

S'ha de posar el nom, cognoms, DNI i especialitat a cada full que s'entregui.
 Qualsevol hipòtesi que es faci sobre un problema s'haurà d'argumentar.
 És indispensable fer cada problema en un full separat. Si algun problema no es fa, s'ha d'entregar igualment el corresponent full en blanc amb les dades personals.

Problema 1 (3.5 punts)

La Loteria EuroMillions ens sol·licita una extensió del programa per a la gestió dels seus sorteigs i de les apostes dels jugadors. Els jugadors poden escollir la data del sorteig que desitgen participar per fer les seves apostes. Tals apostes consisteixen en 5 nombres de l'1 al 50, podent-se obtenir premis de diferent valor segons s'encertin 3, 4 o 5 nombres en qualsevol ordre de la combinació guanyadora. Les dades de les apostes i els sorteigs estan ja estructurades de la següent manera:

Una llista (estructura tipus `LApostes`) per emmagatzemar 10000 apostes, on cada aposta (estructura tipus `Aposta`) emmagatzema el DNI del jugador, la data en què desitja jugar (format `aaaammdd`), la seva aposta (llista de 5 nombres), i un camp per emmagatzemar el valor del premi obtingut (o un 0 si no obté premi).
 Una altra llista (estructura tipus `LSorteigs`) per emmagatzemar 30 sorteigs, on cada sorteig (estructura tipus `Sorteig`) emmagatzema:

- la data de realització del sorteig;
- una llista de 5 nombres que és la combinació guanyadora;
- tres valors reals que indiquen la quantitat a percebre per encertar 3, 4 o 5 nombres de la combinació guanyadora.

```
const int A=10000, S=30;
```

```
struct Aposta{
    int data; //format aaaammdd
    string dni;
    float teupremi;
    int bid[5]; //aposta de números entre 1 i 50
};
typedef Aposta LApostes[A];

struct Sorteig{
    int data;
    int ComboWin[5]; //combinació guanyadora
    float p3,p4,p5; //premi per als encerts de 3,4 o 5 números
};
typedef Sorteig LSorteigs[S];
```

Es demanen:

A. [35%] Un subprograma que passat un nombre, entre 1 i 50, i un sorteig (estructura tipus `Sorteig`), retorni un valor per indicar si el nombre està dins de la combinació guanyadora del sorteig o no.

B. [65%] Un subprograma que passada una aposta (estructura tipus `Aposta`) i qualsevol altra informació necessària com a paràmetre, localitzi el sorteig corresponent a la data escollida en l'aposta i que:

- si no hi ha sorteig en aquesta data emeti el següent missatge per pantalla: "No hi ha sorteig per a la data", o
- si existeix el sorteig, verifiqui, utilitzant el subprograma de l'apartat A, quants dels 5 nombres de l'aposta apareixen en la combinació guanyadora del sorteig i, segons sigui el cas, s'introdueixi el valor del premi corresponent (o un 0 si no té premi) en el camp `teupremi` de l'aposta.

Problema 2 (2.5 punts)

Una matriu quasi-màgica es una matriu quadrada (N files x N columnes), on la suma dels elements de cada fila, de cada columna i de la diagonal principal sumen el mateix i aquesta suma s'anomena constant màgica. Un exemple de matriu quasi-màgica de 3 files x 3 columnes seria la següent:

8	1	6
3	5	7
4	9	2

On la constant màgica és 15, és a dir, la suma dels elements de qualsevol fila, columna o la diagonal principal és 15. Es demana un subprograma que donada una matriu quadrada de dimensions $N \times N$ d'enters positius retorni algun valor que informi si la matriu és quasi-màgica o no i que:

- Si ho és, retorni també el valor de la constant màgica.
- Si no ho retorni el valor 0.

Podeu deixar el nombre de files i columnes en funció d' N .

Problema 3 (4 punts)

La nova ESEIAAT ens ha posat en el compromís de gestionar amb un programa de C++ les notes dels 50 alumnes matriculats en les 10 assignatures de fase inicial del grau. El programa treballarà amb dues entitats (llistats):

- 1) El llistat dels 50 alumnes matriculats en fase inicial. De cada alumne emmagatzemarem:
 - a) el seu ID únic (DNI o passaport).
 - b) el nombre N d'assignatures de fase inicial matriculades.
 - c) Un llistat de resultats d'aquestes assignatures (amb màxim 10 elements, però realment amb dades de N assignatures). Cada element d'aquest llistat inclourà dues coses:
 - c1) el codi de l'assignatura.
 - c2) la nota final (0-10) de l'alumne en aquesta assignatura.
- 2) El llistat de les 10 assignatures de la fase inicial del grau. De cada assignatura emmagatzemarem:
 - a) el seu codi (p. ex., el codi de Fonaments d'Informàtica és el 320095)
 - b) el nom del professor responsable (un sol string).

Se'ns sol·licita:

A. [35%] Les estructures de dades adequades per gestionar la informació del programa.

B. [35%] Un subprograma que, donat el codi d'una assignatura (i qualsevol altra informació necessària), calculi i retorni la nota final mitjana dels alumnes matriculats en la mateixa.

C. [30%] Un programa complet que:

c.1) en primer lloc, cridi a "**fase_ini_llegeix**", un subprograma que NO heu de codificar i en el qual suposarem que es llegeix i guarda tota la informació sobre alumnes i assignatures.

c.2) A continuació ofereixi a l'usuari un menú de dues opcions:

- La 1a. permet a l'usuari veure la nota mitjana d'una assignatura. Si l'usuari la tria, el programa haurà de demanar-li que introdueixi el codi de l'assignatura per teclat, després de la qual cosa mostrarà per pantalla la nota mitjana calculada a partir de tots els estudiants matriculats en la mateixa; per a això, obligatòriament s'haurà d'usar el subprograma de l'apt. **B**.

- La 2a. permet a l'usuari veure informació general de les assignatures. Si l'usuari la tria, el programa mostrarà per pantalla la nota mitjana de cadascuna de les assignatures (usant de nou obligatòriament el subprograma de l'apt. **B**.), així com el nom del/la professor/a de l'assignatura.

- Si l'usuari no tria ni la primera ni la segona (és a dir, si s'equivoca), el programa tan solament mostrarà per pantalla el missatge "opció incorrecta".