Q. Given an array.
Find the no. of Special indexes!
↳ after deleting that element

> Sum of all even indexed elements = Sum of all odd indexed elements

Eg.

	0 ✓	1	2 ✓	3	4	5
A:	4	3	2	7	6	-2

del (0)

	0	1	2	3	4
A:	3	2	7	6	-2

$S_o = 8$
$S_e = 8$ ✓

del (1)

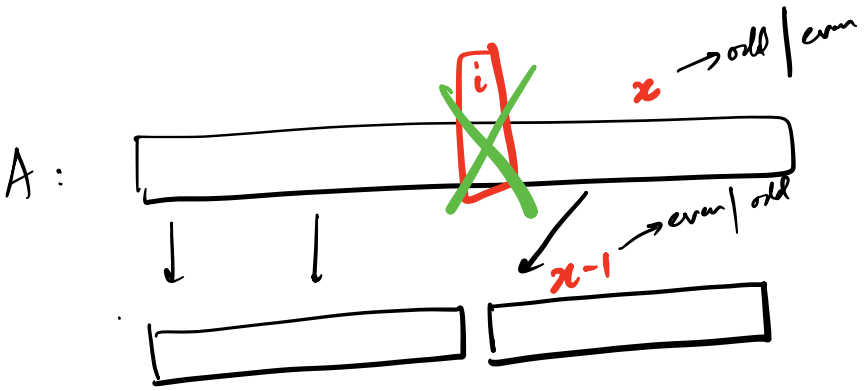
	0	1	2	3	4
A:	4	2	7	6	-2

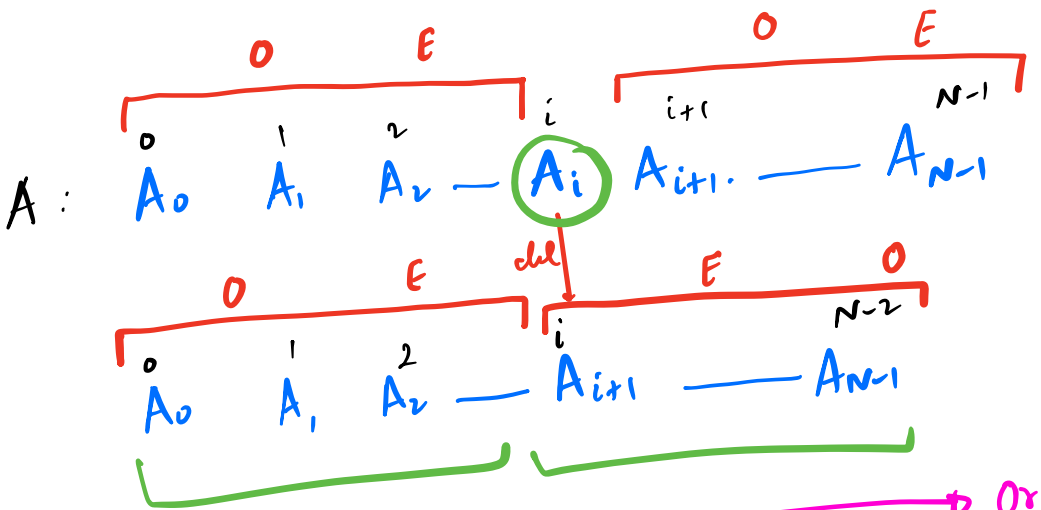
$S_o = 8$
$S_e = 9$ ✗

del (2)

	0	1	2	3	4
A:	4	3	7	6	-2

$S_o = 9$
$S_e = 9$ ✓





Original Array

S_e = $S_e[0, i-1]$ + $S_o[i+1, N-1]$

S_o = $S_o[0, i-1]$ + $S_e[i+1, N-1]$

Idea: Create PSodd , PSeven.

