
Exercici 4 de laboratori

Estructures de Dades i Algorítmica

Curs 2025/26

Índex

1	Presentació	2
1.1	Característiques del programa	2
2	Lliurament	2
3	Joc de proves i sortida	3
3.1	Fitxer de proves: volsCurt.txt	3
3.2	Sortida possible del fitxer de proves: volsCurt.txt	3

1 Presentació

L'aeroport de Girona ens ha demanat un programa per gestionar els vols i el seu ús de les portes d'embarcament a la terminal. S'ha de gestionar una sèrie de vols, $v = \{1, 2, \dots, n\}$, de cadascun dels quals sabem:

- la seva destinació (regional o internacional), i
- l'hora de sortida.

També sabem que l'aeroport:

- disposa de 2 portes amb control de passaports (aptes per a vols internacionals) i 3 més per a vols regionals.
- reserva la porta durant 60 minuts abans de la sortida de cada vol per al seu servei.

Hem de buscar una assignació de vols a portes (i horari) tenint en compte que:

- els vols internacionals només es podran embarcar des d'una porta amb control de passaport. Els vols regionals poden operar-se des de qualsevol porta.
- s'han d'assignar tots els vols a una porta d'embarcament compatible (que estigui disponible i pugui operar el tipus de vol) fent servir el **mínim nombre de portes possible**.

1.1 Característiques del programa

Així, heu de fer un programa que:

- Demani a l'usuari el nom del fitxer de dades i llegeixi els vols del fitxer indicat. Vegeu-ne un exemple a la Secció 3.1.
Podeu fer els preprocessaments de les dades que necessiteu.
- Implementi un **algorisme voraç** per resoldre el problema plantejat.
- Mostri per pantalla la **distribució de vols** en portes proposta. Vegeu-ne un exemple a la Secció 3.2.

2 Lliurament

El dia de l'entrega haureu de tenir el vostre programa a `bas.udg.edu`, preparat per compilar i executar. Això inclou:

1. el codi font (només fitxers `.cpp` i `.h`, res de fitxers objecte ni executables)
2. els fitxers del joc de proves (veure Sec. 3.2),
3. el fitxer `llegeix.me` on expliqueu quin objectiu té cada fitxer del joc de proves (i qualsevol altre comentari que vulgueu fer sobre el vostre codi).

IMPORTANT: Cal que seguiu les instruccions sobre com lliurar les activitats de laboratori que teniu a Moodle. Assegureu-vos que ho feu com us demanem, sobretot les pre- i postcondicions tant a les classes com al `main`.

3 Joc de proves i sortida

Cal que acompanyeu el vostre codi amb un joc de proves. En aquest cas, el joc de proves es redueix a diferents fitxers de dades. Com sempre, us donem un joc de proves bàsic que **heu de complementar**.

El joc de proves que compartim a Moodle consta de 2 fitxers de dades. També hi trobareu les corresponents sortides possible. Vigileu perquè, en aquest cas, segons com tracta el problema l'algoritme voraç que implementeu, la sortida que obteniu pot no ser semblant a la que us donem.

A continuació reproduïm el contingut de volsCurt.txt i la sortida de la nostra implementació de l'algoritme voraç corresponent.

3.1 Fitxer de proves: volsCurt.txt

```
id tipus hora
1 i 07:30
2 r 07:45
3 r 08:30
4 r 09:00
5 i 09:30
6 i 10:00
7 r 10:15
8 i 11:00
9 i 11:30
10 r 11:30
11 i 12:45
12 r 13:30
13 i 13:45
```

Podeu reaprofitar el codi de eines.h per llegir aquestes dades (el separador és el tabulador, '\t').

3.2 Sortida possible del fitxer de proves: volsCurt.txt

```
==> 13 vols llegits.

*****
* Porta 1 (INT): 6 vols *
*-----*
* Vol: 1 [07:30] REG    *
* Vol: 3 [08:45] INT    *
* Vol: 5 [09:45] INT    *
* Vol: 9 [11:30] INT    *
* Vol: 11 [12:45] REG   *
* Vol: 13 [13:45] INT   *
*****

*****
* Porta 2 (INT): 5 vols *
*-----*
* Vol: 2 [07:45] REG    *
* Vol: 4 [09:00] INT    *
* Vol: 6 [10:00] REG    *
* Vol: 10 [11:30] INT   *
* Vol: 12 [13:30] REG   *
*****

*****
* Porta 3 (REG): 2 vols *
*-----*
* Vol: 7 [10:15] REG    *
* Vol: 8 [11:15] REG    *
*****

Num. portes: 3 (INT.: 2)
```