

<https://course.acciojob.com/idle?question=00ad64c7-02b1-4306-bbf7-8e925ea1e978>

## //problem

- EASY
- Max Score: 30 Points



# Max Sum Subarray of size K

---

Given an array of integers Arr of size N and a number, K. Return the maximum sum of a subarray of size K

## Input

The first line contains 2 integers N and K The second line contains N integers denoting elements of the array

## Output

Print an integer denoting the maximum sum subarray of size K

## Example 1

Input:

4 2

100 200 300 400

Output:

700

Explanation: The sum of the last 2 elements is maximum i.e. (0-based indexing)  $\text{Arr}[2] + \text{Arr}[3] = 700$  is maximum

## Example 2

Input:

4 2

100 -200 300 -400

Output:

100

Explanation: Sum of  $\text{Arr}[1] + \text{Arr}[2] = ((-200) + 300) = 100$  which is the maximum sum possible for subarray of size 2

Constraints

$1 \leq N \leq 1000000$

$1 \leq K \leq N$

$-10000 \leq \text{Arr}[i] \leq 10000$

### Topic Tags

- 2-Pointers
- Arrays

# My code

```
// in java
import java.util.*;
import java.lang.*;
import java.io.*;

public class Main
{
    public static void main (String[] args) throws
java.lang.Exception
    {
        //your code here
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        int arr[]=new int[n];
        int c=0; int p=0;
        int m=s.nextInt();
        for(int i=0;i<n;i++)
            arr[i]=s.nextInt();

        for(int j=0;j<=n-m;j++){
            int count=0;
            for(int k=j;k<j+m;k++)
            {
                //System.out.print(arr[k]+ " ");
                count+=arr[k];
            }
        }
    }
}
```

```
//System.out.print("\n");  
if(c==0&&p==0) c=count;  
if(count>c) c=count;  
}  
System.out.print(c);  
}  
}
```