$S_e = Se[0, i-1] + So[i+1, N-1]$

$$S_e = PS_e[i-1] + PS_o[N-1] - PS_o[i]$$

$S_o = So[0, i-1] + Se[i+1, N-1]$

$$S_o = PS_o[i-1] + PS_e[N-1] - PS_e[i]$$

STEPS

1. Calc. PS_o, PS_e from A[]. — O(N)

2. cnt = 0 — O(N)

```
for(i=0; i<N; i++){
    Se = (i-1<0 ? 0 : PSe[i-1]) + PSo[N-1] - PSo[i];
    So = (i-1<0 ? 0 : PSo[i-1]) + PSe[N-1] - PSe[i];
    if(So == Se){
        cnt++;
    }
}
return cnt;
```

TC = O(N)

SC = O(N)

	0	1	2	3	4	5
A:	4	3	2	7	6	-2
PSe =	4	4	6	6	12	12
PSo =	0	3	3	10	10	8

i

del (1)

$S_o = PS_o[i-1] + PS_e[N-1] - PS_e[i]$

$= 0 + 12 - 4 = 8$

$S_e = PS_e[i-1] + PS_o[N-1] - PS_o[i]$

$= 4 + 8 - 3 = 9$) X

i

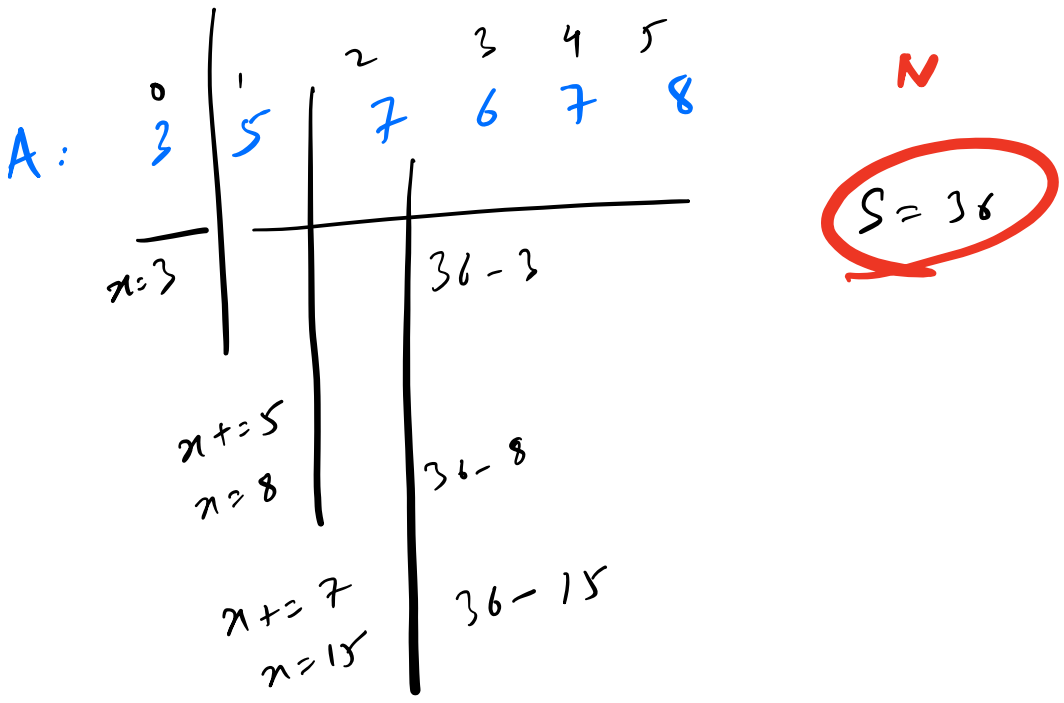
del (0)

$S_o = PS_o[i-1] + PS_e[N-1] - PS_e[i]$

$0 + 12 - 4 = 8$

$S_e = PS_e[i-1] + PS_o[N-1] - PS_o[i]$

$0 + 8 - 0 = 8$ ✓



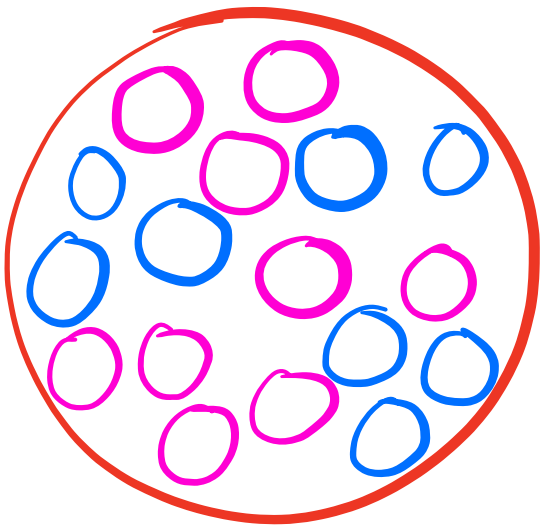
HW → Do it in O(1) space!

