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

L'ajuntament està fent un estudi sobre el tràfic existent en els carrils-bici de la ciutat. Per fer-ho, s'han instal·lat sensors en diversos punts geogràfics de la ciutat. Quan un sensor detecta el pas d'una bicicleta, ho notifica a un ordinador central. Al final del dia es té un fitxer text "senyals.txt" consistent en una seqüència de paraules que identifiquen indrets geogràfics i que es repeteixen en la seqüència cada cop que ha passat una bicicleta en el lloc geogràfic en qüestió.

Ens demanen fer un programa que donat el fitxer text "senyals.txt" i un fitxer text anomenat "carrilDiagonal.txt" que conté només els identificadors d'indrets geogràfics que pertanyen al carril bici de la Diagonal, tregui pel canal standard de sortida la mitjana de tràfic del carril de la Diagonal, i quin indret d'aquest carril hi ha més tràfic de bicicletes.

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}

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

Donada una matriu tridiagonal A, dissenyeu un subprograma que en faci l'eliminació gaussiana el més eficient possible, és a dir, fent el nombre mínim de sumes, multiplicacions i divisions.

Una matriu tridiagonal és una matriu en la qual els únics elements diferents de zero en cada fila (i) són l'element de la diagonal (a_{ii}) i els de les columnes anterior ($a_{i,i-1}$) i posterior ($a_{i,i+1}$):

$$\begin{array}{llllll}
a_{00}x_0+ & a_{01}x_1 & & & & = b_0 \\
a_{10}x_0+ & a_{11}x_1+ & a_{12}x_2 & & & = b_1 \\
 & a_{21}x_1+ & a_{22}x_2+ & a_{23}x_3 & & = b_2 \\
\dots & & & & & \dots \\
 & & & a_{n-1,n-2}x_{n-2} & +a_{n-1,n-1}x_{n-1} & = b_n
\end{array}$$

Recordeu que l'eliminació gaussiana es calcula per $k = 0, 1, \dots, n-2$, aplicant les fórmules següents:

$$\left.\begin{array}{l} a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} \cdot a_{kj}^{(k)} \\ m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \end{array}\right\} \begin{array}{l} \text{per } i = k+1, \dots, n-1 \\ \text{i per } j = k+1, \dots, n-1 \end{array}$$

Podeu suposar que $a_{kk}^{(k)} \neq 0$ sempre i que no cal aplicar pivotatge.

Noteu que aquestes fórmules són generals i es poden simplificar en el cas de les matrius tridiagonals.

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

Donada aquesta definició de tipus per l'estructura de dades Polinomi:

```
typedef struct{
        int grau;    /* grau màxim */
        float *coef; /* Es guarden tots els coeficients del polinomi */
}Polinomi;
```

Implementeu les operacions:

`Polinomi CrearPolinomi(int grau)`
/* {**Pre:** $grau = G$} */
/* {**Post:** Creat el polinomi nul de grau màxim G $0 \cdot x^G + 0 \cdot x^{G-1} + \ldots + 0 \cdot x + 0$} */
`void AssignaCoef(Polinomi *const p,int exp, float coef)`
/* {**Pre:** $p = A_0 + A_1x + A_2x^2 + \ldots + A_nx^n$ i $0 \leq exp \leq n$ i $exp = E$ i $coef = C$} */
/* {**Post:** $p = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ i $a_E = C$ i per tot $i \neq E, 0 \leq i \leq n$ es té $a_i = A_i$} */
`void DestruirPolinomi(Polinomi *const p)`
/* {**Pre: cert**} */
/* {**Post:** Destruit el polinomi p} */

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

```
typedef struct{
        int n;       /* nombre de polinomis */
        Polinomi *p; /* taula de polinomis   */
}SistemaPolinomi;
```

implementeu les operacions:

`SistemaPolinomi CrearSistemaPolinomi(int num, int g)`
/* {**Pre:** $num = N$ i $g = G$} */
/* {**Post:** Creat un sistema de N polinomis nuls de grau màxim G} */
`void AssignaCoefPolSistemaPolinomi(SistemaPolinomi *const s,int k, int exp, float coef)`
/* {**Pre:** $s = P_0, P_1, P_2, \ldots P_n$ i $0 \leq k \leq n$ i $k = K$ i $exp = E$ i $coef = C$ i $P_k = A_0 + A_1x + \ldots + A_gx^g$ on g és el grau màxim del polinomi} */
/* {**Post:** $s = p_0, p_1, \ldots p_n$ i per tot $i \neq K, 0 \leq i \leq n$ es té $p_i = P_i$ i $P_K = a_0 + a_1x + a_2x^2 + \ldots + a_gx^g$ i $a_E = C$ i per tot $i \neq E, 0 \leq i \leq n$ es té $a_i = A_i$} */
`void DestruirSistemaPolinomi(SistemaPolinomi *const s)`
/* {**Pre: cert**} */
/* {**Post:** Destruit el sistema de polinomis s} */