

Poseu el nom a tots els fulls  
Lliureu els problemes en fulls separats  
Les respostes han de ser justificades

**Problema 1****5 punts****Temps estimat: 45 min**

Hem dissenyat un robot que es pot moure. El moviment del robot està programat de manera que en cada moment agafa, de manera aleatòria, un vector (3D) que, un cop normalitzat, pertany a un conjunt finit de vectors possibles. Volem estudiar el moviment que fa el robot. Per això disposem d'un fitxer seqüencial de text "direccions" que conté el conjunt de vectors (3D) normalitzats que ens dona **totes** les direccions i sentits possibles en que es pot moure el robot. El fitxer acaba amb el vector nul.

Fem moure el robot i els seus moviments s'enregistren en el fitxer "moviments" com una seqüència de vectors (no normalitzats) que descriuen el trajecte seguit. Els dos fitxers esmentats acaben amb el Vector nul.

Es demana que dissenyeu un algorisme, que a partir dels fitxers "direccions" i "moviments", obtingui un fitxer "tendències" on hi hagi una seqüència amb totes les parelles del tipus  $(\vec{v}_i, n_i)$  on  $\vec{v}_i$  és un vector de norma igual a 1 i  $n_i, n_i > 0$ , és el nombre de vegades en que el robot ha fet un moviment en una direcció i sentits equivalents al que representa  $\vec{v}_i$ . Cada  $\vec{v}_i$  només apareixerà un cop en el fitxer.

Disposem de

**acció** *llegirVectorFST*(entsor  $f : FST$ , sor  $v : Vector$ )

{**Pre:**  $f$  obert per lectura}

{**Post:**  $v$  representa el vector actual de  $f$ }

**funció** *vectorSentinella*(ent  $v : Vector$ ) **retorna booleà**

{**Pre:** **cert**}

{**Post:** *vectorSentinella*( $v$ ) és **cert** si  $v$  és el vector sentinella (vector nul). És **fals** en el cas que no ho sigui.}

**acció** *escriureVectorFST*(entsor  $f : FST$ , sor  $v : Vector$ )

{**Pre:**  $f$  obert per escriptura}

{**Post:** A la seqüència del fitxer  $f$  se li afegeix l'escriptura del vector  $v$ }

**funció** *normalitza*(ent  $v : Vector$ ) **retorna Vector**

{**Pre:**  $v = \vec{V}$  i  $\vec{V} = (V_0, V_1, \dots, V_{d-1})$  i  $V$  no és el vector nul ( $V \neq (0, 0, \dots, 0)$ )}

{**Post:** *normalitza*( $v$ ) =  $(\frac{V_0}{\|\vec{V}\|}, \frac{V_1}{\|\vec{V}\|}, \dots, \frac{V_{d-1}}{\|\vec{V}\|})$  on  $\|\vec{V}\| = \sqrt{\sum_{i=0}^{d-1} V_i^2}$ }

**acció** *inicialitzar*(sor  $d : TauDireccions$ )

{**Pre:** **cert**}

{**Post:**  $d$  és una taula de freqüències de direccions buida, és a dir, per tot  $\vec{v}_i$ ,  $freq(d, \vec{v}_i) = 0$ }

**acció** *afegirDireccio*(entsor  $d : TauDireccions$ , ent  $v : Vector$ )

{**Pre:**  $freq(d, v) = N \wedge N \geq 0$ }

{**Post:**  $freq(d, v) = N + 1$ }

**funció** *freq*(ent  $d : TauDireccions$ , ent  $v : Vector$ ) **retorna enter**

{**Pre:** **cert**}

{**Post:** *freq*( $d, v$ ) és el nombre de vegades de  $v$ }

**Problema 2****2 punts****Temps estimat: 15 min**

Dissenyau la funció *normalitza* del problema anterior. El vector normalitzat ( $\vec{n}$ ) del vector  $\vec{v}$  ( $\vec{v} = (v_0, v_1, \dots, v_{d-1})$ ) de dimensió  $d$  es calcula de la següent forma:

$$\vec{n} = \left( \frac{v_0}{\|\vec{v}\|}, \frac{v_1}{\|\vec{v}\|}, \dots, \frac{v_{d-1}}{\|\vec{v}\|} \right) \text{ on } \|\vec{v}\| = \sqrt{\sum_{i=0}^{d-1} v_i^2}$$

