Push i pop d'elements marcats a la classe Stack — X16671_ca

En aquest exercici estendrem la classe `Stack` afegint un nou mètode anomenat `push_mark`. A priori, aquest mètode fa el mateix que el mètode `push` existent. També afegirem un altre nou mètode anomenat `pop_mark`. L'efecte de `pop_mark` és anar eliminant elements del cim de la pila fins que, o bé hi ha un element al cim que va ser afegit amb `push_mark`, o bé s'ha buidat la pila.
Fixeu-vos en el següent exemple de programa i en la seva execució descrita en els seus comentaris, a on denotem entre corxets els elements afegits amb `push_mark`:

```
Stack<string> s;  // s:
s.push("a");      // s: a
s.push("b");      // s: a,b
s.push_mark("c"); // s: a,b,[c]
s.push("d");      // s: a,b,[c],d
s.push("e");      // s: a,b,[c],d,e
s.push_mark("f"); // s: a,b,[c],d,e,[f]
s.push("g");      // s: a,b,[c],d,e,[f],g
s.push("h");      // s: a,b,[c],d,e,[f],g,h
s.push("i");      // s: a,b,[c],d,e,[f],g,h,i
s.pop();          // s: a,b,[c],d,e,[f],g,h
s.pop_mark();     // s: a,b,[c],d,e,[f]
s.pop_mark();     // s: a,b,[c],d,e,[f]
s.pop();          // s: a,b,[c],d,e
s.pop();          // s: a,b,[c],d
s.pop_mark();     // s: a,b,[c]
s.pop_mark();     // s: a,b,[c]
s.pop();          // s: a,b
s.pop_mark();     // s:
```

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `stack.hh`, a on hi ha una implementació de la classe genèrica `Stack`. Haureu de buscar dins `stack.hh` les següents línies:

```
struct Item {
  T value;
  Item* next;
  // Afegeix el que calgui per a poder recordar si un element està marcat.
  // ...
};

...

// Modifica aquesta funció per a recordar que l'element afegit no està marcat
void push(T value) {
  Item *pnewitem = new Item();
  pnewitem->value = value;
```

```
    pnewitem->next = ptopitem;
    ptopitem = pnewitem;
    _size++;
  }

  ...

  // Pre:
  // Post: S'afegeix value al cim de la pila, com a element marcat.
  // Descomenteu les següents dues linies i implementeu el mètode:
  // void push_mark(T value) {
  // ...
  // }

  // Pre:
  // Post: S'han eliminat del cim de la pila el mínim nombre d'elements necessa
  //       per tal de garantir que o bé el cim de la pila té un element que va
  //       afegit amb push_mark, o bé la pila és buida.
  //       En particular, si el cim de la pila ja tenia un element marcat, no s
  // Descomenteu les següents dues linies i implementeu el mètode:
  // void pop_mark() {
  // ...
  // }
```

Afegiu el que cal dins de l'struct Item, i descomenteu les linies que s'indiquen i implementeu els mètodes.
D'entre els fitxers que s'adjunten a l'exercici també hi ha `main.cc` (programa principal), i el podeu compilar directament, doncs inclou `stack.hh`. Només cal que pugeu `stack.hh` al jutge.
Observació: En aquest exercici es prefereix una solució basada en manegar punters abans que una solució basada en cridar a mètodes primitius de la pròpia classe.
Observació: En els jocs de proves no es copiaran piles. Per tant, no cal que adapteu les funcions per a copiar piles, i per tant no cal decidir que passa amb els elements marcats d'una pila quan es copien en una altra pila.

Entrada

L'entrada del programa té una primera línia amb o bé `int` o bé `string`, que indica el tipus T dels elements de la pila `s` amb la que treballarà el programa, que se suposa inicialment buida. Després, hi ha una seqüència d'instruccions del següent tipus que s'aniran aplicant sobre la pila:

```
push x (x és de tipus T)
pop
top
size
print
push_mark x (x és de tipus T)
pop_mark
```

Se suposa que la seqüència d'entrada serà correcta (sense pop ni top sobre pila buida).
El programa principal que us oferim ja s'encarrega de llegir aquestes entrades i fer les crides als corresponents mètodes de la classe pila. Només cal que feu els canvis abans esmentats.

Sortida

Per a cada instrucció `top`, s'escriurà el top actual de la pila, per a cada instrucció `print`, s'escriurà el contingut de la pila, i per a cada instrucció `size`, s'escriurà la mida de la pila. El programa que us oferim ja fa això. Només cal que implementeu els mètodes abans esmentats i les modificacions que considereu oportunes a la classe.

Exemple d'entrada 1

```
pop_mark
size
pop_mark
push_mark z
pop_mark
size
pop
pop_mark
size
push a
push b
size
pop_mark
size
pop_mark
size
push_mark a
top
print
size
push b
top
print
size
push c
top
print
size
push d
top
print
size
push_mark bb
top
print
size
push ccc
top
print
size
push dd
top
print
size
push ee
top
print
size
pop
top
print
size
pop_mark
top
print
size
pop_mark
top
print
size
pop_mark
top
print
size
pop
top
print
size
pop_mark
top
print
size
push dd
top
print
size
push eeee
top
print
size
pop_mark
top
print
size
pop
print
size
```

Exemple de sortida 1

```
0
1
0
2
0
0
a
a
1
b
a b
2
c
a b c
3
d
a b c d
4
bb
a b c d bb
5
ccc
a b c d bb ccc
6
dd
a b c d bb ccc dd
7
ee
a b c d bb ccc dd ee
8
dd
a b c d bb ccc dd
7
bb
a b c d bb
5
bb
a b c d bb
5
d
a b c d
4
a
a
1
dd
a dd
2
eeee
a dd eeee
3
a
a
1

0
```

Exemple d'entrada 2

```
push b
push b
push_mark ca
top
size
top
push_mark ad
size
push c
pop
pop
push aab
push ccb
push bb
top
push a
pop_mark
top
push bbc
size
size
pop
push c
push cb
push bd
pop
size
push_mark ddb
push_mark c
pop_mark
push db
push c
size
top
top
pop
push ca
size
push_mark b
top
size
size
size
push bc
push dcc
push ac
top
top
push cc
push aa
pop
size
size
size
pop
push db
size
top
push da
```

```
pop
size
top
push c
size
push a
push dd
top
push ddb
push acd
push_mark d
push_mark c
push a
pop_mark
top
push a
pop
push cbc
size
top
pop_mark
print
```

Exemple de sortida 2

```
ca
3
ca
4
bb
ca
4
4
5
9
c
c
9
b
10
10
10
ac
ac
14
14
14
14
db
14
db
15
dd
c
22
cbc
b b ca c cb ddb c db ca b bc dcc ac db c a dd ddb acd c
```

Observació

Avaluació sobre 10 punts: (Afegiu comentaris si el vostre codi no és prou clar)

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics. Per exemple, una solució que superi tots els jocs de proves però que manegui incorrectament la memòria serà invalidada i tindrà nota 0.
Una solució basada en cridar a mètodes primitius de la pròpia classe pot tenir una certa penalització en la nota.

Informació del problema

Autor : PRO2
Generació : 2024-04-23 16:46:52