Proyecto 1: Colección de espadas

Tobi es un guerrero espadachín que está obsesionado con el orden, particularmente a Tobi le encanta coleccionar espadas que a su vez usa siempre que está en una feroz batalla. Tobi tiene una larga mesa donde coloca todas sus espadas identificadas con un numero entero, cada vez que obtiene una nueva espada para su colección la coloca lo más a la derecha de la mesa por delante de todas las demás ya colocadas. Tobi hace esto ya que cada vez que él toma una espada para tener una batalla, también toma la espada que está más a la derecha (es decir la última que coloco en su colección). Cabe destacar que cuando Tobi va a una batalla la espada que el lleva termina rompiéndose en pedazos debido a la gran fuerza que aplica sobre ella. Esta manera en la que Tobi guarda y usa sus espadas puede provocar que las espadas que inicialmente coloco en su colección no las use y por tanto se desgasten con el tiempo, para evitar este problema Tobi lo que hace es tomar la primera mitad (la mitad izquierda) de su colección (si hay un número impar de espadas, él toma la mayor cantidad que no exceda la mitad del total de espadas) y las coloca lo más a la derecha posible, de manera que la primera espada quede inmediatamente después de la que estaba de ultima, de esta manera Tobi puede usar esas espadas y evitar que se desgasten.

Su tarea es ayudar a Tobi a determinar el orden final de sus espadas una vez realizadas un conjunto de operaciones que simulan los cambios que Tobi hace en su colección de espadas.

Entrada

La primera línea de entrada contiene un entero N indicando el número de cambios en su colección, luego vendrán N líneas, cada una con una cadena de caracteres que indica la operación a realizar, si la cadena es “insertar” entonces vendrá un entero en la misma línea que hay que insertar en su colección, si es “tomar” entonces se tomara una espada para la próxima batalla, y por ultimo si es “reordenar” entonces se deberá aplicar la operación de mover las espadas tal y como lo hace Tobi.

Salida

La salida solo serán 2 líneas, la primera línea es un entero que indica la cantidad de espadas que posee Tobi luego de realizar todas las operaciones, la segunda línea son los enteros de las espadas en el orden en que quedaron luego de aplicar las operaciones.

Ejemplos

Entrada	**Salida**
8 insertar 1 insertar 2 insertar 4 insertar 3 insertar 5 insertar 8 tomar reordenar	5 4 3 5 1 2
3 insertar 1 insertar 2 insertar 3 reordenar	3 2 3 1
2 insertar 1 tomar	0

Consideraciones:

- N será un entero positivo de hasta 1.000.000.

- Inicialmente la mesa está vacía es decir no contiene ninguna espada.
- Los enteros que identifican a las espadas serán enteros cualesquiera y pueden repetirse.
- Todas las operaciones deben ser implementadas con un orden de complejidad constante es decir de O(1). Con excepción de imprimir el resultado.
- La entrada siempre será válida, es decir las cadenas siempre estarán bien escritos tal y como se describe, además nunca vendrá una instrucción "tomar" si la mesa esta vacía.
- La entrada y salida debe ser estricta, es decir su programa debe leer los datos tal y como se especifican, e imprimir las respuestas tal como se especifica en el ejemplo. Cualquier información adicional que solicite o proporcione será penalizada.
- La lectura y escritura de datos debe ser hecho usando la Entrada/Salida estándar de C++.
- El programa debe ser realizado en C++ sin el uso de STL para las estructuras de datos, a excepción de la clase **string** provista por C++.
- Debe emplear Programación Orientada a Objetos en su programa.
- El código fuente debe estar intradocumentado adecuadamente. Se debe entregar un informe en formato PDF con la explicación de la solución de su proyecto (sin incluir código), explicación de las estructuras usadas, la complejidad de cada una de las funciones descritas y la identificación del autor. La ausencia de este archivo califica al proyecto como no entregado.
- Será compilado y evaluado en el compilador g++ (Linux). Si el proyecto no compila no será corregido.
- El proyecto debe ser realizado de manera individual.
- La revisión del proyecto no es obligatoria y solo se revisaran proyectos que el grupo docente considere necesario.
- El proyecto debe entregarse el día 24 de Febrero de 2017 a través de la página de la asignatura, hasta las 11:59 pm como hora límite. No se aceptarán proyectos fuera de las horas establecidas para ello, y se considerarán como no entregados.

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
ALGORITMOS Y ESTRUCTURAS DE DATOS

Las copias entre proyectos tendrán una calificación de 0 puntos, además de una sanción para todos los involucrados