Q Given a Cicular table.

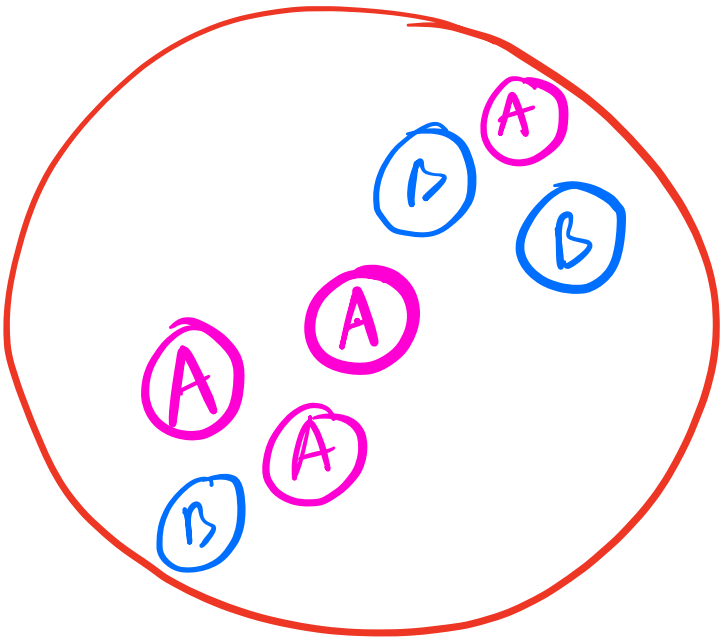
2 players A & B A play first.

they play in turn

they have ∞ supply of cicular disks of same size (size disk < size table)



If both of them play optimally, who will WIN?



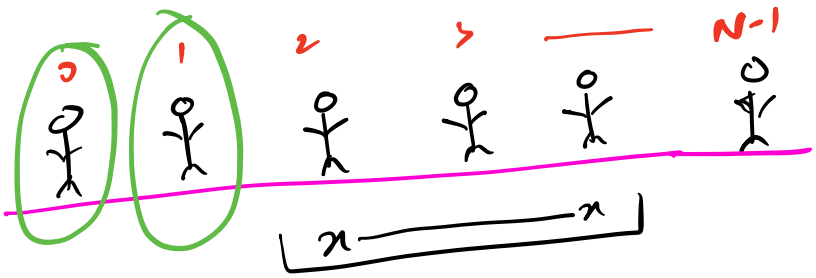
A would ALWAYS WIN

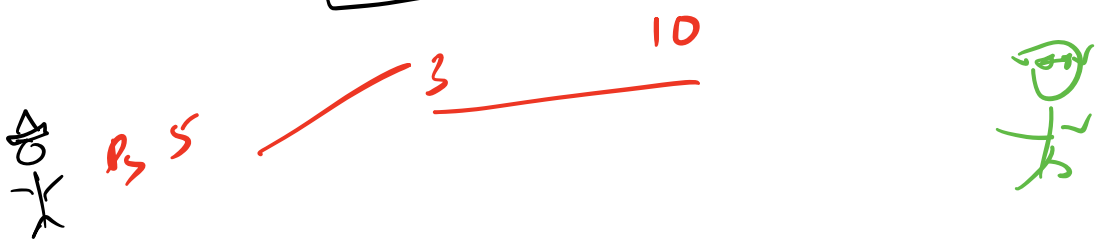
→ 1st place in CETER

After that copy cat B's move!

