

Ejercicio 1

Escriba a través de un programa C++ un archivo binario llamado *grupo.dat*, formado por un conjunto de 200 pares de números generados aleatoriamente. Cada par de datos se conforma por un entero y un flotante.

```
//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Autor: Prof. Gerardo Sas.
//sitio web: http://e-fich.unl.edu.ar
//Guia de estudios Nº 6 - Ejercicio Nº 1

#include <iostream>
#include <cstdlib>
#include <fstream>
#include <iomanip>
using namespace std;

int main(int argc, char *argv[]) {
    //Creo un objeto de la clase fstream para manejar el archivo binario
    fstream archi("grupo.dat", ios::binary | ios::in | ios::out | ios::trunc);
    if(!archi.is_open()){cout<<"E R R O R "; return 0;};
    float f; int i;
    for (int c=1; c<=100; c++)
    { f=(float)(rand()/(1.0*rand())); // genera flotantes
      i=rand()% 1000; // genera enteros menores que 1000;
      cout<<c<<" "<<setw(10)<<f<<" "<<setw(10)<<i; //muestro por pantalla los numeros generados
      archi.write(reinterpret_cast<char*>(&f), sizeof(f)); //escribe en el archivo los flotantes
      archi.write(reinterpret_cast<char*>(&i), sizeof(i)); //escribe en el archivo los enteros
      cout<<" "<<tellp(): " "<<setw(10)<<archi.tellp()<<endl; //Tellp() devuelve la posicion en byte, que en
este caso es la ultima.
    }; // del for
    //calculo el largo (en pares de numeros)
    archi.seekp(0, ios::end);
    int c, largo= (archi.tellp()/(sizeof(f)+sizeof(i)));
    cout<<endl<<"Cantidad de pares: "<<largo<<endl;
    archi.seekg(0, ios::beg); //muevo el puntero de lectura al principio del archivo
    for(c=0; c<largo; c++){ //Comienza la lectura por pares
        archi.read(reinterpret_cast<char*>(&f), sizeof(f));
        archi.read(reinterpret_cast<char*>(&i), sizeof(i));
        cout<<c<<" "<<setw(10)<<f<<" "<<setw(10)<<i //Muestro los pares
        <<" "<<tellg(): " "<<setw(10)<<archi.tellg()<<endl;
    }
    archi.close();
}
//Programación Orientada a Objetos - Prof. Gerardo Sas
```

Ejercicio 2

Escriba un programa que abra el archivo generado en el ejercicio anterior y solicite al usuario que ingrese un entero, un flotante y una posición. El programa debe sobrescribir el par en la posición ingresada por el usuario por el nuevo par. Luego muestre la lista de datos en consola mostrando un par por línea.

```

//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Autor: Prof. Gerardo Sas.
//sitio web: http://e-fich.unl.edu.ar
//Guia de estudios Nº 6 - Ejercicio Nº 2
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <iomanip>
using namespace std;

int main(int argc, char *argv[]) {
    float f,auxf; int i,auxi, pos;
    cout<<"Ingrese un float : "; cin>> auxf;
    cout<<"Ingrese un Integer: "; cin>> auxi;
    cout<<"Ingrese una Posicion (0 a 99): "; cin>> pos;

    //Creo el objeto archi del tipo fstream
    fstream archi("grupo.dat", ios::binary|ios::in|ios::out|ios::ate);
    if(!archi.is_open()){cout<<"E R R O R "; return 0;};
    //archi.seekp(0,ios::end);//no es necesario porque utilice |ios::ate

    int c, largo= archi.tellp()/(sizeof(f)+sizeof(i));//calculo el largo
    cout<<"Largo en pares de numeros: "<<largo<<endl;

    // desplazamos el puntero desde el inicio (beg de begin)
    //metodo seekg
    archi.seekg(pos*(sizeof(f)+sizeof(i)),ios::beg);

    //leemos el par de datos
    archi.read(reinterpret_cast<char*>(&f), sizeof(f));
    archi.read(reinterpret_cast<char*>(&i), sizeof(i));
    //muestro por pantalla
    cout<<"Par de valores del "<<pos<<" lugar:"<<setw(10)<<f<<" "<<setw(10)<<i<<endl;

    // Proceso de reemplazo
    // metodo seekp
    // nos desplazamos hacia el par a reemplazar
    archi.seekp((pos*(sizeof(f)+sizeof(i))),ios::beg);

    //ahora estamos en la posicion
    archi.write(reinterpret_cast<char*>(&auxf), sizeof(auxf));//escribe en el archivo los flotantes
    archi.write(reinterpret_cast<char*>(&auxi), sizeof(auxi));//escribe en el archivo los enteros
    archi.seekg(0,ios::beg);//muestro desde el principio
    for(c=0; c<largo; c++){
        archi.read(reinterpret_cast<char*>(&f), sizeof(f));
        archi.read(reinterpret_cast<char*>(&i), sizeof(i));
        cout<<c<<": "<<setw(10)<<f<<" "<<setw(10)<<i<<endl;
    };
    //cerramos el archivo
    archi.close();
}

