

Array & Vectors & deque

1. Back insertion sequence containers provide a `push_back` method that allow new elements to be added at the end of the container's range, and a `pop_back` method that allows the element at the end of the range to be removed from the container. Front insertion sequence containers provide a `push_front` method that allows new elements to be added at the beginning of the container's range and a `pop_front` method that allows the element at the beginning of the range to be removed from the container.

In your main function, declare variables of each of the following parameterized sequence container types : `vector<int>`, `deque <int>`. Build your solution , incorporating fixing any errors or warnings that occur for operations not supported by the container. Try using each of the above methods on each of the container variables in your main function, displaying errors and warnings when the operations are not supported. As the answer to this exercise, say whether each of the containers allow back insertion, front insertion, neither or both.

2. Assume these are the marks for advanced programing
`{ 20,25,25,50,40,22,25,45,42,44,33,35,44,48}`
 - a. Test if all marks are odd or not.
 - b. Test if all marks are odd or not.
 - c. Test if all the marks are above 20;
 - d. Test if anybody got 50;
 - e. Test if nobody has negative marks
 - f. Using for each function print the marks with 5 marks added (attendance marks)
 - g. Find all the indices of the array where the mark is 44
 - h. Find the average marks without writing your own function
 - i. Sort the marks
 - j. Print the adjacent difference between the marks.
 - k. Here are the marks of another class
`44, 43, 25, 39, 33, 38}` `{30,44, 35,22, 38, 49, 33, 41,`
Combine the marks of both the class (union)
 - l. Assume the following elements in an array `{1,3,2,5}`. Sort the array. Use permutation and generate all possible order of the sequence.

Stack

3. Given a stack of integers, sort it in ascending order using another temporary stack.

Test cases

Input : `[34, 3, 31, 98, 92, 23]`

Output : `[3, 23, 31, 34, 92, 98]`

Input : [3, 5, 1, 4, 2, 8]
Output : [1, 2, 3, 4, 5, 8]

4. <https://leetcode.com/problems/build-an-array-with-stack-operations/>

Queue

5. Create a Queue and do the following operations
- Create a queue
 - Check Queue Status
 - Add elements 3 6 7 (use emplace)
 - Swap
 - Print the first element one by one till the queue is empty
 - Insert 10 20 to queueone and 30 40 50 to queuetwo.
 - Swap the contents of the queue
 - Add the elements from queueone to queuetwo
 - Print the size of all the queues.
 - Implement queue of size 10 using array operations. Use STL array member functions

Priority queue

6. Print the maximum vector that is in a queue

Define priority queue like this

```
priority_queue<vector<int> > pq;
```

Test cases

Input 1 : { 10, 20, 30, 40 }

Input 2 : { 10, 20, 35, 40 }

Output will be : { 10, 20, 35, 40 }

Input 1 : { 10, 20, 30, 40 }

Input 2 : { 10, 20, 35, 40 }

Input 3 : { 30, 25, 10, 50 }

Output will be : { 30, 25, 10, 50 }

Input 1 : { 10, 20, 30, 40 }

Input 2 : { 10, 20, 35, 40 }
Input 3 : { 30, 25, 10, 50 }
Input 4 : { 20, 10, 30, 40 }
Input 5 : { 5, 10, 30, 40 }

Output will be : { 30, 25, 10, 50 }

7. Print minimum vector that is in queue.

8. <https://www.codechef.com/problems/SAVKONO>

Save Konoha

Pain is the leader of a secret organization whose goal is to destroy the leaf village(Konoha). After successive failures, the leader has himself appeared for the job. Naruto is the head of the village but he is not in a condition to fight so the future of the village depends on the soldiers who have sworn to obey Naruto till death.

Naruto is a strong leader who loves his villagers more than anything but tactics is not his strong area. He is confused whether they should confront Pain or evacuate the villagers including the soldiers (he loves his villagers more than the village). Since you are his advisor and most trusted friend, Naruto wants you to take the decision.

Pain has a strength of Z

and is confident that he will succeed. Naruto has N soldiers under his command numbered 1 through N . Power of i -th soldier is denoted by A_i . When a soldier attacks pain, his strength gets reduced by the corresponding power of the soldier. However, every action has a reaction so the power of the soldier also gets halved i.e. A_i changes to $[A_i/2]$

. Each soldier may attack any number of times (including 0). Pain is defeated if his strength is reduced to 0 or less.

Find the minimum number of times the soldiers need to attack so that the village is saved.

Input:

- First line will contain T
- , number of test cases. Then the test cases follow.
- The first line of each test case contains two space-separated integers N and Z
- .
- The second line contains N space-separated integers A_1, A_2, \dots, A_N

- the attacking power of the soldiers.

Output:

For each test case, print a single line. If Pain cannot be defeated, this line should contain the string "Evacuate" (without quotes). Otherwise, it should contain the minimum number of times the soldiers need to attack.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N \leq 10^5$
- $1 \leq Z \leq 10^8$
- $1 \leq A_i \leq 10^4$
-

Sample Input:

```
1
5 25
```

```
7 13 8 17 3
```

Sample Output:

```
2
```

EXPLANATION:

Soldier with power 17 hits Pain, The new powers of soldiers become [7,13,8,8,3]

and the strength of pain is reduced to 8, after that any soldier with a power greater than equal to 8 can hit Pain and thus defeat him, hence answer is 2.