

Pilas con elemento mínimo

En este problema se debe implementar un TAD `Stack_min` que ofrezca las operaciones típicas de una pila: `push`, `pop`, `top` y `empty` y un nuevo método `min_stack` que permita consultar el elemento mínimo de la pila. Todas las operaciones se deben ejecutar en tiempo constante en el caso peor.

Requisitos de implementación.

En la representación de la nueva pila se empleará el tipo `stack` de la librería STL. Observad que para conseguir que la operación `min_stack` tenga coste constante se debe guardar información sobre los elementos mínimos almacenados en la pila, de forma que se pueda acceder a esta información en tiempo constante. El problema radica en que cuando se desapila un elemento que es el mínimo de la pila, debemos obtener el nuevo mínimo en tiempo constante, es decir, no se pueden desapilar y volver a apilar los elementos para buscar ese nuevo mínimo. En lugar de esto se realizará una implementación que tenga coste en memoria lineal en el número de elementos de pila. Se seleccionará una estructura de datos auxiliar en la que llevaremos precalculados los mínimos.

El problema se puede resolver con coste constante en memoria utilizando una única variable en lugar de una estructura auxiliar. La idea de esta solución puede consultarse en la página

<https://www.geeksforgeeks.org/design-a-stack-that-supports-getmin-in-o1-time-and-o1-extra-space/?ref=gcse>.

Esta solución aporta menos a la asignatura que la solución que utiliza una estructura auxiliar, por lo que se recomienda practicar la solución que utiliza la estructura auxiliar.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso se escribe en dos líneas. En la primera se indica el número de operaciones que se van a realizar sobre la pila y el tipo de elementos que se guardan en la pila (I para valores enteros, C para caracteres). En la segunda línea se indica cada una de las operaciones. La operación de apilar (`push`) un elemento se representa con el carácter A seguido del valor que se desea apilar. La operación de desapilar (`pop`) se denota con el carácter D. La operación de consultar la cima (`top`) se denota con el carácter C y por último la operación de consultar el mínimo (`minimo`) se denota con el carácter M. La entrada termina con un valor cero que no debe procesarse.

El número máximo de operaciones que se pueden realizar sobre la pila es mayor que cero y menor de 150 000. Si la pila es de enteros, estos son mayores o iguales que cero y menores que 10^6 . Si la pila es de caracteres, estos son caracteres del alfabeto inglés.

Salida

Para cada caso de prueba se escribirá una línea por cada una de las operaciones que devuelven un elemento, con el elemento que devuelve la operación. Si alguna de las operaciones produce error se escribirá una línea indicando el error que se ha producido. Cada caso termina con una línea con tres guiones.

Entrada de ejemplo

```
12 I
A 3 A 2 A 1 M D C M D C M D M
20 I
A 1 A 2 A 3 M D C M D M D A 4 C A 2 A 7 M C M D D M
6 I
M C A 2 M D D
14 I
A 7 M A 4 M A 6 M A 3 M D M D M D M
8 C
A b A b A b M D M D M
17 C
A e A b A e A b A a A a M D M D M D M D M D M
0
```

Salida de ejemplo

```
1
2
2
3
3
ERROR: Pila vacia
---
1
2
1
1
4
2
7
2
4
---
ERROR: Pila vacia
ERROR: Pila vacia
2
ERROR: Pila vacia
---
7
4
4
3
4
4
7
---
b
b
b
---
a
a
b
b
b
e
---
```

Autor: Isabel Pita