

Kadane's Algorithm

Given an array **Arr[]** of **N** integers. Find the contiguous sub-array(containing at least one number) which has the maximum sum and return its sum.

Input:

`N = 5`

`Arr[] = {1,2,3,-2,5}`

Output:

`9`

Explanation:

Max subarray sum is 9

of elements (1, 2, 3, -2, 5) which is a contiguous subarray.

Example 2:

Input:

`N = 4`

`Arr[] = {-1,-2,-3,-4}`

Output:

`-1`

Explanation:

Max subarray sum is -1
of element (-1)

Find first and last positions of an element in a sorted array

Given a sorted array with possibly duplicate elements, the task is to find indexes of first and last occurrences of an element x in the given array.

```
Input : arr[] = {1, 3, 5, 5, 5, 5, 67, 123, 125}  
x = 5
```

```
Output : First Occurrence = 2  
Last Occurrence = 5
```

```
Input : arr[] = {1, 3, 5, 5, 5, 5, 7, 123, 125 }  
x = 7
```

```
Output : First Occurrence = 6  
Last Occurrence = 6
```

Search in a sorted and rotated array

Given a sorted and rotated array A of N distinct elements which is rotated at some point, and given an element key. The task is to find the index of the given element key in the array A.

Example 1:

Input:

N = 9

A[] = {5, 6, 7, 8, 9, 10, 1, 2, 3}

key = 10

Output:

5

Explanation: 10 is found at index 5.

Example 2:

Input:

N = 4

A[] = {3, 5, 1, 2}

key = 6

Output:

-1

Allocate minimum number of pages

You are given **N** number of books. Every *i*th book has **A_i** number of pages.

You have to allocate contiguous books to **M** number of students. There can be many ways or permutations to do so. In each permutation, one of the **M** students will be allocated the maximum number of pages. Out of all these permutations, the task is to find that particular permutation in which the maximum number of pages allocated to a student is **minimum** of those in all the other permutations and print this minimum value.

Each book will be allocated to exactly one student. Each student has to be allocated at least one book.

Input:

N = 4

A[] = {12,34,67,90}

M = 2

Output:

113

Explanation:

Allocation can be done in following ways:

{12} and {34, 67, 90} Maximum Pages = 191

{12, 34} and {67, 90} Maximum Pages = 157

{12, 34, 67} and {90} Maximum Pages = 113

Therefore, the minimum of these cases is 113, which is selected as the output.

Input: [10, 20, 10, 30], M = 2

Output: 40