

Teque

time limit per test: 1 second (1 second for Python)
memory limit per test: 976.6 megabytes
input: standard input
output: standard output
points: 100 · problem code: teque

You are implementing a data structure called a **teque**.

A teque stores integers and supports six operations:

- `push_front x` : insert x at the front.
- `push_back x` : insert x at the back.
- `push_middle x` : if the current size is n , insert x at position $\lfloor \frac{n+1}{2} \rfloor$ (0-indexed).
- `pop_front` : remove and output the front value.
- `pop_back` : remove and output the back value.
- `pop_middle` : remove and output the value at index $\lfloor \frac{n-1}{2} \rfloor$ (0-indexed).

Process all operations in order.

Input

The first line contains an integer q , the number of operations.

Each of the next q lines is one operation in one of these forms:

- `push_front x`
- `push_back x`
- `push_middle x`
- `pop_front`
- `pop_back`
- `pop_middle`

It is guaranteed that every pop operation is valid (the teque is non-empty).

Output

For every pop operation, output one line containing the removed value.

Constraints

- $1 \leq q \leq 2 \cdot 10^5$
- $0 \leq x \leq 10^9$

Example 1

Input

```
10
push_back 1
push_back 2
push_front 3
push_middle 4
pop_middle
pop_front
push_middle 5
pop_back
pop_middle
pop_front
```

Output

```
1
3
2
4
5
```

Example 2

Input

```
8
push_middle 10
push_middle 20
push_middle 30
pop_middle
pop_middle
pop_middle
push_back 7
pop_front
```

Output

```
30
10
20
7
```