```

Ejercicio 3

Escriba un programa que genere un archivo binario con una lista de 100 enteros ordenados de forma creciente (por ejemplo: 0, 5, 10, 15, 20, 25...). Luego escriba otro programa que permita insertar (no reemplazar) un nuevo dato de tipo entero, ingresado por el usuario, en el archivo manteniendo el orden creciente. Evite utilizar vectores auxiliares.

```
//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Autor: Prof. Gerardo Sas.
//sitio web: http://e-fich.unl.edu.ar
//Guia de estudios Nº 6 - Ejercicio Nº 3
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <iomanip>
#include <ctime>
using namespace std;

int main(int argc, char *argv[]) {
    //Declaro el objeto archi
    fstream archi("crecer.dat" , ios::binary |ios::in|ios::out|ios::trunc);
    for (int c=5; c<=35; c+=5)
        //Escribo el archivo con 5,10,15,20,25,30,35
        archi.write(reinterpret_cast<char*>( &c), sizeof(c));
    //El archivo ya fue creado
    int i, pos, dato=0;
    archi.seekg(0,ios::end);//Me posciono al final del archivo
    int largo= (archi.tellg()/sizeof(i));//Calculo cantidad de elementos
    archi.seekg(0,ios::beg);//Llevo el puntero al inicio para comenzar la lectura
    cout<<"Lee archivo original: //5,10,15,20,25,30,35"<<endl;
    cout<<endl<<"Largo en cantidad de numeros: "<<largo<<endl;
    for (int c=0; c<largo; c++){
        archi.read(reinterpret_cast<char*>( &i), sizeof(i));
        cout<<"Posicion:"<<(i/5)-1<<" "<<i<<endl;}
    //*****
    srand( time(NULL) );//Para generar numeros distintos
    i=rand()% 50; // genera entero menor que 50;
    cout<<"Numero generado: "<<i<<endl;
    pos = -1;//Posicion de lectura, la primera es 0.-
    archi.seekg(0,ios::beg);//Llevo el puntero al inicio para comenzar la lectura
    //Mientras no se pase del tamaño y el dato leído sea menor al numero a insertar
    while((pos<largo-1)&&(dato < i)){
        archi.read(reinterpret_cast<char*>(&dato), sizeof(dato));
        pos++;//cuento que lei otro.-
        //cout<<"Tellg: "<<archi.tellg()<<endl;
    }
    cout<<i<<" -- Encontro la posicion siguiente: "<<pos<<endl;
    if (i <= dato){//Si NO esta al final del archivo escribe en el lugar que leyo el último
        archi.seekp((pos*sizeof(i)),ios::beg);
        archi.write(reinterpret_cast<char*>( &i), sizeof(i));//escribe en el archivo el entero ordenado
        pos++;//cuenta una posicion mas.
        while(pos<largo){//Si no llego al final del archivo
            archi.seekg((pos*sizeof(i)),ios::beg);//Posiciono para leer el siguiente
            archi.read(reinterpret_cast<char*>(&i), sizeof(i));//Leo
            archi.seekp((pos*sizeof(i)),ios::beg);//Retrocedo y escribo el anterior
```

```

        archi.write(reinterpret_cast<char*>( &dato), sizeof(dato)); //Escribe el siguiente ordenado
        dato= i; //Refresco dato con el que acabo de leer
        pos++; //cuento una posicion mas.-
    } //del while*/

} //del if.-
else
    dato=i; //En el caso que sea mas grande que el ultimo
//ahora me falta escribir el ultimo
archi.seekg(0, ios::end); //cuando ya lei todos
archi.seekp(0, ios::end); //voy al final y escribo el ultimo
archi.write(reinterpret_cast<char*>( &dato), sizeof(dato)); //escribe en el archivo el ultimo
//*****
int c; //Lee y muestra el archivo con el dato nuevo insertado
archi.seekg(0, ios::end);
largo= (archi.tellg()/sizeof(i)); //Vuelve a calcular el largo del archivo
cout<<endl<<"Largo en cantidad de numeros: "<<largo<<endl;
archi.seekg(0, ios::beg); //Muevo el puntero de lectura al principio del archivo
for(c=0; c<largo; c++){ //Leo y muestro los numeros del archivo
    archi.read(reinterpret_cast<char*>(&i), sizeof(i));
    cout<<"Posicion:"<<c<<" "<<i<<endl;
}
archi.close();
} //Programación Orientada a Objetos - Prof. Gerardo Sas

