NibilaAmutha L2 Task

1. Given an array arr[] of N non-negative integers representing the height of blocks. If width of each block is 1, compute how much water can be trapped between the blocks during the rainy season.

Input:

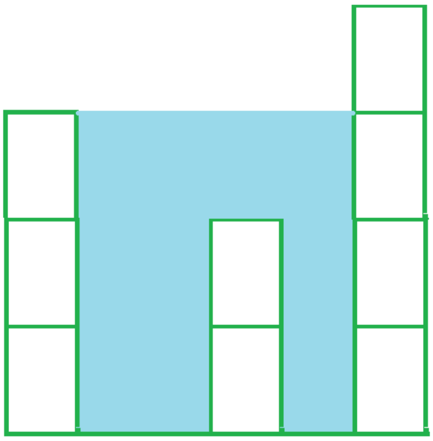
N = 6

arr[] = {3,0,0,2,0,4}

Output:

10

Explanation:



Bars for input {3, 0, 0, 2, 0, 4}
Total trapped water = 3 + 3 + 1 + 3 = 10

Input:

N = 4

arr[] = {7,4,0,9}

Output:

10

Explanation:

Water trapped by above block of height 4 is 3 units and above block of height 0 is 7 units. So, the total unit of water trapped is 10 units.

Input:

N = 3

arr[] = {6,9,9}

Output:

0

Explanation: No water will be trapped.

Your Task: You don't need to read input or print anything. The task is to complete the function trappingWater() which takes arr[] and N as input parameters and returns the total amount of water that can be trapped.

Expected Time Complexity: O(N) **Expected Auxiliary Space: O(N)**

Constraints:

$3 \leq N \leq 10\text{^}^{6}$ $0 \leq A_{i} \leq 10\text{^}^{8}$

2. Given a **sorted** array **arr[]** of distinct integers. Sort the array into a wave-like array(In Place). In other words, arrange the elements into a sequence such that arr[1] >= arr[2] <= arr[3] >= arr[4] <= arr[5]..... If there are multiple solutions, find the lexico graphically smallest one.

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

Output: 2 1 4 3 5

Explanation: Array elements after sorting it in wave form are 2 1 4 3 5.

Input:
n = 6
arr[] = {2,4,7,8,9,10}

Output: 4 2 8 7 10 9

Explanation: Array elements after sorting it in wave form are 4 2 8 7 10 9.

Your Task:

The task is to complete the function **convertToWave**(), which converts the given array to a wave array.

NOTE: The given array is sorted in ascending order, and you don't need to return anything to make changes in the original array itself.

Expected Time Complexity: O(n).

Expected Auxiliary Space: O(1).

Constraints:

1 ≤ n ≤ 10^^{6} 0 ≤ arr[i] ≤10^^{7}

3. Given an integer array **coins[ ] of size N** representing different denominations of currency and an integer **sum**, find the number of ways you can make **sum** by using different combinations from coins[ ].
Note: Assume that you have an infinite supply of each type of coin.

Input:
sum = 4 ,
N = 3
coins[] = {1,2,3}

Output: 4

Explanation: Four Possible ways are:{1,1,1,1},{1,1,2},{2,2},{1,3}.

Input:
Sum = 10 ,
N = 4
coins[] ={2,5,3,6}

Output: 5

Explanation: Five Possible ways are:{2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5} and {5,5}.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **count()** which accepts an array coins**[ ] its size N and sum** as input parameters and returns the number of ways to make change for given sum of money.

Expected Time Complexity: O(sum*N)

Expected Auxiliary Space: O(sum)

Constraints:

1 <= sum, N <= 10

4. Union of two arrays can be defined as the common and distinct elements in the two arrays. Given two sorted arrays of size **n** and **m** respectively, find their union.

