




## EXERCICIS RESOLTS EN CLASSE DE TEORIA

 **(P1) (FitxerAutos.cpp)** Un fitxer anomenat “autos.txt” emmagatzema informació sobre components mecànics d'automòbils. La fitxa de cada component mecànic conté la informació següent:

- El nom de la peça (poden ser moltes paraules).
- Unitats disponibles (és un nombre enter).
- Preu de la peça (pot tenir decimals).

L'estructura del fitxer representa en la primera línia el nombre de peces que hi ha. En la línia següent, el nom de la primera peça i en una altra línia, les unitats disponibles i el preu. Aquestes dues últimes línies es repetiran per a cadascun dels components. Per exemple:

```
3
corretja de distribució ford mondeo 1.4
24 123.85
disc de fre audi a3
12 12.4
disc de fre opel corsa
30 3.70
```

Feu un programa que llija la informació del fitxer, la carregue en memòria i la mostre per pantalla. El programa serà capaç d'emmagatzemar un màxim de 500 peces. Açò és, el fitxer no tindrà mai més peces, les quals estaran escrites correctament segons el format anterior. Seguiu l'esquema de programa que us proporcionem a continuació:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

#define TAM 500 //nombre màxim de peces

struct peca
{
    string nom;
    int unitats;
    float preu;
};

/* Definició de prototips */
void LlegirFitxer (ifstream &f, peca vector[TAM], int & nom );
void MostrarMemòria (peca vector[TAM] , int nom);

/* Implementació de les funcions */
int main ( void )
{
    peca llista[TAM];
    int quants;
    string nom;
    ifstream fitx;

    cout << "Aquest programa llig el fitxer de peces mecàniques.\n" ;
    cout << "Dóna'm el nom del fitxer que conté la informació:\n" ;
    cin >> nom;
    fitx.open( nom.c_str());
```



```

if(!fitx)
    cout << "Error obrint el fitxer\n";
else
{
    LlegirFitxer( fitx, llista, quants );
    fitx.close();
    MostrarMemòria (llista , quants );
}
system("pause");
return 0;
}

```

- (P2) (FitxerAutosVersio2.cpp)** Feu una versió nova del programa anterior en què afegiu una funció que permeta recórrer tota la informació que hi ha a la memòria i la guarde de nou al fitxer (és a dir, sobreescriu el fitxer anterior). Tingueu en compte que haureu d'escriure el fitxer segons el format especificat anteriorment.

### EXERCICIS PER A RESOLDRE

- (FitxerAutosVersio3.cpp)** Afegiu al programa anterior una funció que permeti introduir les dades d'una peça nova. L'addició és farà en memòria tot i comprovant que hi ha prou espai per afegir peces noves. Si no n'hi ha, el programa principal mostrarà l'avís corresponent. Caldrà, a més a més, que aquesta funció cride una altra funció encarregada de sol·licitar les dades de la peça.

Així mateix, modifiqueu el programa principal perquè mostre un menú (repetidament, fins que es trie l'opció Eixir) amb les opcions següents:

- a- Afegir una peça nova.
- b- Desar el fitxer.
- c- Mostrar totes les peces.
- d- Eixir.

Tingueu en compte que, si el fitxer amb les dades no s'ha pogut carregar, el programa mostrarà l'avís corresponent, però mostrarà igualment el menú anterior.

- (P3) (FitxerMuntanya.cpp)** Feu un programa per a treballar amb informació sobre muntanyes. Aquestes dades es desaran a un fitxer que contindrà la informació següent:

Nom de la muntanya (pot ser més d'una paraula) i alçària d'aquesta (que s'expressarà en metres).

(Vegeu l'exemple del format d'aquest fitxer al final de l'enunciat.)

El programa permetrà fer, mitjançant un menú, les tasques següents:


- 1.- Veure tot el fitxer (nom de la muntanya i l'alçària d'aquesta).
- 2.- Afegir una muntanya.
- 0.- Eixir.

Cal definir una estructura de dades adient al problema i fer una descomposició modular del programa. Penseu que no sabem quanta informació sobre muntanyes hi haurà dins del fitxer, però podeu suposar que mai superarem les 1.000 muntanyes.




Exemple “muntanyes.txt”:

