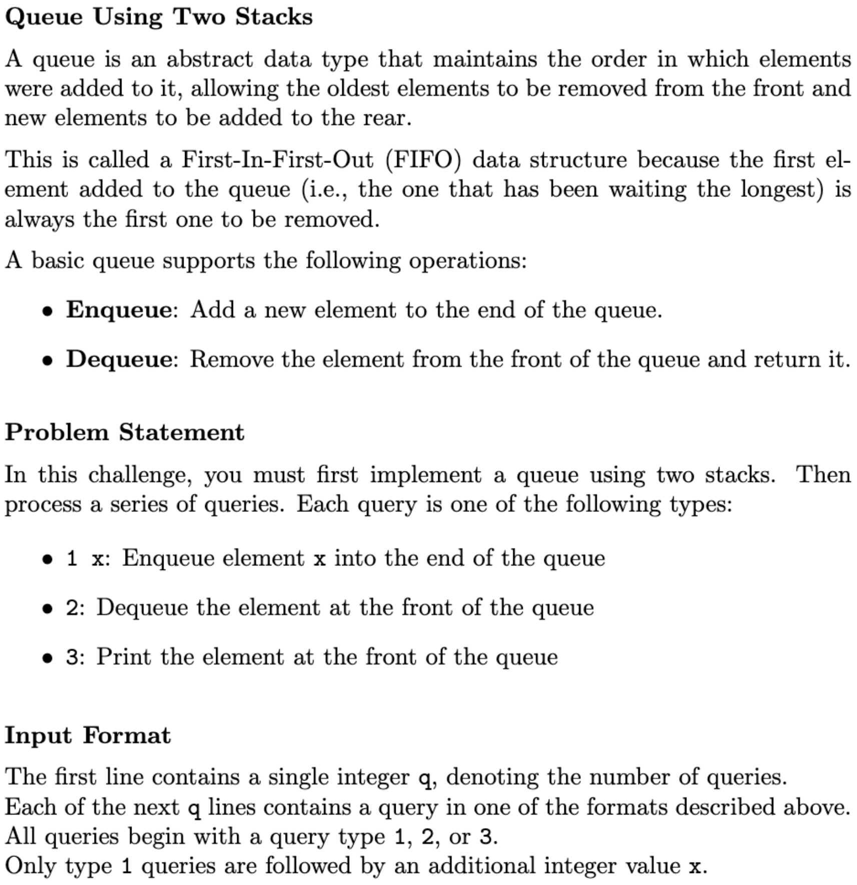
Problem statement:



Queue Using Two Stacks

A queue is an abstract data type that maintains the order in which elements were added to it, allowing the oldest elements to be removed from the front and new elements to be added to the rear.

This is called a First-In-First-Out (FIFO) data structure because the first element added to the queue (i.e., the one that has been waiting the longest) is always the first one to be removed.

A basic queue supports the following operations:

- **Enqueue**: Add a new element to the end of the queue.
- **Dequeue**: Remove the element from the front of the queue and return it.

Problem Statement

In this challenge, you must first implement a queue using two stacks. Then process a series of queries. Each query is one of the following types:

- `1 x`: Enqueue element `x` into the end of the queue
- `2`: Dequeue the element at the front of the queue
- `3`: Print the element at the front of the queue

Input Format

The first line contains a single integer `q`, denoting the number of queries.
Each of the next `q` lines contains a query in one of the formats described above.
All queries begin with a query type `1`, `2`, or `3`.
Only type `1` queries are followed by an additional integer value `x`.

Constraints

It is guaranteed that a valid answer always exists for each query of type `2` and `3`.

Output Format

For each query of type `3`, print the value at the front of the queue on a new line.

Sample Input

```
10
1 42
2
1 14
3
1 28
3
1 60
1 78
2
2
```

Sample Output

```
14
14
```