

## Enunciat

La RENFE està fent unes enquestes per conèixer informació sobre els tipus de desplaçaments que fan els usuaris a les diverses línies a les que dona servei. Les persones enquestades han d'indicar la línia en la que viatgen, l'estació d'origen i la de destinació. RENFE vol fer, per cada línia, una classificació de l'ús del seu servei basat en el nombre de zones atravesades (NZA). El nombre de línies total és de 5 i a cada línia hi ha, com a màxim, 10 zones. El NZA es calcula en funció de l'estació origen i la destinació i un possible mètode és el que es dona al final de l'enunciat.

Es disposa d'un fitxer de nom "enquestes.txt" amb una seqüència d'enquestes. Cada enquesta consta de tres dades (3 enters) corresponents a la línia (valor entre 1 i 5), l'estació origen (valor entre 1 i 10) i la destinació (valor entre 1 i 10), respectivament. La seqüència acaba amb una enquesta sentinella en la que els 3 enters valen 0.

La RENFE vol disposar, en un fitxer de nom "resultats.txt", de la següent informació per cadascuna de les 5 línies:

- el número de la línia
- una llista de parells (NZA, nombre de persones), per cadascun dels possibles valors de NZA (entre 1 i 10) i ordenats creixentment
- el NZA promig i el NZA amb el nombre màxim de persones (moda)

Si es donés el cas que alguna de les línies no aparegués a cap enquesta, igualment s'han d'escriure tots els parells (NZA, nombre de persones) i al lloc del NZA promig i NZA moda s'hi ha d'escriure dos 11.

Es disposa de la funció següent que implementa el mètode per calcular el NZA entre dues estacions.

**funció** *NZAentreEstacions*(ent *eo, ed* : enter) **retorna** enter

{Pre: *eo* = *EO* i *ed* = *ED*}

{Post: *NZAentreEst*(*ep, ed*) és el nombre de zones que hi ha entre les estacions *EO* i *ED*}

**retorna** (*ed* - *eo* + 1)

**ffunció**

Dissenyeu un algorisme que obtingui l'esmentada informació.

## Guió de la pràctica

### Material de que es disposa

- Fitxers *fst.h* i *fst.c*, amb l'especificació i implementació del tipus FST
- Fitxers *taufreq.h* i *taufreq.c* amb amb l'especificació i implementació del tipus TauFreq
- Fitxers amb jocs de proves: *joc1.txt* i *joc2.txt*
- Fitxers amb els resultats esperats en els anteriors jocs de proves: *res1.txt* i *res2.txt*

### Guió

1. Dissenyeu l'algorisme que es demana a l'enunciat, aplicant l'anàlisi descendent apropiat.
  - Cal fer abstracció de l'objectes *enquesta* i *taula de línies*
  - Cal usar la funció *NZAentreEstacions*, donada a l'enunciat
  - Cal usar els tipus de dades *FST* i *TauFreq* donats
2. Traduiu a C l'anterior algorisme generant el fitxer *renfe.c* amb el programa principal i possibles subprogrames, els fitxers *enquesta.h* i *enquesta.c* amb l'especificació i implementació del tipus de l'objecte *enquesta* i els fitxers *taulin.h* i *taulin.c* amb l'especificació i implementació del tipus de l'objecte *taula de línies*. En el fitxer *renfe.c* hi haurà la funció *NZAentreEstacions* convenientment traduïda.

3. Compileu els fitxers `renfe.c`, `enquestes.c`, `taulin.c`, `fst.c` i `taufreq.c`
4. Munteu els fitxers objecte obtinguts
5. Executeu el programa resultant usant els jocs de proves “`joc1.txt`” i “`joc2.txt`” donats. Per cada prova cal que el fitxer d’entrada (“`joc*.txt`”) es digui “`enquestes.txt`” i comprovar que el fitxer que resulta de l’execució, “`resultats.txt`”, sigui igual que “`res*.txt`”.

**Exemple**

Pel següent joc de proves (“`joc1.txt`”):

```
5 2 3
5 3 4
2 1 3
2 3 5
1 1 2
1 2 4
3 4 5
3 1 4
1 1 5
3 1 5
2 2 5
5 1 2
0 0 0
```

El resultat és el següent (“`res1.txt`”):

```
1
1 0 2 1 3 1 4 0 5 1 6 0 7 0 8 0 9 0 10 0
3.33333 2
2
1 0 2 0 3 2 4 1 5 0 6 0 7 0 8 0 9 0 10 0
3.33333 3
3
1 0 2 1 3 0 4 1 5 1 6 0 7 0 8 0 9 0 10 0
3.66667 2
4
1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0
11 11
5
1 0 2 3 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0
2 2
```