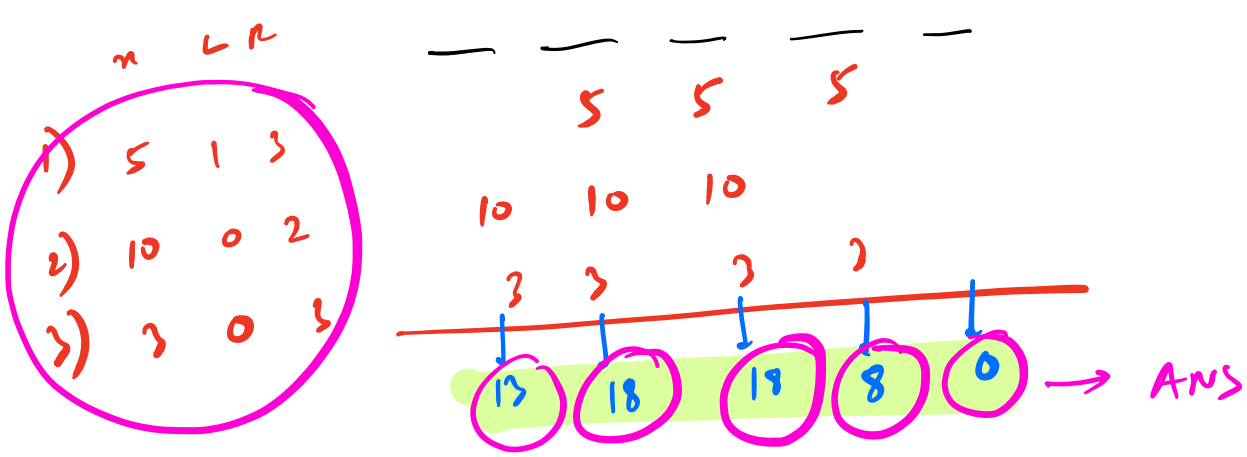
Given N beggars & M worshippers

Every worshipper would give Rs x to ALL beggars in [L, R]





find the money will ALL beggars at the end!

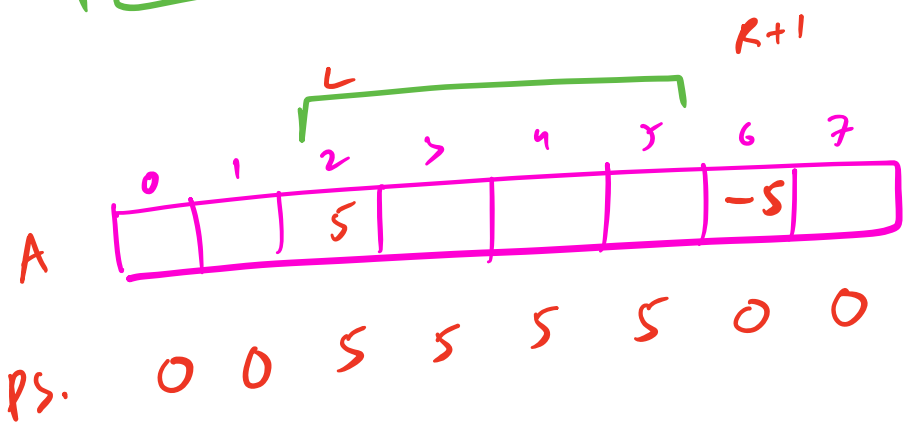


```
int A[N] = {0};
for( i=0; i<M; i++) {
    // m, L, R;
    for( k=L; k<=R; k++) {
        A[k] += m;
    }
}
ret A[];
```

TC : O(NM)

SC : O(1)

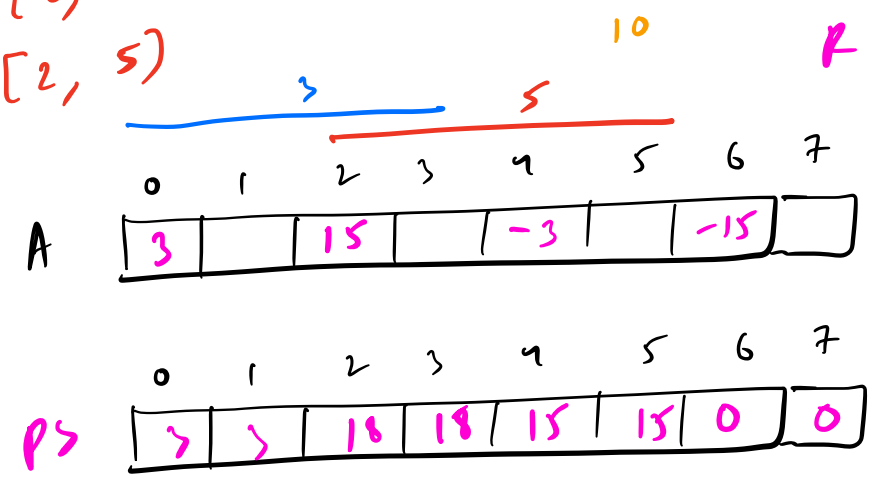
+5 [2, 5]



+5 [2, 5]

+3 [0, 3]

+10 [2, 5)



Idea:

+x [L, R]

A[L] += x → R+1 < N ✓

A[R+1] -= x

do this for all Q queries!