Input:

n = 5, arr1[] = {1, 2, 3, 4, 5}

m = 3, arr2 [] = {1, 2, 3}

Output: 1 2 3 4 5

Explanation: Distinct elements including both the arrays are:1 2 3 4 5

Input:

n = 5, arr1[] = {2, 2, 3, 4, 5}

m = 5, arr2[] = {1, 1, 2, 3, 4}

Output: 1 2 3 4 5

Explanation: Distinct elements including both the arrays are: 1 2 3 4 5

Input:

n = 5, arr1[] = {1, 1, 1, 1, 1}

m = 5, arr2[] = {2, 2, 2, 2, 2}

Output: 1 2

Explanation: Distinct elements including both the arrays are: 1 2.

Your Task:

You do not need to read input or print anything. Complete the **function findUnion()** that takes two arrays **arr1[]**, **arr2[],** and their size **n and m** as input parameters and returns a list containing the **union of the two arrays.**

Expected Time Complexity: O(n+m).

Expected Auxiliary Space: O(n+m).

Constraints:

1 <= n, m <= 10^5

1 <= arr[i], brr[i] <= 10^6

5. Given an array containing **N** integers and an integer **K.**, Your task is to find the length of the longest Sub-Array with the sum of the elements equal to the given value **K.**

Input :
A[] = {10, 5, 2, 7, 1, 9}
K = 15
Output : 4
Explanation: The sub-array is **{5, 2, 7, 1}**.

Input :
A[] = {-1, 2, 3}
K = 6
Output : 0
Explanation: There is no such sub-array with sum 6.

Your Task:
This is a function problem. The input is already taken care of by the driver code. You only need to complete the function **lenOfLongSubarr()** that takes an array **(A)**, sizeOfArray **(n)**, sum **(K)**and **returns** the required length of the longest Sub-Array. The driver code takes care of the printing.

Expected Time Complexity: O(N).
Expected Auxiliary Space: O(N).

Constraints:
1<=N<=10^5
-10^5<=A[i], K<=10^5

6.Given two sorted arrays of distinct elements. There is only 1 difference between the arrays. First array has one element extra added in between. Find the index of the extra element.
Input:
N = 7
A[] = {2,4,6,8,9,10,12}
B[] = {2,4,6,8,10,12}
Output: 4
Explanation: In the second array, 9 is missing and it's index in the first array is 4.
Input:
N = 6
A[] = {3,5,7,9,11,13}
B[] = {3,5,7,11,13}

Output: 3

Your Task: You don't have to take any input. Just complete the provided function **findExtra**() that takes array A[], B[] and size of A[] and return the valid index (**0 based indexing**).

Expected Time Complexity: O(log N).

Expected Auxiliary Space: O(1).

Constraints:

2<=N<=10^4

1<=Ai<=10^5

7. Given an array of distinct elements. Find the third largest element in it. Suppose you have A[] = {1, 2, 3, 4, 5, 6, 7}, its output will be 5 because it is the 3 largest element in the array A.

Input:

N = 5

A[] = {2,4,1,3,5}

Output: 3

Input:

N = 2

A[] = {10,2}

Output: -1

Your Task: Complete the function **thirdLargest()** which takes the array a[] and the size of the array, n, as input parameters and returns the third largest element in the array. It return -1 if there are less than 3 elements in the given array.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(1)

Constraints:

1 ≤ N ≤ 10^5

1 ≤ A[i] ≤ 10^5

8. Given an array **Arr** of size **N** containing positive integers. Find the maximum sum of a sub sequence such that no two numbers in the sequence should be adjacent in the array.

Input:

N = 6

Arr[] = {5, 5, 10, 100, 10, 5}

Output: 110

Explanation: If you take indices 0,3 and 5,then Arr[0]+Arr[3]+Arr[5]=5+100+5=110

Input:

N = 4

Arr[] = {3, 2, 7, 10}

Output: 13

Explanation:3 and 10 forms a non continuous sub sequence with maximum sum.

Your Task: You don't need to read input or print anything. Your task is to complete the function **findMaxSum()** which takes the array of integers **arr** and **n** as parameters and returns an integer denoting the answer.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(1)

Constraints:

$1 \le N \le 10^6$

$1 \le Arr_i \le 10^7$

9. Given two sorted arrays array1 and array2 of size **m** and **n** respectively. Find the median of the two sorted arrays.

