

Escuela Colombiana de Ingeniería Julio Garavito**05 de Abril 2022****Parcial Práctico 02****Tiempo Límite:** 180 Minutos**Profesor Encargado:** Sebastián Camilo Martínez Reyes**Nombre del estudiante:****Número de Carné:**

Este examen contiene 5 paginas (incluida esta sección introductoria) en donde encontrará 3 preguntas.

Cantidad máxima de puntos posible: **7**.

Puntaje para máxima calificación: **5**.

Recomendaciones

Tómese 5 minutos para leer todos los enunciados, defina una estrategia de solución y proceda a resolver el examen. Recuerde que no está permitido el uso de material externo al suministrado por el profesor, el uso de telefonos celulares está prohibido. No podrá dejar el lugar del examen a menos de haber terminado el mismo.

Cualquier violación a las recomendaciones anteriormente mencionadas o cualquier otra incluida en el reglamento estudiantil de la **Escuela Colombiana de Ingeniería Julio Garavito**, acarreará la anulación del presente examen y desencadenará los procesos correspondientes.

Tabla de puntaje (Para uso exclusivo del calificador)

Pregunta	Puntos	Resultado
1	3	
2	2	
3	2	
Total:	7	

1. (3 puntos) A- Deque 1

Dado un mazo ordenado de N cartas numeradas de $1..N$, donde la carta 1 esta a la cabeza y la carta N al final del mazo. Siempre y cuando hayan al menos dos cartas en el mazo, se ejecuta la siguiente operación: **Se descarta la primera carta del mazo, luego la nueva primera carta se envía al final del mazo.**

Su tarea es encontrar la secuencia de cartas descartadas y la ultima carta que queda en el mazo.

Condiciones Especiales

El uso de colecciones, arreglos y estructuras de datos por libreria está restringido para este problema, realicé las implementaciones de las estructuras de datos que considere necesarias (Clases) para la solución.

Nota: Colecciones dentro de las clases también está restringido.

Input

Cada linea de entrada (Excepto la ultima) contiene un número $2 \leq n \leq 1000000$. La ultima línea contiene 0 y no debe ser procesada.

Output

Para cada caso de entrada, produzca dos líneas de salida. La primera línea mostrará la secuencia de cartas descartadas, la secunda linea mostrará la ultima carta del mazo.

Sample Input

```
7
19
10
6
0
```

Sample Output

```
Discarded cards: 1, 3, 5, 7, 4, 2
Remaining card: 6
Discarded cards: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 4, 8, 12, 16, 2, 10, 18, 14
Remaining card: 6
Discarded cards: 1, 3, 5, 7, 9, 2, 6, 10, 8
```

Remaining card: 4
Discarded cards: 1, 3, 5, 2, 6
Remaining card: 4

2. (2 puntos) C- Fila en el Banco

Adriana está sorprendida por cómo son las filas en los bancos en Colombia. En Alemania, por ejemplo, las filas oscilan entre 0 y 5 personas, pero en Colombia no. En medio de su aburrimiento ha decidido programar.

Adriana quiere un algoritmo que reciba el nombre de los clientes que llegan al banco y que imprima sus nombres en pantalla cuando sean llamados según el turno asignado.

Condiciones Especiales

El uso de colecciones, arreglos y estructuras de datos por librería está restringido para este problema, realicé las implementaciones de las estructuras de datos que considere necesarias (Clases) para la solución.

Nota: Colecciones dentro de las clases también está restringido.

Input

La primera línea del input contiene un entero con la cantidad de casos a procesar. Cada caso consiste de

1. una línea con un entero $0 \leq n \leq 1000000$
2. seguida por n líneas, cada una con un nombre o con la **string** Siguiente. El orden en que aparecen los nombres de los clientes indica el orden en que deben ser atendidos. Los nombres no contienen espacios y pueden haber varios clientes con el mismo nombre.

Output

Por cada caso del input, haga lo siguiente: por cada Siguiente del input, imprima una línea con el nombre del cliente que debe pasar a ser atendido. Si no hay clientes en espera, imprima No hay fila.

Sample Input

```
1
17
Siguiente
Adriana
Siguiente
Siguiente
Sebastián
Sergio_Sierra
Adriana
Wilmer_Garzón
Siguiente
Siguiente
Siguiente
Felipe_Díaz
Carlos_Antonio_Pinzón
Siguiente
David_Bravo
Siguiente
Siguiente
```

Sample Output

```
No hay fila
Adriana
No hay fila
Sebastián
Sergio_Sierra
Adriana
Wilmer_Garzón
Felipe_Díaz
Carlos_Antonio_Pinzón
```

3. (2 puntos) D- Paréntesis Balanceados

Usted tiene una cadena que se compone de paréntesis `()` y `[]`. Una cadena de este estilo se dice que es correcta si:

1. Si es una cadena vacía
2. Si A y B son correctas, AB es correcta
3. Si A es correcta, (A) y $[A]$ es correcta

Input

El archivo contiene un entero n y una secuencia de n cadenas de paréntesis '(' u '['.

Hay una cadena por línea.

Output

Una secuencia de 'Yes' o 'No'

Sample Input

```
3
([)]
(([])))
([() [] ()])()
```

Sample Output

```
Yes
No
Yes
```

La motivación es lo que te pone en
marcha, el hábito es lo que hace que
sigas

Jim Ryun