```

Ejercicio 4

Con un software de tratamiento de imagenes llamado *WellDesign* se han generado 78 archivos con los nombres "design1.wds", "design2.wds", ... "design78.wds". El software fue creado para *Windows XP*. Estos archivos presentan el problema que en el nuevo sistema operativo *Windows 10* al tratar de abrirlos con *WellDesign* presentan un molesto mensaje. Al consultar al soporte tecnico de la empresa que desarrollo *WellDesign* nos mencionan que si se modifica la cabecera de cada archivo se soluciona el problema. A partir del byte 178 se debe reemplazar "mswpx" por "msw10", y a partir del byte 239 reemplazar el valor entero 2003 por el valor 2016. Escriba un programa en C++ que realice esas correcciones en los 78 archivos almacenados.

```

//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Autor: Prof. Gerardo Sas.
//sitio web: http://e-fich.unl.edu.ar/moodle/
//Guia de estudios Nº 6 - Ejercicio Nº 4
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <string.h>
#include <cstring>
#include <sstream>
using namespace std;

void createWellDessing(string nombre,int tamanio){
    //Declaro el objeto
    fstream archi(nombre.c_str(), ios::binary |ios::in|ios::out|ios::trunc);
    if(!archi.is_open()){cout<<"E R R O R - al crear archivo: "<<nombre<<endl;

```

```

    }else{
    for (int c=0; c<=(tamano/sizeof(c)); c++){
        //Escribo el archivo con numeros al azar
        int num= rand()%256;
        archi.write(reinterpret_cast<char*>( &num), sizeof(num));
    }//El archivo ya fue creado
    archi.seekg(0,ios::end);//Me posciono al final del archivo
    int largo= (archi.tellg());//Calculo cantidad de elementos
    cout<<nombre<<" ; Largo en cantidad de bytes: "<<largo<<endl;
    archi.close();
    }
}

void arreglar(string nombre){
    fstream archi(nombre.c_str(), ios::binary |ios::in|ios::out);
    if(!archi.is_open()){cout<<"E R R O R - al crear archivo: "<<nombre<<endl;
    }else{
        cout<<"Apertura exitosa de "<<nombre<<endl;
        archi.seekp(0,ios::end);
        cout<<"Tamaño de archivo en bytes: "<<archi.tellp()<<endl;
        archi.seekp(178,ios::beg);
        char cadena[5];
        strcpy(cadena,"msw10");
        archi.write(cadena,sizeof(cadena));
        int anio= 2016;
        archi.seekp(239,ios::beg);
        archi.write(reinterpret_cast<char*>(&anio), sizeof(anio));
    }
}

int main(int argc, char *argv[]) {
    string nombre,cad;
    int contArchivos;
    std::stringstream aux;
    for(contArchivos=1; contArchivos <= 5; contArchivos++){
        nombre="";cad= "desing";aux.str("");
        aux<<cad<<contArchivos;
        cout<<aux.str()<<endl;
        nombre= aux.str();
        nombre+= ".wds";
        cout<<"Creando "<<nombre<<endl;
        createWellDessing(nombre, rand()%100+300);
    }

    for(contArchivos=1; contArchivos <= 5; contArchivos++){
        nombre="";cad= "desing";aux.str("");
        aux<<cad<<contArchivos;
        cout<<aux.str()<<endl;
        nombre= aux.str();
        nombre+= ".wds";
        cout<<"arreglando "<<nombre<<endl;
        arreglar(nombre);
        cout<<"arreglado!!! "<<nombre<<endl;
    }
    return 0;
}