```
Mulhacén - Sierra Nevada - Granada
3482
Aneto - Pirineu central - Osca
3404
Moncayo - Serra del Moncayo - Saragossa
2313
Javalambre - Serra de Javalambre - Terol
2020
Massís de las Nieves - Las Palmas - Gran Canaria
1949
Teide - Tenerife - Tenerife
3718
Penyagolosa - Massís del Penyagolosa - Castelló
1814
Calderona - Serra de Javalambre - València
1837
Aitana - Serra d'Aitana - Alacant
1558
```

 **(Op1) (FitxerMuntanyaVersio2.cpp)** Amplieu el programa “FitxerMuntanya.cpp” perquè realitze mitjançant un menú, les tasques següents:


- 1.- Veure tot el fitxer (nom de la muntanya i alçària).
- 2.- Afegir-hi una muntanya.
- 3.- Mostrar la informació de totes les muntanyes que continguen una paraula introduïda per l'usuari.
- 4.- Desar el fitxer.
- 0.- Eixir.

**(Nota:** Les opcions noves són la 3 i la 4.)

 **(Op2) (FitxerMuntanyaVersio3.cpp)** Amplieu el programa “FitxerMuntanyaVersio2.cpp” perquè el menú permeta realitzar les tasques següents:

- 1.- Veure tot el fitxer.
- 2.- Afegir-hi una muntanya.
- 3.- Mostrar la informació de totes les muntanyes que continguen una paraula introduïda per l'usuari.
- 4.- Desar el fitxer.
- 5.- Veure el nom i l'alçària d'aquelles muntanyes que superen una altitud introduïda per l'usuari.
- 6.- Calcular-ne l'altitud mitjana.
- 0.- Eixir.

**(Nota:** Les opcions noves són la 5 i la 6. Recordeu-vos de fer una descomposició modular.)

 **(P4) (Matriu.cpp)** Amb un editor de textos qualsevol, creeu un fitxer anomenat “matriu1.dat” que continga la informació següent i amb el format següent:

- ✓ Primera línia: un enter que representa el nombre de files de la matriu.
- ✓ Segona línia: un enter que representa el nombre de columnes de la matriu.
- ✓ Línies següents: una fila de la matriu en cada línia, on cadascun dels elements està separat del següent per un espai.



A continuació, feu un programa que llija el contingut de la matriu des del fitxer i el mostre en la pantalla en forma de matriu:	Exemple de “matriu1.dat”:
4    -34	3
4    2	2
0    7	4 -34
	4 2
	0 7

**Nota:** La grandària màxima de les matrius serà 10 x 10.

**(P5) (MultiplicarMatriu.cpp)** Feu un programa que llija 2 fitxers “matriu1.dat” i “matriu2.dat” (2 matrius amb el format de l'exercici anterior), les multiplique (si són multiplicables), mostre el resultat per pantalla i el dese també en un altre fitxer amb el mateix format que els anteriors (“matriu3.dat”).

**(Punts.cpp)** Amb un editor de textos qualsevol, creeu un fitxer anomenat “punts.dat” que continga la informació següent i amb el format següent:

- ✓ Primera línia: un enter que representa el nombre de punts que conté el fitxer.
- ✓ Línies següents: en cada línia, les coordenades d'un punt.

Llavors, feu un programa que llija el contingut del fitxer de punts i els mostre amb el format següent:	Exemple de “punts.dat”:
(2,2,7)	3
(5,9,8)	2 2 7
(11,6,30)	5 9 8
	11 6 30

**Nota:** El nombre màxim de punts que emmagatzemarà el fitxer serà 100.

**(Reemplaçar.cpp)** Escriviu una funció que simule la funció “reemplaçar” que ofereixen els editors de text. La funció rebrà el nom del fitxer en què es troba el text, la cadena que es vol buscar i la cadena que s'emprarà per a fer la substitució. La funció tornarà, a més a més, el nombre de reemplaçaments que ha fet dins del fitxer.

Crideu a aquesta funció des del programa principal, que demanarà a l'usuari el nom del fitxer, la paraula que vol reemplaçar i la nova. En acabant, mostrarà per pantalla el nombre de reemplaçaments.