Input:

m = 3, n = 4

array1[] = {1,5,9}

array2[] = {2,3,6,7}

Output: 5

Explanation: The middle element for {1,2,3,5,6,7,9} is 5

Input:

m = 2, n = 4

array1[] = {4,6}

array2[] = {1,2,3,5}

Output: 3.5

Your Task: The task is to complete the function **MedianOfArrays()** that takes array1 and array2 as input and returns their median.
Can you solve the problem in expected time complexity?

Expected Time Complexity: O(min(log n, log m)).

Expected Auxiliary Space: O((n+m)/2).

Constraints:

$0 \le m,n \le 10^4$

$1 \le array1[i], array2[i] \le 10^5$

10. Given two arrays **A** and **B** contains integers of size **N** and **M**, the task is to find numbers which are present in the first array, but not present in the second array.

Input: N = 6, M = 5

A[] = {1, 2, 3, 4, 5, 10}

B[] = {2, 3, 1, 0, 5}

Output: 4 10

Explanation: 4 and 10 are present in first array, but not in second array.

Input: N = 5, M = 5
A[] = {4, 3, 5, 9, 11}
B[] = {4, 9, 3, 11, 10}
Output: 5
Explanation: Second array does not contain element 5.

Your Task: This is a function problem. You don't need to take any input, as it is already accomplished by the driver code. You just need to complete the function **findMissing**() that takes array **A,** array **B,** integer **N,** and integer **M** as parameters and returns an array that contains the missing elements and the order of the elements should be the same as they are in array A.

Expected Time Complexity: O(N+M).
Expected Auxiliary Space: O(M).

Constraints:
$1 \le N, M \le 10^6$
$-10^6 \le A[i], B[i] \le 10^6$

11.The hiring team aims to find 3 candidates who are great collectively. Each candidate has his or her ability expressed as an integer. 3 candidates are great collectively if product of their abilities is maximum. Given abilities of **N** candidates in an array arr[], find the maximum collective ability from the given pool of candidates.

Input:
N = 5
Arr[] = {10, 3, 5, 6, 20}
Output: 1200
Explanation:
Multiplication of 10, 6 and 20 is 1200.

Input:
N = 5
Arr[] = {-10, -3, -5, -6, -20}
Output: -90
Explanation:
Multiplication of -3, -5 and -6 is -90.

Your Task:
You don't need to read input or print anything. Your task is to complete the function **maxProduct()** which takes the array of integers **arr[]** and **n**

as input parameters and returns the maximum product.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(1)

Constraints:

$3 \le N \le 10^7$

$-10^5 \le Arr[i] \le 10^5$

12. Given two strings s and t, return *the number of distinct subsequences of s which equals t.*

A string's **subsequence** is a new string formed from the original string by deleting some (can be none) of the characters without disturbing the remaining characters' relative positions. (i.e., "ACE" is a subsequence of "ABCDE" while "AEC" is not).

The test cases are generated so that the answer fits on a 32-bit signed integer.

Example 1:

Input: s = "rabbbit", t = "rabbit"
Output: 3
Explanation:
As shown below, there are 3 ways you can generate "rabbit" from S.
rabbb**it**
rab**bbit**
rabb**bit**

Example 2:

Input: s = "babgbag", t = "bag"

Output: 5

Explanation:

As shown below, there are 5 ways you can generate "bag" from S.

bab**g**bag

babgba**g**

babgb**ag**

ba**b**gb**ag**

babg**bag**

13. You are given a string s. You can convert s to a palindrome by adding characters in front of it.

Return *the shortest palindrome you can find by performing this transformation.*

Input: s = "aacecaaa"

Output: "aaacecaaa"

Input: s = "abcd"

Output: "dcbabcd"

14. You are given an array **A** of size **N.** You need to first push the elements of the array into a stack and then print minimum in the stack at each pop.

Example 1:

Input:
N = 5
A = {1 2 3 4 5}
Output:
1 1 1 1 1
Explanation:
After pushing elements to the stack,
the stack will be "top -> 5, 4, 3, 2, 1".
Now, start popping elements from the stack
popping 5: min in the stack is 1.popped 5
popping 4: min in the stack is 1. popped 4
popping 3: min in the stack is 1. popped 3
popping 2: min in the stack is 1. popped 2
popping 1: min in the stack is 1. popped 1

Example 2:

Input:
N = 7
A = {1 6 43 1 2 0 5}
Output:
0 0 1 1 1 1 1

15.Given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.

