

Ejercicio tipo parcial 2

a) La FIFA tiene en el sistema de gestión del campeonato mundial los siguientes archivos entre otros:

Países.dat

Código de país (1 a 32)

Nombre de país, Grupo

Jugadores.dat

Código de jugador (char[5])

Nombre de jugador

Código de país, Edad

Goles.dat

Código de partido

Minutos

Tiempo (1: 1º tiempo; 2: 2º tiempo)

Código de jugador, Tipo de gol (1: penal; 2: en contra; 3: juego)

Partidos.dat

Código de partido (char [5])

Código de equipo 1

Código de equipo 2

Fecha, Hora, Código de sede

Sedes.dat

Código de sede

Nombre ciudad, Nombre estadio

Desarrollar un programa C++ que resuelva los siguientes puntos:

- 1) Generar un archivo con el código y el nombre del jugador, el nombre del país al que pertenece, y la cantidad de goles convertidos. En el archivo sólo deben aparecer los jugadores que hicieron goles, sin tener en cuenta los goles en contra.
- 2) Informar los datos del jugador que más cantidad de goles a favor convirtió.

b) Sobrecargar un operador de alguna de las clases del ejercicio anterior, que sea de utilidad para la resolución de los puntos.

```

//Ejercicio: tipo parcial 2
//Autor:DEK
//Fecha: 19-06- 2014
//Comentario:

#include <iostream>
#include <cstdlib>

using namespace std;

class Pais{
private:
    int codigoPais;
    char nombrePais[30];
public:
    int getCodigoPais(){return codigoPais;}
    const char *getNombrePais(){return nombrePais;}
    bool leerDeDisco(int);
};

bool Pais::leerDeDisco(int pos){
    FILE *p;
    p=fopen("paises.dat","rb");
    if(p==NULL)exit(1);
    fseek(p, sizeof *this*pos,0);
    bool leyo=fread(this, sizeof *this, 1, p);
    fclose(p);
    return leyo;
}

class Jugador{
private:
    char codigoJugador[5];
    char nombreJugador[30];
    int codigoPais;
    int edad;
public:
    const char* getCodigoJugador(){return codigoJugador;}
    const char* getNombreJugador(){return nombreJugador;}
    int getCodigoPais(){return codigoPais;}
    bool leerDeDisco(int);
};

bool Jugador::leerDeDisco(int pos){
    FILE *p;
    p=fopen("jugadores.dat","rb");
    if(p==NULL)exit(1);
    fseek(p, sizeof *this*pos,0);
    bool leyo=fread(this, sizeof *this, 1, p);

```

```

fclose(p);
return leyo;
}

```

```

class Gol{
private:
    int codigoPartido;
    int minutos, tiempo;
    char codigoJugador[5];
    int tipo;
public:
    const char *getCodigoJugador(){return codigoJugador;}
    int getTipo(){return tipo;}
    bool leerDeDisco(int);
};

```

```

bool Gol::leerDeDisco(int pos){
    FILE *p;
    p=fopen("goles.dat","rb");
    if(p==NULL)exit(1);
    fseek(p, sizeof *this*pos,0);
    bool leyo=fread(this, sizeof *this, 1, p);
    fclose(p);
    return leyo;
}

```

```

class GolesJugador{
private:
    char codigoJugador[5];
    char nombreJugador[30];
    char nombrePais[30];
    int cantidadGoles;
public:
    void setCodigoJugador(const char*codJ){
        strcpy(codigoJugador, codJ);}
    void setNombreJugador(const char *nom)
        {strcpy(nombreJugador,nom);}
    void setNombrePais(const char *np)
        {strcpy(nombrePais,np);}
    void setCantidadGoles(const int cant ){cantidadGoles=cant;}
    void grabarEnDisco();
    const char *getCodigoJugador(){return codigoJugador;}
    const char *getNombreJugador(){return nombreJugador;}
    const int getCantidadGoles(){return cantidadGoles;}
    bool leerDeDisco(int p);
    //sobrecarga para el punto b
    bool operator>(const GolesJugador &aux){
        if(this->cantidadGoles>aux.cantidadGoles) return true;
    }
};

```

```

        return false;
    }
    void Mostrar();
};

void GolesJugador::grabarEnDisco(){
    FILE *p;
    p=fopen("golesJuga.dat","ab");
    if(p==NULL)exit(1);
    fwrite(this, sizeof *this, 1, p);
    fclose(p);
}

bool GolesJugador::leerDeDisco(int pos){
    FILE *p;
    p=fopen("golesJuga.dat","rb");
    if(p==NULL)exit(1);
    fseek(p, sizeof *this*pos,0);
    bool leyo=fread(this, sizeof *this, 1, p);
    fclose(p);
    return leyo;
}

void GolesJugador::Mostrar(){
    cout<<"CODIGO DE JUGADOR: ";
    cout<<codigoJugador<<endl;
    cout<<"NOMBRE DE JUGADOR: ";
    cout<<nombreJugador<<endl;
    cout<<"PAIS: ";
    cout<<nombrePais<<endl;
    cout<<"CANTIDAD DE GOLES: ";
    cout<<cantidadGoles<<endl;
}

//////////PROTOTIPO FUNCIONES GLOBALES//////////
void punto1();
    int contarGoles(const char *);
    const char *buscarNombrePais(int);
void punto2();

//////////
int main(){
    punto1();
    punto2();
    system("pause");
    return 0;
}

```

```

void punto1(){
    Jugador obj;
    GolesJugador aux;
    int pos=0, cantGoles;
    while(obj.leerDeDisco(pos++)){
        cantGoles=contarGoles(obj.getCodigoJugador());
        if(cantGoles>0){
            aux.setCantidadGoles(cantGoles);
            aux.setCodigoJugador(obj.getCodigoJugador());
            aux.setNombreJugador(obj.getNombreJugador());
            aux.setNombrePais(buscarNombrePais(obj.getCodigoPais()));
            aux.grabarEnDisco();
        }
    }
}

```

```

int contarGoles(const char *codigoJugador){
    Gol obj;
    int pos=0, cant=0;
    while(obj.leerDeDisco(pos++)){
        if(strcmp(codigoJugador, obj.getCodigoJugador())==0)
            if(obj.getTipo()!=2)
                cant++;
    }
    return cant;
}

```

```

const char *buscarNombrePais(int cp){
    Pais obj;
    int pos=0;
    while(obj.leerDeDisco(pos++)){
        if(cp==obj.getCodigoPais())
            return obj.getNombrePais();
    }
}

```

// sin sobrecarga de operador >

```

void punto2(){
    GolesJugador obj, aux;
    int pos=0, maxGoles=0;
    while(obj.leerDeDisco(pos++)){
        if(obj.getCantidadGoles()>maxGoles){
            maxGoles=obj.getCantidadGoles();
            aux=obj;
        }
    }
}

```

```
    cout<<"GOLEADOR: ";
    aux.Mostrar();
}

//sobrecargando el operador > para la clase GolesJugador
void punto2(){
    GolesJugador obj, aux;
    aux.setCantidadGoles(0);
    int pos=0;
    while(obj.leerDeDisco(pos++)){
        if(obj>aux){
            aux=obj;
        }
    }
    cout<<"GOLEADOR: ";
    aux.Mostrar();
}
```