

## Tema 5d: Registres

---

**Problemes marcats en groc** → Problemes avaluables

La resta de problemes són de dificultat intermèdia entre els avaluables. Si trobeu dificultats en passar d'un problema avaluable al següent, feu algun/s dels problemes no marcats com avaluables per a tenir un aprenentatge més progressiu.

### Problema 1

Crear un registre anomenat TipusAlumne amb els següents camps:

Nom	String de 10 elements
Cognom1	String de 15 elements
Cognom2	String de 15 elements
Nota	Real

Fer un programa que declari una variable de tipus TipusAlumne i demani a l'usuari que introdueixi els valors necessaris per a inicialitzar els camps de la variable declarada. Després, el programa ha d'imprimir per pantalla els valors dels camps introduïts en el format:

<cognom1> <cognom2>, <nom> <nota>

### Problema 2

Crear un registre anomenat TipusAlumne amb els següents camps:

Nom	String de 10 elements
Cognom1	String de 15 elements
Cognom2	String de 15 elements
Nota	Real

Fer un programa que declari un array de dimensió 5 del tipus TipusAlumne i demani a l'usuari que introdueixi els valors necessaris per a inicialitzar els camps de totes les posicions de l'array. Després, el programa ha d'imprimir per pantalla els valors dels camps de tots els registres de l'array en el format:

<cognom1> <cognom2>, <nom> <nota>

i un alumne/a per línia.

### Problema 3

Declarar un nou tipus de dades, TLibre, com un registre amb els camps: títol, editorial, pàgines i preu. A continuació, fer un programa on es declari dues variables del nou tipus de dades i es demani per teclat a l'usuari les dades dels seus dos llibres preferits (primer un i després l'altre) per omplir les dues variables que s'han creat. Per últim, mostrar per pantalla el contingut dels dos registres per a comprovar que s'han guardat les dades correctament. El format del missatge de sortida ha de ser:

Llibres preferits:

<títol 1> <editorial 1> <pàgines 1> <preu 1>  
<títol 2> <editorial 2> <pàgines 2> <preu 2>

#### Problema 4

Declarar un nou tipus de dades, TEquip, com un registre on guardar informació sobre equips de la lliga de futbol (nom de l'equip, ciutat d'on és, punts que porta a la lliga, pressupost de l'equip en milions d'euros). El registre ha de tenir els camps: nom, ciutat, punts i pressupost.

Fer els procediments i funcions següents:

- Procediment **OmplirDadesEquip** que rep com a paràmetre un registre de tipus TEquip, demana a l'usuari les dades d'un equip i omple els camps del registre passat com a paràmetre.
- Procediment **MostrarDadesEquip** que rep com a paràmetre un registre de tipus TEquip i mostra per pantalla els valors dels seus camps (en la mateixa línia de la pantalla i amb els camps separats per un espai).
- Funció **CompararEquips** que rep com a paràmetre dos registres de tipus TEquip i retorna un enter. La funció compara els punts a la lliga que porten els dos equips i retorna -1 si els punts del primer equip són menys que els punts del segon equip, 0 si els dos equips porten els mateixos punts i 1 si el primer equip porta més punts que el segon.

Al programa principal, declarar dues variables de tipus TEquip i utilitzar el procediment OmplirDadesEquip per a omplir els dos registres. A continuació, utilitzar el procediment MostrarDadesEquip per a mostrar el contingut dels registres per pantalla en format:

```
<equip 1> <ciutat 1> <punts 1> <pressupost 1>  
<equip 2> <ciutat 2> <punts 2> <pressupost 2>
```

Per últim, utilitzant la funció CompararEquips, comprovar quin equip té més punts a la lliga i escriure missatge "XXXX te <mes/menys/els mateixos> punts a la lliga que YYYY", on XXXX i YYYY seran els noms dels equips 1 i 2 respectivament.

#### Problema 5

Declarar una estructura de dades per a representar números complexos. A continuació, fer un programa que simuli una calculadora de nombres complexos que permeti sumar, restar i multiplicar dos nombres complexos i calcular el conjugat d'un nombre complex.

El programa ha de presentar un menú amb totes les opcions possibles i demanar a l'usuari que triï una opció. A continuació s'ha d'efectuar l'operació seleccionada i mostrar el resultat per pantalla. S'ha de permetre que l'usuari faci tantes operacions com vulgui, fins que seleccioni l'opció d'acabar. Implementeu cada operació (suma, resta, multiplicació i conjugat) com un procediment que rebí els registres necessaris per a l'operació i per a retornar el resultat de l'operació.

