Title: Tasks

Problem Statement:

Little Timmy is a very forgetful boy, therefore he needs a to-do list to keep track of his tasks for the day. Can you help him implement his to-do list?

He has n commands, each command will be of the form:

Command	Description
`ADD_FRONT X`	Adds the new task to the front of the to-do list.
`ADD_BACK X`	Adds the new task to the back of the to-do list.
`ADD_MIDDLE X`	Adds the new task to the middle of the to-do list. If k is the size of the list before insertion, the insertion index for X is: (k+1)/2, based on 0-indexing.
`REMOVE_FRONT`	Removes the task at the front of the to-do list.
`REMOVE_BACK`	Removes the task at the back of the to-do list.
`REMOVE_MIDDLE`	Removes the task at the middle of the to-do list. If k is the size of the list, the removal index is: k/2, based on 0-indexing.
`GET I`	Gets the ith task from the current todo list. `I` <= current size of the list. Tasks are 0-indexed.

X is a string between 1 to 10 characters long. It is guaranteed that the REMOVE operation will be called on a non-empty to-do list.

Input:

On the first line, an integer n. (1 <= n <= 10^6)

On the next n lines, there will be one command per line. Each command will be of the form shown above in the table.

Output:

For every **"`GET I`"** operation, print a line, which is the ith task.

Example + Explanation:

Sample Input 1:	To-do List

`9`
`ADD_BACK A`
`ADD_FRONT B`
`ADD_MIDDLE C`
`GET 0`
	`[]`
	`[A]`
	`[B, A]`
	`[B, C, A]`
	`[B, C, A]`

GET 1	[B, C, A]
GET 2	[B, C, A]
ADD_MIDDLE D	[B, C, D, A]
GET 1	[B, C, D, A]
GET 2	[B, C, D, A]

Sample Output 1:

```
B
C
A
C
D
```

Sample Input 2:	To-do List
8	[]
ADD_MIDDLE A	[A]
ADD_MIDDLE B	[A, B]
ADD_MIDDLE C	[A, C, B]
ADD_MIDDLE D	[A, C, D, B]
REMOVE_MIDDLE	[A, C, B]
REMOVE_MIDDLE	[A, B]
REMOVE_MIDDLE	[A]
GET 0	[A]

Sample Output 2:

```
A
```