

题目描述

使用链表实现一个双端队列，双端队列的头部和尾部均可以进行插入值和取出值操作，你可以使用以下5种指令操作双端队列：

- H3
1. `pushFront [val]`，其中 `val` 为一个 `int`，且 $val \geq 0$ ，例如 `pushFront 1`，该指令可以将 `val` 放在节点中 插入 到双端队列的 最前面。
 2. `pushBack [val]`，其中 `val` 为一个 `int`，且 $val \geq 0$ ，例如 `pushBack 2`，该指令可以将 `val` 放在节点中 插入 到双端队列的 最后面。
 3. `popFront`，该指令可以 打印 双端队列 最前面有效节点的 `val`，并且将其 移出 双端队列，如果双端队列内没有有效节点，则打印-1。
 4. `popBack`，该指令可以 打印 双端队列 最后面有效节点的 `val`，并且将其 移出 双端队列，如果双端队列内没有有效节点，则打印-1。
 5. `getSize`，该指令可以 打印 双端队列的 有效节点数。

结构设计

我们设计好了部分结构体和部分方法声明，可以将它们运用到自己的代码中，也可以自己进行设计和实现，但必须使用链表。

H3

```
//链表节点
struct Node{
    Node* next;//前一个节点
    Node* prev;//后一个节点
    int val;//该节点存放的数字
};

//双端队列
struct Deque{
    int size;//有效节点数
    Node* front;//虚拟头节点，不作为有效节点
    Node* rear;//虚拟尾节点，不作为有效节点
};

void push_front (Deque* self, int value);
void push_back (Deque* self, int value);
void pop_front (Deque* self);
void pop_back (Deque* self);
```

输入输出

输入

第一行输入一个自然数 `n`，表示接下来会进行 `n` 次操作。

接下来的 `n` 行每行为一条指令，具体如题目描述所示。

H3

输出

H6

`popFront`，`popBack`，`getSize` 三条指令存在输出，均输出 `int`，且需要进行换行。

输入输出示例

示例1

H6

输入

H3

5

H6

```
pushFront 1
pushBack 2
popBack
popBack
popFront
```

输出

```
2 //popBack的输出
1 //popBack的输出
-1 //popFront的输出
```

示例2

输入

H6

```
5
pushFront 1
pushFront 2
getSize
popBack
getSize
```

输出

```
2 //getSize的输出
1 //popBack的输出
1 //getSize的输出
```

