

Aquest és un problema del jutge per fer lliuraments de l'examen de la pràctica

- **Aquest examen dura una hora i quaranta-cinc minuts**
- **No es corregirà cap lliurament que no compili**
- L'examen es fa sense apunts
- El 50% de la nota és de l'execució, l'altre 50% de la correcció manual
- Es corregirà el darrer enviament amb més jocs de proves correctes
- En cas de no passar cap joc de proves es farà la correcció manual del darrer enviament que compili
- El nombre d'enviaments no fa baixar la nota

Descarregueu els fitxers públics per obtenir el material:

- enunciat detallat d'aquest examen
- enunciat de la pràctica
- especificació de `BinTree`
- especificació de `list`
- fitxer `llegeixme.txt`
- codi ja implementat
- fitxer `Makefile`
- joc de proves públic

Després de llegir l'enunciat de l'examen atentament, així com el fitxer `llegeixme.txt`, apliqueu la instrucció `make plantillas`. Us apareixeran els fitxers `priv_Cluster.hh` i `solution.cc` que haureu fer servir de plantilla. A més, `solution.cc` conté l'especificació de les operacions públiques que heu d'implementar. Les operacions privades que necessiteu, que només podran ser de la classe `Cluster` les decidiu vosaltres: heu d'escriure la capçalera a `priv_Cluster.hh` i la implementació a `solution.cc`.

Les operacions que heu d'implementar, pertanyen a les classes `Cluster` i `Pendientes`. Ara us resumim les novetats més importants de la implementació que us donem d'aquestes classes.

Els atributs d'un objecte `Cluster` són:

```
BinTree<int> arq;  
vector<Procesador> chips;  
int n_procesador;
```

i el seu invariant de representació és:

```
chips.size() = n_procesador
arq té n_procesador elements, que són una permutació de [1..n_procesador]
```

Els atributs d'un objecte `Pendientes` són:

```
list <Proceso> pro_pen;
```

i el seu invariant de representació és:

```
els processos de pro_pen estan ordenats decreixentment per temps
i, en cas d'empat, creixentment per id
```

Haureu de fer servir una operació ja implementada de `Cluster`

```
void poner_proceso_en_procesador(int id_procesador, const Proceso & p, bool & puesto);
/* PRE : 1 <= id_procesador <= numero de elementos del p.i.; no hay ningún
proceso en el procesador id_procesador del p.i. con el mismo id que p */
/* POST : si p cabe en el procesador id_procesador del p.i., pasa a
ejecutarse en el dicho procesador y "puesto" es cierto, en caso
contrario el proceso no se ejecuta en el procesador y "puesto" es
falso */
```

així com diverses operacions de `Procesador` i `Proceso` que trobareu especificades als corresponents fitxers `.hh`.

Entrada

Una seqüència d'instruccions seguint el format de l'enunciat de l'examen i del joc de proves públic.

Sortida

El seu resultat seguint el format de l'enunciat de l'examen i del joc de proves públic.

Observació

El Jutge prova el vostre lliurament mitjançant 4 jocs de proves:

- `sample`: el joc de proves públic.
- `privat1`: joc de proves similar al públic.
- `privat2`: joc de proves privat que fa èmfasi en afegir processos pendents.
- `privat3`: joc de proves privat que fa èmfasi en afegir un procés al cluster.

Heu de lliurar un fitxer `solution.tar` creat amb la instrucció `make tar` definida al `Makefile` amb una implementació eficient de les operacions que es demanen.

Exemple d'entrada 1

```
configurar_cluster 10
5 3 2 0 8 0 0 0 1 9 0 0 7 4 6 0 0 0 10 0
10
20
30
40
50
10
20
30
40
50
10
agregar_proceso_pendiente 100 10 10
enviar_procesos_a_cluster 10
consultar_procesador 1
consultar_procesador 5
consultar_procesador 9
```


Procesador 4
105 20 69
Procesador 1

Procesador 2
103 60 8
109 30 20

Informació del problema

Autoria: PR02

Generació: 2026-01-25T21:04:32.168Z

© *Jutge.org*, 2006–2026.
<https://jutge.org>