

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

Problema 1**3.5 punts****Temps estimat: 45 min**

Un gran magatzem controla l'entrada i sortida dels productes que guarda mitjançant l'enregistrament del seu codi de barres. Per actualitzar el seu inventari disposa dels següents FSTs: El FST "entrada" conté una seqüència de codis de barres, un codi per cada unitat de producte que ha entrat al magatzem des del darrer inventari efectuat. El FST "sortida" conté una seqüència de codis de barres, un codi per cada unitat de producte que ha sortit del magatzem des del darrer inventari efectuat. Tant el FST "entrada" com el de "sortida" acaben amb un codi de barres sentinella. El FST "inventari" conté una seqüència de parelles (c, n) on c és el codi de barres d'un producte i n el nombre de productes del codi de barres associat que hi havia al magatzem abans d'efectuar-se les entrades i sortides. Aquest darrer FST conté els codis de barres de tots els possibles productes que poden entrar o sortir del magatzem i acaba amb una parella en que el seu codi de barres és el mateix sentinella que pel cas del FST "entrada".

Es demana dissenyar un algorisme que a partir dels FST "entrada", "sortida", i "inventari" descrits abans, construeixi un altre FST "NouInventari" que contingui els codis de barres amb el mateix ordre que el FST "inventari" juntament amb el nous nombres de productes resultants de les entrades i sortides efectuades. El nombre de parelles del FST "NouInventari" serà el mateix que el del FST "inventari".

acció *LlegirCodiBarresFST*(**entsor** $f : FST$, **sor** $cb : codiBarres$)¹

funció *CodiBarresSentinella*(**ent** $c : codiBarres$) **retorna booleà**

{**Pre:** cert}

{**Post:** *CodiBarresSentinella*(c) = cert si i només si c és el codi de barres sentinella}

acció *EscriureCodiBarresFST*(**entsor** $f : FST$, **ent** $cb : codiBarres$)

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

{**Post:** Escriu al fitxer f el codi de barres cb }

acció *Inicialitzar*(**sor** $t : TauFreq$)

{**Pre:** cert}

{**Post:** t és una taula de freqüències buida}

acció *Afegir*(**entsor** $t : TauFreq$, **ent** $cb : codiBarres$)

{**Pre:** $Freq(t, cb) = F$ }

{**Post:** $Freq(t, cb) = F + 1 \wedge$ la freqüència en t de la resta de codis de barres no ha canviat}

funció *Freq*(**ent** $t : TauFreq$, **ent** $cb : codiBarres$) **retorna enter**

{**Pre:** cert}

{**Post:** $Freq(t, cb)$ és la freqüència d' cb en t }

¹ Precondicions i Postcondicions similars a *LlegirEnterFST*, canviant **enter** pel tipus *codiBarres*.

Problema 2**3.5 punts****Temps estimat: 45 min**

Es vol dissenyar una acció que resolgui un sistema de n equacions amb n incògnites mitjançant el mètode de Cramer amb la següent especificació:

acció *resolSistemaCramer*(**ent** $a : Matriu$, **ent** $b : Vector$, **sor** $x : Vector$)

{**Pre:** $FilesMatriu(a) = ColsMatriu(a) = DimVector(b) = DimVector(x)$ i $|a| \neq 0$ }

{**Post:** x és tal que $a \cdot x = b$ }

El mètode de Cramer, a partir d'un sistema d'equacions com

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}}_A \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}}_x = \underbrace{\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}}_b$$

indica que cal fer el següent: Cada component x_i del vector \mathbf{x} es calcula fent $x_i = \frac{|B_i|}{|A|}$ i B_i és una matriu quadrada $n \times n$ tal que tots els seus components excepte els de la columna i coincideixen amb els components de la matriu A , i els components que hi han a la columna i de B_i coincideixen amb els del vector de termes independents \mathbf{b} . És a dir,

$$B_i = \begin{pmatrix} a_{11} & \cdots & a_{1,i-1} & b_1 & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2,i-1} & b_2 & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,i-1} & b_n & \cdots & a_{nn} \end{pmatrix}$$

Dissenyeu l'acció *resolSistemaCramer* que porti a terme el mètode exposat tenint en compte que disposeu de la següent funció:

funció *Determinant*(ent a : Matriu) retorna real

{Prec: $FilesMatriu(a) = ColsMatriu(a)$ }

{Post: $Determinant(a) = |a|$ }

NOTA: No està permès en aquest problema fer assignacions directes entre una variable de tipus *Matriu* i una variable o expressió de tipus *Matriu*.

Problema 3

3 punts

Temps estimat: 30 min

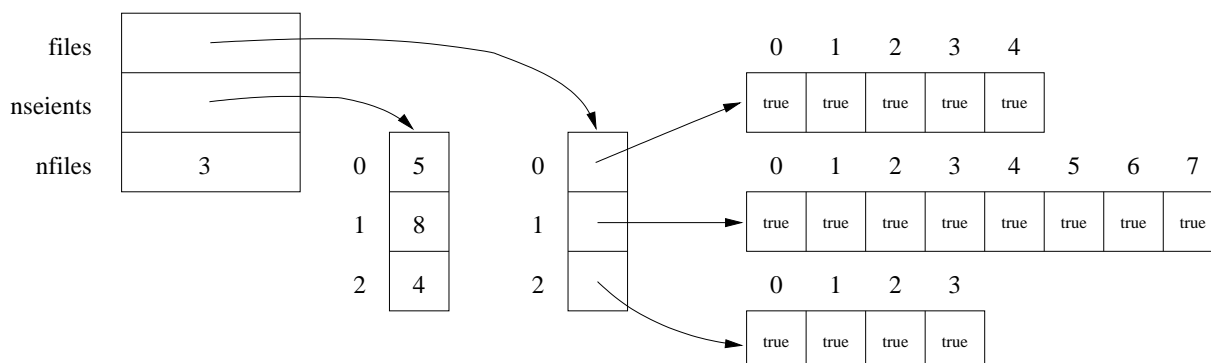
Per tal de conèixer quins són els seients ocupats d'un teatre, ens proposen la definició del tipus *Teatre* següent:

```
typedef struct {
    bool **files;
    int *nseients;
    int nfiles;
} Teatre;
```

Aquesta definició de tipus serveix per teatres on el número de seients pot ser diferent per cada fila. Així, el camp *nseients* és una taula d'enters en la que a la posició $i-1$ hi ha el número de seients de la fila i . Les files es numeren a partir d'1. El camp *files* és una matriu de booleans amb la particularitat que cada fila 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 la fila 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

```
Teatre CrearTeatre(void);
```

que llegeix del Canal Estàndard d'Entrada el número de files del teatre i la seqüència de número de seients de cada fila, reserva l'espai necessari i inicialitza tots els seients a lliure. Per exemple, si el CSE conté 3 5 8 4, *CrearTeatre* retorna un objecte de tipus *Teatre* com el següent:



Per tal de llegir enters, podeu utilitzar el subprograma

```
void LlegirEnter(int *const a);
```