Create PS[]. → ANS

```
int A[N] = {0};
for( i=0; i<M; i++) {          ⟶ M
    // x, L, R
    A[L] += x;
    if(R+1 < N) {
        A[R+1] -= x;
    }
}
for( i=1; i<N; i++) {          ⟶ N
    A[i] += A[i-1];
}
ret A[];
```

TC: O(N+M)

SC: O(1)

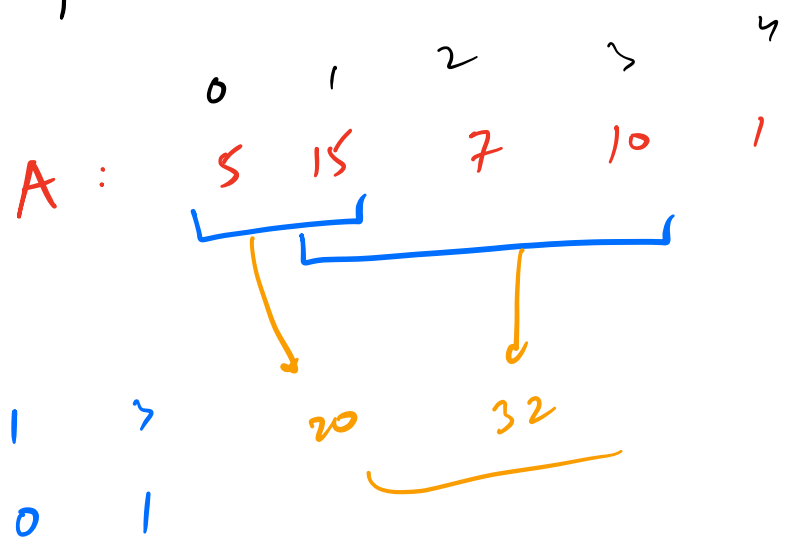
Q Given an array A of N elements.

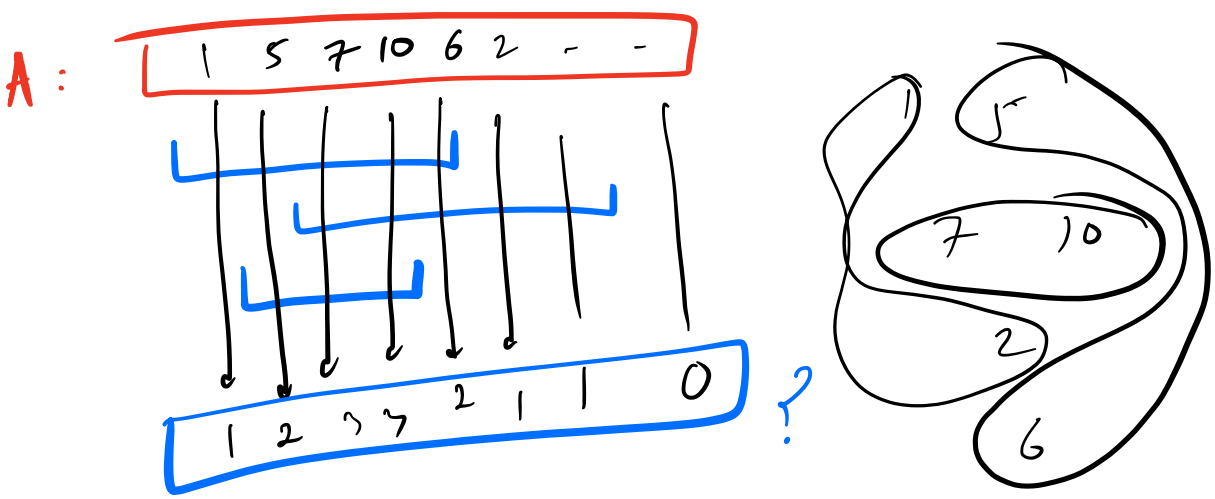
Given Q queries [L, R]

ANS of a query → Sum of all elements in the index range [L, R].

You can rearranging the elements of the array!

GOAL: to obtain the MAX Sum of all queries!





Idea: put the largest element at most frequently covered index under query!