Punt mig d'una llista — X33439_ca

Escriviu un programa que simuli el manteniment d'una llista d'enters, a base de llegir instruccions que la van actualitzant, i instruccions que la van consultant. La llista d'enters se suposa inicialment buida, i les instruccions son dels següents tipus: afegir elements al principi o final de la llista, treure elements del principi o final de la llista, o consultar el valor que hi hagi exactament en el punt mig de la llista. En cas que la llista tingui mida parell, no hi ha punt mig i s'escriurà `error` quan es demani consultar el punt mig.
És obligatori utilitzar només el constructor de tipus `list` com a mecanisme d'emmagatzemament massiu de dades. En particular, no es pot usar ni `vector`, ni `stack`, ni `queue`. Podeu declarar tants `list` com volgueu i del tipus que volgueu (que no sigui cap altre mecanisme d'emmagatzemament massiu), i podeu usar iteradors sobre llistes.

Entrada

La entrada consisteix en un nombre arbitrari de linies, cadascuna amb una instrucció. Les instruccions poden ser de la següent forma:

```
push_front x
push_back x
pop_front
pop_back
get_mid_value
```

Sortida

Les instruccions `pop_front` i `pop_back` poden escriure `error` a la sortida, i la instrucció `get_mid_value` pot escriure `error` o un valor a la sortida. Se sobreentén que la llista que simulem està inicialment buida, i que l'efecte de cada instrucció és el següent:

- `push_front x` afegeix l'enter `x` al principi de la llista.
- `push_back x` afegeix l'enter `x` al final de la llista.
- `pop_front` elimina l'element que es troba al principi de la llista. Si la llista és buida ha d'escriure `error` i salt de linia.
- `pop_back` elimina l'element que es troba al final de la llista. Si la llista és buida ha d'escriure `error` i salt de linia.
- `get_mid_value` escriu l'element que es troba just enmig de la llista si aquesta té mida senar, i escriu `error` si la llista té mida parell. També escriu un salt de linia. Per tant, la sortida tindrà tantes linies com instruccions `get_mid_value` hi hagi més totes aquelles altres que hagin produït `error`.

Exemple d'entrada 1

```
pop_front
get_mid_value
```

```
pop_back
push_front 0
push_back 3
pop_front
```

```
push_back 0
get_mid_value
pop_back
push_back 2
pop_back
push_back 4
get_mid_value
push_front 3
get_mid_value
push_back 7
push_back 1
get_mid_value
push_front 5
push_back 6
push_back 8
push_back 9
get_mid_value
```

Exemple de sortida 1

```
error
error
error
error
error
3
4
7
```

Exemple d'entrada 2

```
push_back -385
push_front 450
push_front -538
pop_back
pop_back
pop_front
pop_front
get_mid_value
push_front -639
pop_back
pop_front
push_front -157
push_front -387
pop_front
push_front 137
pop_back
get_mid_value
push_back -476
push_front -470
get_mid_value
get_mid_value
push_front -687
get_mid_value
push_back 320
push_front -372
push_front -212
push_back -813
push_front -680
push_front -834
pop_back
push_back -433
pop_back
push_front -892
push_back 40
get_mid_value
```

Exemple de sortida 2

```
error
error
error
137
137
137
error
-687
```

Exemple d'entrada 3

```
push_back 3
get_mid_value
pop_front
get_mid_value
push_front 4
```

```
get_mid_value
pop_back
get_mid_value
push_front 1
get_mid_value
push_back 2
get_mid_value
push_front 3
get_mid_value
```

Exemple de sortida 3

```
3
error
4
error
1
error
1
```

Observació

Per tal de superar els jocs de proves públics i aconseguir una nota raonable, podeu fer una implementació senzilla mantenint la llista representada en un `list<int>`, i a on el càlcul de `get_mid_value` tingui cost lineal i consisteixi en recorrer la llista fins el punt mig i obtenir així el valor. De fet, us recomanem que no us lieu i seguiu aquest enfoc.
Però els jocs de proves privats són grans. Per tal d'aconseguir superar-los tots i obtenir així la màxima nota, convindrà trobar un enfoc més eficient.

Informació del problema

Autor : PRO1
Generació : 2022-09-13 23:03:28