Disposeu de la funció

**funció** *arrel*( $r : \text{real}$ ) **retorna real**

{**Pre:**  $r = R$ }

{**Post:** *arrel*( $r$ ) =  $\sqrt{R}$ }

**PROBLEMA 3 AL DORS**

## Problema 3

3 punts

Temps estimat: 30 min

Es vol celebrar un congrés multitudinari a l'ETSEIB en el que serà necessari reservar les aules i els seus seients respectius tenint en compte també en quines plantes estan situades les aules. Per a fer-ho ens proposen la següent definició del tipus *AulesETSEIB*:

```
typedef struct {
    int nPlantes;
    AulesPlanta *planta;
} AulesETSEIB;

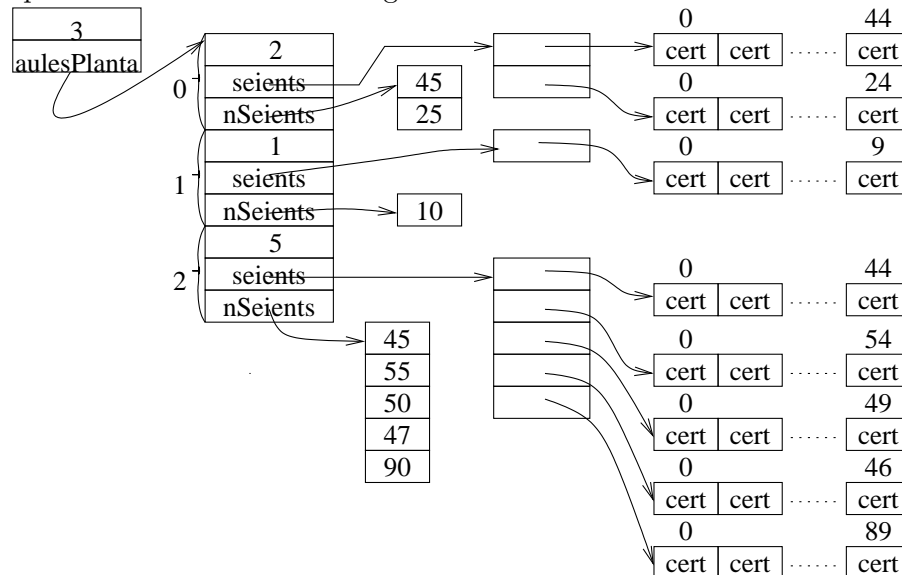
typedef struct {
    int nAules;
    int *nSeients;
    bool **seients;
} AulesPlanta;
```

La definició té en compte el nombre diferent d'aules que hi ha a cada planta. A més, cada aula, té un nombre diferent de seients. El camp *nSeients* és una taula de enters en la què a la posició *i-1* hi ha el número de seients de l'aula *i*. Les aules es numeren a partir d'1. El camp *seients* és una mena de matriu de booleans amb la particularitat que cada fila representa una aula *i* i per tant, pot tenir un número de seients diferent. Així, el booleà corresponent a la posició *i-1, j-1* de la matriu indica que el seient *j* de l'aula *i* està lliure si val cert i ocupat si val fals. Els seients es numeren a partir d'1. Es demana que dissenyeu el subprograma

```
AulesETSEIB CrearAulesETSEIB(void);
```

que llegeix del Canal Estàndard d'Entrada (CEE) el número de plantes reservades de l'ETSEIB seguit d'una seqüència de sub-seqüències per cada planta existent. Cada subseqüència consistirà amb el nombre d'aules que té la planta seguida d'una subseqüència del nombre de seients que hi ha a cada aula de la planta.

El subprograma haurà de fer la reserva l'espai necessari i inicialitzar tots els seients a lliure. Per exemple, si el CSE conté 3 2 45 25 1 10 5 45 55 50 47 90, *CrearAulesETSEIB* retorna l'objecte de tipus *AulesETSEIB* com el següent:



Cal que structureu el disseny amb 3 funcions:

- 1 Primer dissenyeu `AulesETSEIB CrearAulesETSEIB(void)`; usant la funció `AulesPlanta CrearPlantaETSEIB(void)`;
- 2 Dissenyeu `AulesPlanta CrearPlantaETSEIB(void)`; usant la funció `bool *CrearSeients(int n)`;
- 3 **No cal** que dissenyeu `bool *CrearSeients(int n)`; que crearà i inicialitzarà *n* seients.
- 4 Per tal de llegir enters del CEE, podeu utilitzar el subprograma `void LlegirEnter(int *const a)`;