#### Problema 6

Declarar un nou tipus de dades, TCoordenada, com un registre amb els camps graus, minuts i segons que seran de tipus enter, i el camp lletra que serà un caràcter.

Declarar també un nou tipus de dades, TUbicació, com un registre amb els camps lloc que serà una cadena de caràcters de longitud 50 i els camps latitud i longitud que seran de tipus TCoordenada.

Fer la funció LatitudCorrecta que rebrà un paràmetre de tipus TCoordenada i retornarà un enter que serà 1 si la coordenada passada com a paràmetre és una latitud correcta () i

0 si no ho és. Una latitud és correcta si  $0 \leq \text{graus} \leq 180$ ,  $0 \leq \text{minuts} \leq 60$  i  $0 \leq \text{segons} \leq 60$  i la lletra és 'N' o bé 'S'.

Fer la funció LongitudCorrecta que rebrà un paràmetre de tipus TCoordenada i retornarà un enter que serà 1 si la coordenada passada com a paràmetre és una longitud correcta () i 0 si no ho és. Una latitud és correcta si  $0 \leq \text{graus} \leq 180$ ,  $0 \leq \text{minuts} \leq 60$  i  $0 \leq \text{segons} \leq 60$  i la lletra és 'E' o bé 'O'.

Per últim, fer un programa on es declari una variable del tipus TUbicació i es demani a l'usuari que introdueixi per teclat les dades dels camps de la variable. Després d'omplir la latitud, cal comprovar que la latitud introduïda sigui correcta i si no ho és, s'ha de mostrar per pantalla el missatge "Error: latitud incorrecta" i tornar a demanar la latitud. El procés s'ha de repetir mentre no s'introdueixi una latitud correcta. Després d'omplir la longitud, cal fer el mateix.

Una vegada les dades de la ubicació són correctes, mostrar per pantalla els valors del registre amb el format:

"XXXX – latitud: GG graus MM' SS''L, longitud: GG graus MM' SS''L"

on XXXX serà el nom de la ubicació, GG, MM, SS seran els valors de graus, minuts i segons de les coordenades i L serà la lletra de cada coordenada.

### **Problema 7**

Un polígon es pot definir a partir de les coordenades de cadascun dels costats que el componen. En aquest problema volem:

- Definir les estructures necessàries per guardar un polígon de  $n$  costats, amb un màxim de 30 costats:
  - Declarar un nou tipus de dades, TPunt, com un registre per a guardar les coordenades d'un punt. Aquest registre ha de tenir camps per a guardar les coordenades (x,y) d'un punt.
  - Declarar un nou tipus de dades, TPoligon, com un registre per a guardar la informació d'un polígon. Aquest registre ha de tenir com a camps el número de costats del polígon (enter) i un array de tipus TPunt per a guardar els vèrtexs del polígon.
- Fer el procediment LlegirPoligon que rep com a paràmetre un polígon i omple els seus camps: primer demana el número de vèrtexs del polígon i a continuació demana els valors de les coordenades x i y de tots els vèrtexs del polígon. Els vèrtexs s'han d'entrar ordenats.
- Fer la funció PerimetrePoligon que rep com a paràmetre un polígon i retorna el perímetre del polígon.
- Fer un programa que declari una variable de tipus TPoligon, cridi el procediment LlegirPoligon per a omplir la variable amb la informació d'un polígon i calculi el seu perímetre utilitzant la funció PerimetrePoligon. Com a resultat, s'ha de mostrar el missatge: "Perimetre del poligon: XX" on XX serà el valor del perímetre del polígon.

### Problema 8

Utilitzant el tipus de dades TEquip del problema 4 i alguna de les operacions fetes en aquell problema, volem crear una taula de classificació dels equips.

Per a fer-ho, cal crear els següents procediments:

- Procediment **InicialitzarClassificacio** que rep com a paràmetre un array de tipus TEquip i un enter amb la dimensió de l'array, i posa tots els camps de totes les posicions als següents valors: equip a "-----", ciutat a "-----", punts a 0 i pressupost a 0.
- Procediment **MostrarClassificacio** que rep com a paràmetre un array de tipus TEquip i un enter amb la dimensió de l'array, i mostra per pantalla la classificació que conté amb format:  
1.- <nom\_equip> <punts>  
2.- <nom\_equip> <punts>  
..... etc.
- Procediment **DesplacarClassificacio** que rep com a paràmetres un array de tipus TEquip amb la classificació, un enter amb la dimensió de l'array i un enter corresponent a un índex de l'array. El procediment ha de desplaçar una posició cap a la dreta tots els elements de la taula a partir de l'índex indicat.