Example 1:

Input: nums = [3,0,1]

Output: 2

Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0,3]. 2 is the missing number in the range since it does not appear in nums.

Example 2:

Input: nums = [0,1]

Output: 2

Explanation: n = 2 since there are 2 numbers, so all numbers are in the range [0,2]. 2 is the missing number in the range since it does not appear in nums.

Example 3:

Input: nums = [9,6,4,2,3,5,7,0,1]

Output: 8

Explanation: n = 9 since there are 9 numbers, so all numbers are in the range [0,9]. 8 is the missing number in the range since it does not appear in nums.

16.Given two string arrays word1 and word2, return true if the two arrays represent the same string, and false otherwise. A string is represented by an array if the array elements concatenated in order forms the string.

Example 1:

Input: word1 = ["ab", "c"], word2 = ["a", "bc"]

Output: true

Explanation:

word1 represents string "ab" + "c" -> "abc"

word2 represents string "a" + "bc" -> "abc"

The strings are the same, so return true.

Example 2:

Input: word1 = ["a", "cb"], word2 = ["ab", "c"]

Output: false

Example 3:

Input: word1 = ["abc", "d", "defg"], word2 = ["abcddefg"]

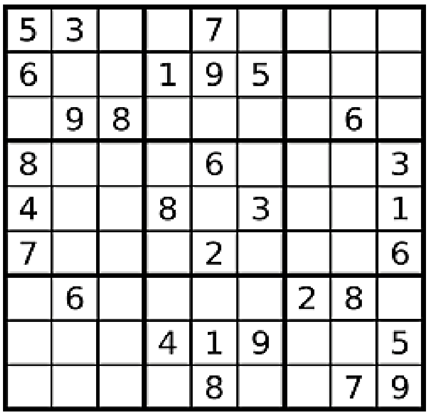
Output: true

17. Write a program to solve a Sudoku puzzle by filling the empty cells.

A sudoku solution must satisfy **all of the following rules**:

1. Each of the digits 1-9 must occur exactly once in each row.
2. Each of the digits 1-9 must occur exactly once in each column.
3. Each of the digits 1-9 must occur exactly once in each of the 9 3x3 sub-boxes of the grid.

The '.' character indicates empty cells

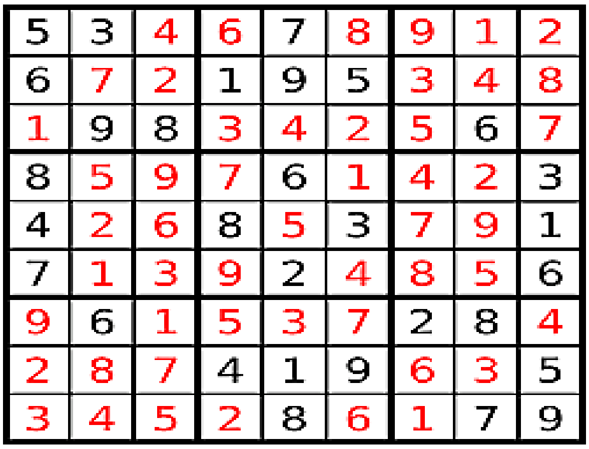


5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Input: board = [["5","3",".",".","7",".",".",".","."],["6",".",".","1","9","5",".",".","."],[".","9","8",".",".",".",".","6","."],["8",".",".",".","6",".",".",".","3"],["4",".",".","8",".","3",".",".","1"],["7",".",".",".","2",".",".",".","6"],[".","6",".",".",".",".","2","8","."],[".",".",".","4","1","9",".",".","5"],[".",".",".",".","8",".",".","7","9"]]

