



Maximum points you can obtain from cards

arr = [6 2 3 4 7 2 1 7 1] K=4





$lsum = \cancel{0} 15$	$rsum = 0$	Sum 15
$lsum = 11$	$rsum = 1$	12
$lsum = 8$	$rsum = 8$	<div>16</div>
$lsum = 6$	$rsum = 9$	15
$lsum = 0$	$rsum = 11$	11





		Sum
$lsum = \cancel{15}$	$rsum = 0$	15

$lsum = 11$	$rsum = 1$	12
-------------	------------	----

$lsum = 8$	$rsum = 8$	<div>16</div>
------------	------------	---------------

$lsum = 6$	$rsum = 9$	15
------------	------------	----

$lsum = 0$	$rsum = 11$	11
------------	-------------	----





?

```
lsum = 0, rsum = 0, maxsum = 0  
for (i = 0 → k-1) lsum = lsum + nums[i];  
maxsum = lsum;
```

```
        rindex = n-1  
        for (i = k-1; i >= 0; i--)  
        {  
            lsum = lsum - nums[i];  
            rsum =
```





```
for ( i = 0 ; i < n ; i++ ) sum = sum + nums[i] ;  
maxSum = sum ;
```

```
rindex = n - 1
```

```
for ( i = k - 1 ; i >= 0 ; i-- )  
{
```

```
    sum = sum - nums[i] ;
```

```
    sum = sum + nums[rindex] ;
```

```
    rindex = rindex - 1 ;
```



arr = [6 2 3 4 7 2 1 7 1] K=4

Annotations: A bracket is drawn over the first four elements [6, 2, 3, 4]. The elements 3 and 4 are crossed out with 'X'. The elements 7 and 1 at the end are checked with '✓'. Arrows point to the 7 and 1 with the label 'n-1'.

```
func (nums, k)
```

```
{
    sum = 0, rsum = 0, maxsum = 0
```

```
    for (i = 0 → k-1) {sum = sum + nums[i];
        maxsum = sum;
```

```
        rinden = n-1
```

```
        for (i = k-1 ; i >= 0 ; i--)
```

```
        {sum = sum - nums[i];
```

```
        rsum = rsum + nums[rinden];
```

```
        rinden = rinden - 1;
```

```
        maxsum = max(maxsum, sum + rsum);
```

```
}
```





```
lsum = rsum + nums[rinden];  
rinden = rinden - 1;
```

```
    msum = max(msum, lsum + rsum);  
}  
return msum;
```





```
func (nums, k)
```

```
{  
    sum = 0, rsum = 0, maxsum = 0  
    O(k) ← for (i = 0 → k-1) { sum = sum + nums[i];  
        maxsum = sum;
```

```
        rindex = n-1  
    O(k) ← for (i = k-1; i >= 0; i--)
```

```
{  
        sum = sum - nums[i];  
        rsum = rsum + nums[rindex];  
        rindex = rindex - 1;
```

```
        maxsum = max(maxsum, sum + rsum);  
    }
```

```
    return maxsum;  
}
```

TC → $O(2 \times k)$
SC → $O(1)$

THANKS FOR WATCHING!

Don't forget to subscribe



TAKE U FORWARD