad ans = 1 \wedge 2 \wedge 3 \wedge 4$

---

**Q** Sort 0's & 1's in an Array.

| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

output → 

| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

int i = 0, pointer.     traverse.

| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
↑
i

for ( i → arr. size )

    if ( arr[i] == 0 ) {

        count 0 ++;

| int count0 = 0; |
| int count1 = 0; |

$$\}$$

else if ( arr [ i ] == 1) Count 1 ++;

$$\}$$



Count 1 = 4

Count 0 = 4

| O | O | O | O | | | | | |
|---|---|---|---|---|---|---|---|---|

$$\{ \text{for ( int } i = 0; \quad i < \underline{\text{Cont } 0}, \quad i + +) \{$$

$$\quad\quad \text{arr} [ i ] = 0;$$

$$\}$$

```cpp
#include<iostream>
using namespace std;
int main(){
    int arr[]={0,1,0,1,0,1,0,1,1,1};
    int n=10;
    int count1=0;
    int count0=0;
    for(int i=0;i<n;i++){
        if(arr[i]==0)count0++;
        else if(arr[i]==1)count1++;
    }
    cout<<"count1 is : "<<count1<<endl;
    cout<<"count0 is : "<<count0;
    cout<<endl;
    cout<<"[ ";
    for(int i=0;i<count0;i++){
        cout<<0<<",";
    }
    for(int j=0;j<count1;j++){
        cout<<1<<",";
    }
    cout<<" ]";
}
```