Output: [["5","3","4","6","7","8","9","1","2"],["6","7","2","1","9","5","3","4","8"],["1","9","8","3","4","2","5","6","7"],["8","5","9","7","6","1","4","2","3"],["4","2","6","8","5","3","7","9","1"],["7","1","3","9","2","4","8","5","6"],["9","6","1","5","3","7","2","8","4"],["2","8","7","4","1","9","6","3","5"],["3","4","5","2","8","6","1","7","9"]]

Explanation: The input board is shown above and the only valid solution is shown below:



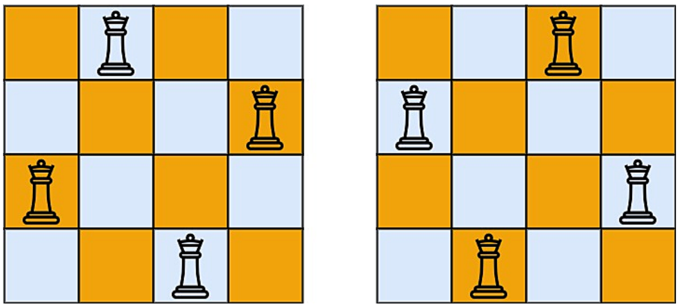
5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

18. The **n-queens** puzzle is the problem of placing n queens on an n x n chessboard such that no two queens attack each other.

Given an integer n, return *all distinct solutions to the **n-queens** puzzle*. You may return the answer in **any order**.

Each solution contains a distinct board configuration of the n-queens' placement, where 'Q' and '.' both indicate a queen and an empty space, respectively.

Example 1:



Input: n = 4
Output: [[".Q..","...Q","Q...","..Q."],["..Q.","Q...","...Q",".Q.."]]
Explanation: There exist two distinct solutions to the 4-queens puzzle as shown above

Example 2:

Input: n = 1
Output: [["Q"]]

19. Write a program to convert a number into a mono-digit number.

Conditions:

a) You are allowed to add and subtract the consecutive digits (starting from left).

b) You are allowed to do only one operation on a digit.

c) You cannot perform any operation on a resultant digit of the previous operation.

d) Your code should also find if a given number cannot be converted to a mono digit number.

Input	Output
72581	7(2+5)81
	77(8-1)
	777
3962	cannot create a mono digit number

20. Write a program to print all permutations of a given string. Note here you need to take all combinations as well, say for the input ABC the output should be as follows:

Input: ABC
Output:

```
A
B C
AB AC BA BC CA CB
ABC ACB BCA BAC CBA CAB
```

21. You’re given an array. Print the elements of the array which are greater than its previous elements in the array.

Input : 2, −3, −4, 5, 9, 7, 8

Output: 2 5 9 You should solve this question in O(n) time

22. You’re given an even number n. If n=4, you have to print the following pattern :

```
4444
4334
4334
4444
```

If n=6, then the pattern should be like this :

```
666666
655556
654456
654456
655556
666666
```

L3 Questions

1) Application description:

There are ‘n’ number of points in a highway out of which some points collect toll.

Each toll has its own charging scheme according to the vehicles and whether or not they are a VIP user.

If they are VIP user, 20% discount apply.

If the vehicle passes 3 toll gates, it has to pay in all the 3 toll gates according to the scheme of respective tolls.

There were 4 modules.

1. Given the details of vehicle type, start and destination……display the total toll paid during the journey and print the amount after applying the discount.

2. Display the details of all the tolls…..like what are all the vehicles(vehicle number) passed that respective toll and the amount each vehicle paid……and the total amount charged in that toll.

    3. Display the details of all the vehicles …….like what are all the journeys did it take….the
    start and destination of the same……tolls it passed during that journey….amount paid in
    that journey…..and the total amount paid by that vehicle.

    4. Assume the highway as a circular path……we have to find the short route and identify
    the tolls between that route and calculate the amount.

