

- 1) A public multi storied parking lot which has no monitoring systems installed has cars being parked in it on a daily basis. A management of the parking lot believes that there are more number of cars being parked in the parking lot than the number of spots available. Given the entry and exit times for all the cars that use the parking lot, can you help find out the maximum number of cars being parked at any given point in time.

Example Input & Output:

Input: arr[] = {9:00, 9:40, 9:50, 11:00, 15:00, 18:00}

dep[] = {9:10, 12:00, 11:20, 11:30, 19:00, 20:00}

Output: 3

Input: arr[] = {9:00, 9:40}

dep[] = {9:10, 12:00}

Output: 1

- 2) Given an array arr of size N and an integer K, the task is to find the maximum for each and every contiguous subarray of size K.

Example Input & Output:

Input: arr[] = {1, 2, 3, 1, 4, 5, 2, 3, 6}, K = 3

Output: 3 3 4 5 5 6

Input: arr[] = {8, 5, 10, 7, 9, 4, 15, 12, 90, 13}, K = 4

Output: 10 10 10 15 15 90 90

- 3) Given two integer arrays A and B of size N.

There are N gas stations along a circular route, where the amount of gas at station i is A[i].

You have a car with an unlimited gas tank and it costs B[i] of gas to travel from station i

to its next station (i+1). You begin the journey with an empty tank at one of the gas stations.

Return the minimum starting gas station's index if you can travel around the circuit once, otherwise return -1.

You can only travel in one direction. i to i+1, i+2, ... n-1, 0, 1, 2.. Completing the circuit means starting at i and ending up at i again.

Example Input & Output:

Input 1:

A = [1, 2]

B = [2, 1]

Output 1:

1

Explanation 1:

If you start from index 0, you can fill in A[0] = 1 amount of gas. Now your tank has 1 unit of gas. But you need B[0] = 2 gas to travel to station 1.

If you start from index 1, you can fill in A[1] = 2 amount of gas. Now your tank has 2 units of gas. You need B[1] = 1 gas to get to station 0. So, you travel to station 0 and still have 1 unit of gas left over. You fill in A[0] = 1 unit of additional gas, making your current gas = 2. It costs you B[0] = 2 to get to station 1, which you do and complete the circuit.

4) Given an array `arr[]` of size `N` and a number `K`, the task is to find the length of the smallest subsequence such that the sum of the subsequence is greater than or equal to number `K`.

Input: `arr[] = {2, 3, 1, 5, 6, 3, 7, 9, 14, 10, 2, 5}`, `K = 35`

Output: 4

Smallest subsequence with the sum greater than or equal to the given sum `K` is {7, 9, 14, 10}

Input: `arr[] = {1, 2, 2, 2, 3, 4, 5, 4, 7, 6, 5, 12}`, `K = 70`

Output: -1

Subsequence with sum greater than equal to the given sum is not possible.