

Poseu el nom a tots els fulls

Entregueu els problemes en fulls separats

Les respostes han de ser justificades

Problema 1

4 punts

Temps estimat: 1h 15 min

El fitxer de text "verbs.txt" conté una seqüència de paraules, on cada paraula correspon a l'arrel d'un verb català. Altrament dit, conté les arrels de tots els verbs catalans. El fitxer de text "tirant.txt" conté una seqüència de paraules corresponent a la novel·la *Tirant lo Blanc*. Donats els dos fitxers anteriors, dissenyeu un algorisme que escrigui en el fitxer de text "freq.txt" el nombre de vegades que apareix cada verb català en el *Tirant*. El fitxer "freq.txt" ha de contenir parelles (verb, freqüència) on el verb també està en "verbs.txt" i ha aparegut com a mínim un cop en el *Tirant*. L'escriptura en el fitxer "freq.txt" s'ha de fer amb l'ordre de verbs que hi ha al fitxer "verbs.txt".

Disposeu dels tipus i subprogrames següents:

tipus *Paraula* ftipus

acció *LlegirParaulaFST*(entsor $f : FST$, sor $p : Paraula$)

{Pre: f obert en mode lectura \wedge queda alguna paraula per llegir}

{Post: p és la següent paraula en f }

funció *ParaulaSentinella*(ent $p : Paraula$) retorna booleà

{Pre: cert}

{Post: *ParaulaSentinella*(p) = cert si i només si p és la paraula sentinella}

funció *EsVerbParaula*(ent $p : Paraula$) retorna booleà

{Pre: cert}

{Post: *EsVerbParaula*(p) = cert si i només si p és un verb}

funció *ArrelParaula*(ent $p : Paraula$) retorna *Paraula*

{Pre: cert}

{Post: *ArrelParaula*(p) és l'arrel de p }

acció *EscriureParaulaFST*(entsor $f : FST$, ent $p : Paraula$)

{Pre: f obert per escriptura}

{Post: Escriu al fitxer f la paraula p }

tipus *TauFreq* ftipus

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

{Pre: cert}

{Post: t és una taula de freqüències buida}

acció *Afegir*(entsor $t : TauFreq$, ent $x : Paraula$)

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

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

funció *Freq*(ent $t : TauFreq$, ent $x : Paraula$) retorna enter

{Pre: cert}

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

El tipus *TauFreq* NO té cap més subprograma que els anteriors (llevat dels que especifiqueu i dissenyeu vosaltres).

Problema 2

3 punts

Temps estimat: 45 min

Dissenyar un algorisme que donada una matriu quadrada A , $n \times n$, pel CEE i un FST, "vectors.dat" que conté vectors de dimensió n , construeixi un nou FST format pels vectors de "vectors.dat" que pertanyin al Nucli de la matriu A .

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

{Pre: f obert en mode lectura \wedge queda algun vector per llegir}

{Post: v 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 *EscriureVectorFST*(entsor $f : FST$, ent $v : Vector$)

{Pre: f obert en mode escriptura}

{Post: v s'ha escrit al final d' f }

funció *ModulVector*(ent v : *Vector*) **retorna** *real*

{Pre: cert}

{Post: *ModulVector*(v) és $\|v\|$ }

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

Ens demanen un programa en C tal que donats pel CEE un enter parell n i els coeficients d'una matriu A $n \times n$ donats per files, construeixi i guardi dues submatrius d' A : la submatriu (A_p) formada pels coeficients que estan en una fila parell i un columna parell i la submatriu (A_s) formada pels coeficients de posicions fila senar i columna senar.

Per exemple,

$$\text{A partir de } \overbrace{\begin{pmatrix} 00.0 & 01.0 & 02.0 & 03.0 & 04.0 & 05.0 \\ 10.0 & 11.0 & 12.0 & 13.0 & 14.0 & 15.0 \\ 20.0 & 21.0 & 22.0 & 23.0 & 24.0 & 25.0 \\ 30.0 & 31.0 & 32.0 & 33.0 & 34.0 & 35.0 \\ 40.0 & 41.0 & 42.0 & 43.0 & 44.0 & 45.0 \\ 50.0 & 51.0 & 52.0 & 53.0 & 54.0 & 55.0 \end{pmatrix}}^A \text{ es tindrà } \overbrace{\begin{pmatrix} 11.0 & 13.0 & 15.0 \\ 31.0 & 33.0 & 35.0 \\ 51.0 & 53.0 & 55.0 \end{pmatrix}}^{A_s} \text{ i } \overbrace{\begin{pmatrix} 00.0 & 02.0 & 04.0 \\ 20.0 & 22.0 & 24.0 \\ 40.0 & 42.0 & 44.0 \end{pmatrix}}^{A_p}$$

Un equip de programadors ja ha decidit la implementació a fer. Hi hauran dos representacions de tipus:

MatriuQ per les matrius quadrades:

```
typedef struct {
    float **telem; /*Taula de taules de reals*/
    int n;         /*No. de files o columnes */
} MatriuQ;
```

MatriuQPS per una taula de matrius quadrades:

```
typedef struct {
    MatriuQ *m; /*Taula de matrius quadrades */
    int n;      /*Nombre de matrius quadrades*/
} MatriuQPS;
```

... I ja s'ha fet el següent tros de programa:

```
.....
MatriuQPS Ms;
int i, j, n;
float r;

LlegirEnter(&n);
Ms = CreaMatriuQPS(2, n / 2);
i = 0;
while (i < n) {
    j = 0;
    while (j < n) {
        LlegirReal(&r);
        AssigMatriuQPS(&Ms, i, j, r);
        j = j + 1;
    }
    i = i + 1;
}
.....
```

L'analista en cap ens demana que implementem en C els subprogrames:

MatriuQ *CreaMatriuQ*(int n)

/* Pre: CERT */

/* Post: crea una matriu $n \times n$ */,

MatriuQPS *CreaMatriuQPS*(int n, int dmq)

/* Pre: CERT */

/* Post: creada una taula de n matrius $dmq \times dmq$ */,

void *AssigMatriuQPS*(*MatriuQPS* const * *Matriu*, int i, int j, float r)

/* Pre: *Matriu* conté dos matrius quadrades, r és el valor a_{ij} d'una matriu A . */

/* Post: Si i i j són parells, r es posarà a la matriu 1 (A_p) de *Matriu* en la posició corresponent, si i i j són senars, r es posarà a la matriu 0 (A_s) de *Matriu* en la posició corresponent. En el cas en que les paritats d' i i j siguin diferents, no es farà res.

*/