----

2) Given a MxN matrix filled with '-' and you need to drop the balloons in the desired columns starting from the bottom. You need to print the matrix when a new balloon is dropped. You need to continue getting inputs until the box is full or until the user chooses to stop.

```
  TEST CASE  :

  Enter the matrix size(m*n)  :  3 3
  Enter the column number    :  2
  Enter the color of the balloon  :  R
  Contents of the matrix    :
  - - -
  - - -
  - R -
  Do you wish to continue(Y/N)  :  Y
  Enter the column number    :  2
  Enter the color of the balloon  :  B
  Contents of the matrix    :
  - - -
  - B -
  - R -
  Do you wish to continue(Y/N)  :  Y
  Enter the column number    :  1
  Enter the color of the balloon  :  R
  Contents of the matrix    :
  - - -
  - B -
  R R -
  Do you wish to continue(Y/N)  :  Y
  Enter the column number    :  2
```

```
Enter the color of the balloon  :  R
Contents of the matrix    :
- R -
- B -
R R -
Do you wish to continue(Y/N)  :  N
Program Stopped
```

2. Extended version of the previous problem. Now you need to quit when a row become filled completely.

TEST CASE :

```
Enter the matrix size(m*n)  :  3 3
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
- - -
- R -
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  B
Contents of the matrix    :
- - -
- B -
- R -
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- R -
- B -
- R -
Column is filled completely. Program is terminated.
```

3. Extended version of the previous problem. Now you need to drop balloon in the first free cell from left if the specified column is filled in every row.

TEST CASE :

```
Enter the matrix size(m*n)  :  3 3
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
- - -
- R -
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  B
Contents of the matrix    :
- - -
- - -
B R -
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
- - -
B R R
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
- R -
B R R
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  B
Contents of the matrix    :
- - -
B R -
B R R
Do you wish to continue(Y/N)  :  N
Program terminated.
```

4. Extended version of the previous problem. If any column has three continuous balloons of same colors then we need to burst them.

TEST CASE :

```
Enter the matrix size(m*n)  :  3 3
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
- - -
- R -
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
- - -
R R -
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
- - -
R R R
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
- R -
R R R
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  B
Contents of the matrix    :
- - -
R R -
R R R
Do you wish to continue(Y/N)  :  Y
Enter the column number    :  2
Enter the color of the balloon  :  R
Contents of the matrix    :
- - -
R R R
R R R
```

```
  Do you wish to continue(Y/N)  :  Y
  Enter the column number    :  2
  Enter the color of the balloon  :  R
  Contents of the matrix    :
  - - -
  R - R
  R - R
  Do you wish to continue(Y/N)  :  N
  Program Terminated.
```

5. Extended version of the previous problem. Now you need to burst the three continuous colors in the same row

----
----

3) Lift system

There were 8 modules

1. Display the position of Lift

```
Lift   : L1 L2 L3 L4 L5
Floor  : 0  0  0  0  0
```

2. Assign Lift to the users

```
Input : 2  5
Output : L1 is assigned
Lift   : L1 L2 L3 L4 L5
Floor  : 5   0   0   0   0
```

3. Assign nearest lift by comparing their current positions Assume,

```
Lift   : L1 L2 L3 L4 L5
Floor  : 5   2   7   9   0
Input  : 4 10
Output : L1 is assigned
Lift   : L1 L2 L3 L4 L5
Floor  : 10   2   7   9   0  Explanation : L1 is near to 4 floor
```

4. If two lifts are nearest to the user’s source floor, the assign the lift with same direction of user’s requirement.

Example: if user request to move from 4 to 2 ,and if L3 is in 5th floor & L5 is in 3rd floor, then we should assign L3 because user requested for downward motion so L3 ill move down from 5th floor

5. Restrict L1 & L2 for 0-5th floor , L3 & L4 for 6-10th floor , L5 for 0-

10th

Initially all lifts are at 0th floor.

6. Assign lift with least number of stops

Example:

If L3 is in 9th floor

And L5 is at 8nd floor

If user wants to move from 8 to 0

We should assign L3 because L3 ill stop at 8,7,6 and then 0 NumberOfStops = 3, but L5 ill stop at 8,7,6,5,4,3,2,1,0 and NumberOfStops = 8 so we should assign L3

7. Assign capacity (Number of people capable to travel) to all lift and assign according to the capacity

8. If any lift is under maintenance then their current position should be marked as “-1” and that lift should not be assigned at any cost.