استک‌پشتک

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

با استفاده از Singly-LinkedList، داده ساختاری به نام استک‌پشتک طراحی کنید که عمل های $addLast$، $removeLast$ و $addFirst$ را پشتیبانی کند.

در ابتدا یک استک‌پشتک خالی در اختیار داریم. در هر مرحله، علاوه بر ۳ عمل بالا، می‌توانیم عمل $reverse(i, j)$ را نیز انجام دهیم. این عمل جایگاه عنصر های iام تا jام را برعکس می‌کند. در یک استک‌پشتک که شامل n عنصر است، عنصر سر لیست(front) عنصر اول بوده و عنصر ته لیست(back) عنصر nام می‌باشد.

فرمت و شرح دقیق عمل ها به صورت زیر است:

- ۱. $addFirst\ x$: عنصر x را به ابتدای لیست اضافه کن.
- ۲. $addLast\ x$: عنصر x را به انتهای لیست اضافه کن.
- ۳. $removeLast$: عنصر انتهای لیست را حذف کن و آن را چاپ کن. در صورتی که لیست خالی بود، پیام $empty$ را چاپ کن.
- ۴. $reverse\ i\ j$: جایگاه عنصر های iام «تا» jام را برعکس کن. $(i \leq j)$

شیوه بهینه $reverse$ کردن یک SLList را در کلاس حل تمرین بحث کردیم.

ورودی

در خط اول ورودی عدد $n \leq 5000$ تعداد عمل هایی ست که باید بر روی استک‌پشتک انجام دهید. در n خط بعدی، عمل هایی که باید انجام بشوند به ترتیب مشخص شده اند. عناصر ورودی، نام منفی بوده و کوچک‌تر از 10^6 هستند.

خروجی

با توجه به نوع عمل، خروجی مناسب را چاپ کنید.

مثال

ورودی نمونه ۱

```
6
removeLast
addLast 4
addLast 11
addFirst 9
reverse 2 3
removeLast
```

خروجی نمونه ۱

```
empty
4
```

ورودی نمونه ۲

```
11
addFirst 8
removeLast
addLast 6
removeLast
addFirst 3
addFirst 2
addFirst 1
removeLast
removeLast
removeLast
removeLast
```

خروجی نمونه ۲

```
8
6
3
2
1
empty
```