**(Notes.cpp)** Escriviu un programa que llija des del teclat la informació corresponent als alumnes de l'assignatura Informàtica i l'escriga en un fitxer de text. Aquesta informació estarà formada per l'identificador de l'alumne, el nom complet, la nota de pràctiques i la nota de l'examen.

**(Paraules.cpp)** Segons un estudi d'una universitat anglesa, no importa l'ordre en què les paraules estiguen escrites, només és important que la primera i l'última lletra estiguen escrites en les posicions correctes. La resta poden estar totalment intercanviades i encara es podrien llegir sense



problemes. Açò succeeix perquè no llegim cada lletra separatament, sinó que llegim les paraules com un tot.

Exemple:

*Sgeons un esudit d'una uinevritsat agnlesa, no ipmotra l'odrre en què les prauales esguetin esrcties... Pesornamelnt, em smelba inecrilbe...*

D'acord amb aquesta idea, feu un programa que llija un fitxer de text, modifique cada paraula segons allò indicat (açò és, barregi aleatòriament la posició de les lletres dins d'una paraula tot i mantenint la primera i la darrera lletres de la paraula) i cree un fitxer nou amb el resultat de la modificació. Feu una descomposició modular del programa.

**(Plàstics.cpp)** Una empresa de plàstics pensa que la competència l'espia perquè intercepta els fitxers d'informació que tramet per la xarxa. Per tant, ens demanen que desenvolupem un programa de xifrat per poder enviar els fitxers amb seguretat per la xarxa i desxifrar-los en rebre'ls. Així, abans d'enviar-lo, un fitxer de text s'haurà de xifrar amb una clau que solament coneguen els treballadors de l'empresa.

El sistema de xifrat que ens proposen consisteix a sumar el codi ASCII de cada caràcter al codi ASCII de la clau. El caràcter corresponent al codi ASCII de la suma és el caràcter xifrat. Com que volem que els codis xifrats generats siguin imprimibles, els caràcters xifrats han d'estar entre 32 i 127. Aquells caràcters que corresponguen a un codi ASCII superior a 127 (p. ex. ç,) simplement romandran invariables.

Exemple:

Frase original	La clau està en Rebeca.
Clau repetida	rebecarebecarebecarebe
Frase xifrada	lb"fpfvf"hwyá!gq\$wecgfe

**(Op3) (productes.cpp)** Suposem que tenim dos fitxers de text “productes1.dat” i “productes2.dat” que contenen un vector de registres de productes ordenats pel número de referència del producte. Cada registre té associada una estructura com la següent:

```
struct producte
{
    int referència;
    float quantitat;
};
```

La quantitat màxima de productes que hi haurà en el catàleg serà 1.000.

Escriviu un conjunt de funcions per a fer cadascuna de les tasques següents:

- Una funció que, a partir dels fitxers originals, escriu un fitxer nou que continga una llista única de productes que també estiga ordenada pel número de referència del producte.
- Una funció que escriu en un nou fitxer un llistat dels 10 productes del catàleg dels quals hi ha una quantitat major d'unitats.



- Una funció que escriba en un nou fitxer un llistat d'aquells elements que es troben per damunt de la mitjana d'existències del catàleg de productes.

Recordeu fer una descomposició modular del programa.