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: 60 min

Un centre de càlcul disposa de dos servidors als quals es connecten els seus usuaris. Cada vegada que un usuari es connecta al servidor 1, s'enregistra la data, l'hora i l'identificador de l'usuari al fitxer "login-1.log". El fitxer "login-2.log" conté la mateixa informació, però per al servidor 2. Els identificadors d'usuari són enters entre 1000 i 10000. Els fitxers "login-1.log" i "login-2.log" acaben amb un element que conté una data i hora arbitràries i un identificador d'usuari igual a -1. Per tal de millorar el servei, els responsables del centre de càlcul volen saber el nombre de vegades que s'ha connectat cada usuari en un interval de dies.

Es demana que dissenyeu un algorisme que donat un interval de dies pel Canal Estàndard d'Entrada (CSE) i donats els fitxers "login-1.log" i "login-2.log", escrigui pel Canal Estàndard de Sortida (CSS) una seqüència de parelles formada per un identificador d'usuari i un enter que indica el nombre de vegades que ha entrat al sistema aquest usuari, és a dir, el nombre de vegades que ha entrat al servidor 1 més el nombre de vegades que ha entrat al servidor 2. Només s'han d'escriure els usuaris que han entrat un cop com a mínim. Per exemple, per l'interval de temps del 2/1/2006 al 9/1/2006, i la resta de dades següents:

CSE	fitxer "login-1.log"	fitxer "login-2.log" és
2/1/2006	2/1/2006 10:30 1024	4/1/2006 21:33 2048
9/1/2006	2/1/2006 12:43 5734	6/1/2006 10:21 1024
	3/1/2006 19:28 1024	12/1/2006 8:43 2048
	5/1/2006 14:01 1024	13/1/2006 9:12 -1
	7/1/2006 17:25 2048	
	10/1/2006 8:20 1024	
	11/1/2006 2:12 -1	

el resultat que ha d'escriure l'algorisme al CSS és 1024 4 2048 2 5734 1

Es disposa dels tipus *Data*, *Hora* i *TauFreq*, i dels subprogrames següents:

acció $LlegirData(\textbf{sor}\ d : Data)$[1]

acció $LlegirDataFST(\textbf{entsor}\ f : FST, \textbf{sor}\ d : Data)$[2]

acció $LlegirHoraFST(\textbf{entsor}\ f : FST, \textbf{sor}\ h : Hora)$[2]

funció $DinsInterval(\textbf{ent}\ ii : Data, \textbf{ent}\ if : Data, \textbf{ent}\ d : Data)$ **retorna booleà**

{**Pre: cert**}

{**Post:** $ii \leq d \leq if$, és a dir, la data d està entre ii i if}

acció $Inicialitzar(\textbf{sor}\ t : TauFreq)$

{**Pre: cert**}

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

acció $Afegir(\textbf{entsor}\ t : TauFreq, \textbf{ent}\ uid : \textbf{enter})$

{**Pre:** $1000 \leq uid \leq 10000$ i $Freq(t, uid) = F$}

{**Post:** $Freq(t, uid) = F + 1\ \wedge$ la freqüència en t de la resta d'uids no ha canviat}

funció $Freq(\textbf{ent}\ t : TauFreq, \textbf{ent}\ uid : \textbf{enter})$ **retorna enter**

{**Pre:** $1000 \leq uid \leq 10000$}

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

[1] Precondicions i Postcondicions similars a LlegirEnter, canviant **enter** pel tipus Data o Hora.

[2] Precondicions i Postcondicions similars a LlegirEnterFST, canviant **enter** pel tipus Data o Hora.

Problema 2 *3.5 punts* Temps estimat: 45 min

Dissenyeu un algorisme que donada una matriu quadrada M, $n \times n$, pel CEE i un FST, "vectors.dat" que conté vectors de dimensió n, indiqui si hi ha algun vector v_i, de "vectors.dat", tal que satisfà que l'expressió $v_i + M \cdot v_i$ pertanyi al Núcli de la matriu M. L'algorisme escriurà 'S' en el cas de que existeix l'esmentat vector v_i i 'N' en cas contrari. Es valorarà el disseny descendent plantejat.

A més de les accions i/o funcions que teniu en el formulari, també podeu disposar de:

acció $LlegirVectorFST($**entsor** $f : FST,$ **sor** $v : Vector)$
{**Pre:** f obert en mode lectura $\wedge$ queda algun vector per llegir}
{**Post:** p és el següent vector en f}
funció $VectorSentinella($**ent** $v : Vector)$ **retorna booleà**
{**Pre: cert**}
{**Post:** $VectorSentinella(v)$ és **cert** si i només si v és el vector sentinella}
acció $LlegirMatriu($**sor** $M : Matriu)$
{**Pre: cert**}
{**Post:** M és una matriu $n \times n$ amb valors inicialitzats}

Problema 3 *3 punts* Temps estimat: 30 min

En una aplicació gràfica de dibuix de polígons es té la següent definició pel tipus *Poligon*:

```
typedef struct{
  float x;
  float y;
}Vertex;

typedef struct{
        int nVertexs;    /* nombre de vèrtexs */
        Vertex *vPol; /* taula de vèrtexs */
}Poligon;
```

Implementeu les operacions:

```
Poligon CrearPoligon(int nVertexs)
/*  {Pre: nVertexs = N}                                                          */
/*  {Post: Creat el poligon nul de N vèrtexs. Tots els seus vèrtex són (0,0)}    */
 AssignaVertex(Poligon  * const p , int i, Vertex v)
/*  {Pre: p = (X0,Y0)(X1,Y1)...(Xn-1,Yn-1) i 0 <= i <= n i v = (X,Y) i i = I}    */
/*  {Post: p = (x0,y0)(x1,y1)...(xn-1,yn-1) i (xI,yI) = (X,Y) i per tot j <> I,  */
/*   0 <= j <= n es té (xj,yj) = (Xj,Yj)}                                        */
void DestruirPoligon(Poligon *const p)
/*  {Pre: cert}                                                                  */
/*  {Post: Destruit el poligon p}                                                */
```

Usant la implementació anterior i tenint em compte la següent definició d'estructura de dades que representa un dibuix,

```
typedef struct{
        int n;       /* nombre de poligons */
        Poligon *p; /* taula de poligons   */
} Dibuix;
```

implementeu les operacions:

```
Dibuix CrearDibuix(int num, int nVertexs)
/*  {Pre: num = N i nVertexs = NV}                                               */
/*  {Post: Creat un dibuix de N poligons nuls de NV vèrtexs}                     */
void AssignaVertexPolDibuix(Dibuix * const d, int i, int j, Vertex v)
/*  {Pre: s = P0,P1,P2,...Pn i 0 <= k <= n i i = I i j = J i v = V i Pk és un polígon} */
/*  {Post: s = p0,p1,...pn i per tot k <> i, 0 <= k <= n es té pk = Pk i pI és el resultat */
/*   d'aplicar AssignaVertex(PI,J,V)}                                            */
void DestruirDibuix(Dibuix *const d)
/*  {Pre: cert}                                                                  */
/*  {Post: Destruit el dibuix d}                                                 */
```