A continuació, fer el programa principal que declari un array de tipus TEquip de 5 posicions i l'inicialitzi amb el procediment InicialitzarClassificacio. Declareu també una constant N\_EQUIPS amb valor igual a 5 i utilitzeu-la en tot el programa.

A continuació, fer un bucle que realitzi les següents opcions fins que l'usuari no vulgui inserir cap equip més:

- Mostrar l'estat actual de la classificació (amb el procediment MostrarClassificacio).
- Demanar a l'usuari la posició de la classificació on vol inserir un equip.
- Desplaçar un lloc tots els equips de la classificació que van des de la posició indicada per l'usuari fins el final per a fer lloc al nou equip (amb el procediment DesplacarClassificacio).
- Omplir la posició de la classificació que s'acaba de deixar lliure amb les dades del nou equip (es pot utilitzar el procediment OmplirDadesEquip del problema 4).
- Demanar a l'usuari si vol inserir algun equip més.

Per últim, després de sortir del bucle anterior, mostrar la classificació final.

Important.- Recordeu que les posicions de la classificació van de 1 a 5, mentre que els índexs de l'array on guardem la classificació van de 0 a 4.

### Problema 9

Com ja sabeu, el joc del dominó consta de 28 fitxes, cadascuna d'elles amb dos nombres entre 0 i 6. Cadascun dels 4 jugador té inicialment 7 fitxes.

- a) Feu les declaracions necessàries per poder guardar les 7 fitxes que té cada jugador en una partida de dominó.
- b) Feu un programa que, suposant que ja hem llegit quines fitxes té cada jugador, llegeixi dos valors que corresponen a les fitxes que hi ha als dos extrems del joc i ens digui si un jugador pot tirar fitxa o no.

### **Problema 10**

Fer un programa que defineixi les estructures necessàries per guardar els llibres d'una biblioteca. Supposeu que de cada llibre volem saber el títol, l'autor, si està deixat en préstec i, en aquest cas, la data d'inici del préstec. Inicialment el programa ha de demanar les dades dels llibres de la biblioteca i a continuació ha de permetre fer tres opcions:

- a) Buscar tots els llibres d'un determinat autor introduït per l'usuari.
- b) Realitzar el préstec d'un llibre: l'usuari introdueix el títol del llibre, i si no està deixat en préstec es marca com a prestat introduint la data actual.
- c) Marcar el retorn d'un llibre: l'usuari introdueix el títol del llibre i si estava prestat es marca com a no prestat.

### **Problema 11**

Una empresa vol fer una estadística de les vendes realitzades per cada producte en un any, per cadascun dels mesos de l'any. L'empresa té 10 productes en catàleg.

- a) Fer les declaracions que permetin guardar les vendes de l'empresa.
- b) Demanar les dades dels productes del catàleg i guardar-les a l'estructura de dades declarada.
- c) Fer un programa per:
  - Calcular el total de vendes de cada producte en tot l'any.
  - Calcular el total de vendes en cada mes.
  - Trobar el producte més venut en cadascun dels 12 mesos.
  - Trobar el producte i el mes que han tingut les vendes més altes, així com la quantitat venuda.

### **Problema 12**

Volem fer una aplicació per controlar la prova del salt de longitud en uns campionats d'atletisme. La prova consisteix en que cada atleta fa sis salts i s'escull el millor dels sis, de cara al resultat final. A la prova participen 10 atletes. Nosaltres haurem de guardar algunes dades personals de cada atleta (nom, cognoms, edat, alçada, país) i les marques que obté en cadascun dels sis salts. Després calcularem quin és l'atleta que ha fet el millor salt de tots i per tant, és el guanyador i escriurem les seves dades personals, la seva millor marca, i la mitja de tots els salts vàlids que ha fet.

Per fer-ho, hem de definir els tipus de dades necessaris que ens permetin guardar les dades i les marques de tots els atletes i després hem d'escriure un programa que primer llegeixi les dades personals de tots els atletes, i després vagi llegint les marques de tots els salts. Heu de tenir en compte que les dades dels salts s'han de llegir per ordre de salt, és a dir, primer haurem de llegir les marques del primer salt per tots els atletes, després les dades del segon salt de tots els atletes, i així successivament fins arribar al sisé salt de tots els atletes. Si algun atleta fa algun salt nul, la marca associada al salt ha de ser zero.

Un cop llegides les dades dels atletes i de les seves marques, s'ha de trobar quin és l'atleta guanyador, seleccionant el millor salt de tots i mirant a quin atleta correspon.

El programa ha d'acabar mostrant totes les dades personals de l'atleta guanyador, així com la seva millor marca i la mitja de tots els salts que ha fet, considerant només els salts vàlids.