

EJERCICIO 3

Diseñe un programa que permita crear dos ficheros con información referente a los artículos de un almacén.

- Uno llamado "precios.dat", cuyos elementos son del tipo siguiente:

```
struct reg1 {  
    int codigo;  
    int cantidad;  
    int  precio;};
```

- Y otro llamado "pesos.dat", cuyos elementos son del tipo siguiente:

```
struct reg2{  
    int codigo;  
    int peso;  
    int cantidad;};
```

Los artículos en ambos ficheros estarán ordenados ascendentemente.

```
#include <conio.h>  
#include <iostream>  
#include <fstream>  
using namespace std;  
struct reg1 {  
    int codigo;  
    int cantidad;  
    int precio;  
};  
struct reg2 {  
    int codigo;  
    int cantidad;  
    int peso;  
};  
void creapesos ( );  
void creaprecios( );
```

```

int main() {
char opcion;
do {
    system("cls");
    cout << "Este programa crea los ficheros \n";
    cout <<"Los codigos ordenados  \n";
    cout <<" Elija opcion\n";
    cout <<" 1 - Crear fichero de pesos\n";
    cout <<" 2 - Crear fichero de precios\n";
    cout <<" 3 - Salir\n";
    opcion=getch();
    switch(opcion) {
        case '1': creapesos();      break;
        case '2': creaprecios();    break;
        case '3': break;
        default:cout<<"Opcion incorrecta\n"; break;
    }
    } while (opcion!='3');
return 0;
}

void creaprecios( ) {
    regl r1;
    int ultimo=-1;
    char seguir;
    ofstream f1("precios.dat", ios::binary);
    if (f1){
        cout <<"Se crea fichero de precios\n";
        do {
            do {
                cout <<"Deme codigo \n";
                cin >>r1.codigo;
            } while (r1.codigo<=ultimo);
            ultimo=r1.codigo;
        }
    }
}

```

```

        cout <<"Deme cantidad\n ";
        cin >> r1.cantidad;
        cout <<"Deme precio\n";
        cin >> r1.precio;
        f1.write((char*)&r1, sizeof(reg1));
        cout <<"Desea mas datos (s/n)?\n";
        seguir=getch();
        } while (seguir!='N');
    f1.close();
}
else
    cout <<"Error en la apertura del fichero\n";
}

void creapesos () {
    reg2 r2;
    int ultimo=-1;
    char seguir;

    ofstream f2("pesos.dat", ios::binary);
    if (f2) {
        cout <<"Se procede a crear fichero de pesos\n";
        do {
            do {
                cout <<"Deme codigo \n";
                cin>>r2.codigo;
            } while (r2.codigo<=ultimo);

            ultimo=r2.codigo;

            cout <<"Deme cantidad\n ";
            cin >>r2.cantidad;
            cout <<"Deme peso\n";
            cin >>r2.peso;

```

```

        f1.write((char*)&r2, sizeof(reg2));
        cout <<"¿Desea mas datos (s/n)?\n";
        seguir=getch();
    } while (seguir!='N');

    f2.close();
}
else
    cout <<"Error en la apertura del fichero\n";
}

```

EJERCICIO 4

Se tienen dos ficheros de acceso secuencial con información referente a los artículos de un almacén:

- Uno llamado "precios.dat", cuyos elementos son del tipo siguiente:

```

struct reg1 {
    int codigo;
    int cantidad;
    int  precio;};

```

- Y otro llamado "pesos.dat", cuyos elementos son del tipo siguiente:

```

struct reg2{
    int codigo;
    int cantidad;
    int peso; };

```

Los elementos de ambos ficheros se encuentran ordenados ascendentemente según el campo código. Puede que haya artículos que tengan información sólo en un fichero o en los dos.

Cree un algoritmo que fusione la información de ambos ficheros en otro fichero de acceso secuencial, que se llamará "mezcla.dat", cuyos elementos estarán ordenados ascendentemente según el campo código y serán del tipo:

```

struct reg3 {
    int codigo;
    int cantidad;
    int datos[2];
    /* posición 0 : precio
       posición 1 : peso  */
}

```

```
};
```

Si un artículo tiene información en ambos ficheros, en el campo cantidad se almacenará la suma de las dos cantidades que aparecen.

```
#include <iostream>
#include <fstream>
#include <conio.h>
using namespace std;
struct reg1{
    int codigo;
    int cantidad;
    int precio;
};
struct reg2{
    int codigo;
    int cantidad;
    int peso;
};
struct reg3{
    int codigo;
    int cantidad;
    int datos[2];
};

int main() {
    reg1 r1;
    reg2 r2;
    reg3 r3;
    ifstream f1("precios.dat", ios::binary);
    if (f1){
        ifstream f2("pesos.dat", ios::binary);
        if (f2) {
            ofstream f3("mezcla.dat", ios::binary);
            if (f3) {
```

```

f1.read((char*) &r1, sizeof(reg1));
f2.read((char*)&r2, sizeof(reg2));
while (!f1.eof() && !f2.eof()){
    if (r1.codigo < r2.codigo) {
        r3.codigo      = r1.codigo;
        r3.cantidad    = r1.cantidad;
        r3.datos[0]     = r1.precio;
        r3.datos[1]     = 0;
        f1.read((char*)&r1, sizeof(reg1));
    }
    else if (r1.codigo > r2.codigo) {
        r3.codigo      = r2.codigo;
        r3.cantidad    = r2.cantidad;
        r3.datos[1]     = r2.peso;
        r3.datos[0]     = 0;
        f2.read((char*)&r2, sizeof(reg2));
    }
    else {
        r3.codigo      = r1.codigo;
        r3.cantidad    = r1.cantidad+r2.cantidad;
        r3.datos[0]     = r1.precio;
        r3.datos[1]     = r2.peso;
        f1.read((char*)&r1, sizeof(reg1));
        f2.read((char*)&r2, sizeof(reg2));
    }
    f3.write((char*)&r3, sizeof(reg3));
}
while (!f1.eof()) {
    r3.codigo      = r1.codigo;
    r3.cantidad    = r1.cantidad;
    r3.datos[0]     = r1.precio;
    r3.datos[1]     = 0;
    f1.read((char*)&r1, sizeof(reg1));
    f3.write((char*)&r3, sizeof(reg3));
}

```

```
    }
while (!f2.eof()) {
    r3.codigo      = r2.codigo;
    r3.cantidad    = r2.cantidad;
    r3.datos[1]    = r2.peso;
    r3.datos[0]    = 0;
    f2.read((char*)&r2, sizeof(reg2));
    f3.write((char*)&r3, sizeof(reg3));
}
f1.close();
f2.close();
f3.close();
}
}

else cout <<"Error en la apertura de pesos";
}

else cout <<"Error en la apertura de precios";
getch();
return 0;
}
```