

Poseu el nom a tots els fulls

Entregueu els problemes en fulls separats

Les respostes han de ser justificades

### Problema 1

5 punts

Temps estimat: 1h 15 min

La RENFE està fent unes enquestes per conèixer informació sobre els tipus de desplaçaments que fan els usuaris a la línia de la costa. Les persones enquestades han d'indicar l'estació d'origen i la de destinació. Les estacions estan agrupades per zones i RENFE vol fer una classificació de l'ús del seu servei basat en el nombre de zones travessades (NZA). A la regió on es fan aquestes enquestes hi ha 10 zones.

Es disposa d'un fitxer de nom "enquestes.txt" amb una seqüència d'enquestes. Cada enquesta consta de dues dades corresponents a l'estació origen i la destinació (de tipus *tEstacio*). La seqüència acaba amb dues estacions sentinelles.

La RENFE vol disposar, en un fitxer de nom "resultats.txt", de la següent informació corresponent a la línia enquestada: una llista de parelles (NZA, nombre de persones), per cadascun dels possibles valors de NZA (entre 1 i 10, ordenats creixentment, i apareixeran sempre encara que el nombre de persones sigui zero), el NZA promig i el NZA amb el nombre màxim de persones (moda).

Es disposa de les funcions que treballen sobre el tipus *tEstacio*:

**funció** *NZAentreEst*(ent *eo, ed* : *tEstacio*) **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*}

**acció** *LlegirEstacioFST*(entSOR *f* : *FST*, sor *e* : *tEstacio*)

{Pre: *mode(f) = lectura* i  $\neg df(f)$  i *pd(f) =  $\epsilon \cdot \gamma$*  i *pe(f) = PE*}

{Post: *e* representa  $\epsilon$ . *pd(f) =  $\gamma$* , *pe(f) = PE  $\cdot \epsilon$* }

**funció** *EstacioSent*(ent *e* : *tEstacio*) **retorna** *booleà*

{Pre: }

{Post: *EstacioSent(e)* és cert si i sols si *e* és l'estació sentinella}

Dissenyau un algorisme que obtingui l'esmentada informació.

Cal que useu el tipus Taula de Freqüències del qual s'adjunta l'especificació.

### Problema 2

3 punts

Temps estimat: 45 min

Volem ajustar una recta de regressió  $y = ax + b$  a un conjunt de mostres  $\{x_i, y_i\}, i = 1 \dots n$ . Les dades d'entrada provenen d'un fitxer de text anomenat "punts.txt", el qual conté una seqüència de parelles de coordenades reals acabada amb el sentinella  $-1, -1$ .

Recordem que el mètode de la regressió lineal simple per mínims quadrats determina que

$$a = \frac{\bar{x}\bar{y} - \frac{1}{n} \sum_{i=1}^n x_i y_i}{(\bar{x})^2 - \frac{1}{n} \sum_{i=1}^n x_i^2}$$

$$b = \bar{y} - a\bar{x}$$

essent  $\bar{x}, \bar{y}$  les mitjanes  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ .

Dissenyau un algorisme que a partir de les dades del fitxer trobi els paràmetres que determinen la recta, i els escrigui al canal estàndard de sortida.

### PROBLEMA 3 AL DORS DEL FULL

**Problema 3****2 punts****Temps estimat: 30 min**

Considerem la implementació del tipus *Matriu* que utilitza memòria dinàmica següent:

```
typedef struct {
    float **telem; /* Taula d'apuntadors de taules d'elements */
    int nf;        /* Nombre de files */
    int nc;        /* Nombre de columnes */
} Matriu;
```

Escriu un tros de programa en C que a partir de dues variables del tipus *Matriu*, *A* i *B*, que tenen el mateix nombre de columnes i el mateix nombre de files, i que ja han estat inicialitzades, construeixi una matriu *C* amb el mateix nombre de columnes que *A* i *B* i amb les files de *A* en les posicions parells (0, 2, ...) i les files de *B* en les posicions senars. A més les files de *A* estaran col·locades amb el mateix ordre que en la matriu original i, el mateix passarà amb les files de *B*. Després de l'operació, *A* i *B* seguiran existint. S'han de minimitzar el nombre d'operacions realitzades (reserva i alliberament de memòria, assignacions, etc.). Per exemple:

$$\begin{array}{r}
 \begin{array}{c}
 \overbrace{\begin{pmatrix} 1.0 & 2.0 & 3.0 \\ 4.0 & 5.0 & 6.0 \\ 7.0 & 8.0 & 9.0 \end{pmatrix}}^A, \quad \overbrace{\begin{pmatrix} 10.0 & 11.0 & 12.0 \\ 13.0 & 14.0 & 15.0 \\ 16.0 & 17.0 & 18.0 \end{pmatrix}}^B \\
 \end{array}
 \xrightarrow{\text{entrellaça}}
 \overbrace{\begin{array}{c}
 0 \begin{pmatrix} 1.0 & 2.0 & 3.0 \\ 10.0 & 11.0 & 12.0 \\ 4.0 & 5.0 & 6.0 \\ 13.0 & 14.0 & 15.0 \\ 7.0 & 8.0 & 9.0 \\ 16.0 & 17.0 & 18.0 \end{pmatrix} \\
 1 \\
 2 \\
 3 \\
 4 \\
 5
 \end{array}}^C
 \end{array}$$