```

Ejercicio 5

Escriba una clase para gestionar la tabla de las N mejores puntuaciones de un juego. La clase debe permitir ingresar un nuevo puntaje (y encargarse de ubicarlo en la posición de la tabla que corresponda, o descartarlo), consultar los datos, y almacenar y recuperar los mismos mediante un archivo binario.

```
using namespace std;

void crear(string nomb, int cant){
    /*Para este caso como no tengo el archivo de puntajes voy a generarlo
    1) Creo un vector con numeros al azar.
    2) Utilizo el algoritmo sort para ordenarlo de menor a mayor
    3) Utilizo el algoritmo reverse para invertir el orden
    4) Guardo en el archivo, la lista ordenada
    5) Ahora que tengo el archivo
    */
    string nombArchi;//Nombre del archivo fisico
    vector<double> V;//Vector que guarda el Puntaje
    nombArchi= nomb;
    fstream archi(nomb.c_str() , ios::binary |ios::in|ios::out|ios::trunc);
    if(!archi.is_open()){cout<<"E R R O R "; exit(0);};
    double f;
    for (int c=1; c<= cant; c++){
        f=(double)(rand()%10000)+100; // genera flotantes
        V.push_back(f);//Agrego al final del vector
    };// del for
    sort(V.begin(),V.end());//Ordeno el vector de meno a mayor
    reverse(V.begin(),V.end());//Invierto el orden (Mayor a Menor)
    for (int c=1; c<= cant; c++){
        archi.write(reinterpret_cast<char*>(&V[c]), sizeof(V[c]));//escribe en el archivo
    };// del for
}
//*****

class Puntaje{
    string nombArchi;//Nombre del archivo fisico
    vector<double> V;//Vector que guarda el Puntaje
public:
    Puntaje( string);//Constructor (nombre de archivo, cantidad de Puntajes)
    void agregar(double dato);//Agrega un puntaje en orden decreciente, pero no crece la lista
    void ver_puntajes();
};

Puntaje::Puntaje( string nomb){
    nombArchi= nomb;
    fstream archi(nomb.c_str() , ios::binary |ios::in|ios::out);
    if(!archi.is_open()){cout<<"E R R O R "; exit(0);};
    double f;
    V.clear();//Borro el vector
    //calculo el largo
    archi.seekp(0,ios::end);
```

```

int c, largo= (archi.tellp()/(sizeof(f)));
cout<<endl<<"Cantidad de puntajes: "<<largo<<endl;
archi.seekg(0,ios::beg);//muevo el puntero de lectura al principio del archivo
for(c=0; c<largo; c++){//Comienza la lectura
    archi.read(reinterpret_cast<char*>(&f), sizeof(f));
    V.push_back(f);//Agrego dato leído en el vector
    cout<<c<<" "<<setw(10)<<f<<" "<<setw(10) //Muestro
        <<"<<endl;
}
archi.close();
}

void Puntaje::agregar(double dato){
    V.push_back(dato);//Agrego el dato al final
    sort(V.begin(),V.end());//Ordeno el vector
    reverse(V.begin(),V.end());//Doy vuelta de orden
    V.pop_back();//Elimino el ultimo
    //Creo un objeto de la clase fstream para manejar el archivo binario
    fstream archi(nombArchi.c_str() , ios::binary |ios::in|ios::out|ios::trunc);
    if(!archi.is_open()){cout<<"E R R O R "; exit(0);};
    for (int c=1; c<= (int)V.size(); c++){
        archi.write(reinterpret_cast<char*>(&V[c]), sizeof(V[c]));//escribe en el archivo
    };// del for
}

void Puntaje::ver_puntajes(){
    for(int c=0; c< (int)V.size(); c++){//Comienza la lectura
        cout<<c<<" "<<setw(10)<<V[c]<<endl;
    }
}

int main(int argc, char *argv[]) {
    //Creo un objeto de la clase Puntaje
    crear("Puntaje.bin", 10);
    Puntaje P("Puntaje.bin");
    double d;
    cout<<"Ingrese puntaje a agregar: ";cin>> d;
    P.agregar(d);
    P.ver_puntajes();
}

```

//Programación Orientada a Objetos - Prof. Gerardo Sas

Cuestionario

- ✦ ¿Qué entendemos por archivo?
- ✦ ¿Cuál es la importancia de utilizar archivos?

- ✦ ¿Cuál es la diferencia entre archivos de texto y archivos binarios?
- ✦ ¿Puede abrirse un archivo de texto como binario? ¿Y un archivo binario como archivo de texto?
- ✦ ¿Puede guardarse un objeto de tipo `std::string` en un archivo de texto? ¿Y en un archivo binario? Justifique sus respuestas.

Ejercicios Adicionales

Ejercicio 1

Escriba una función *CreateDummyFile(...)* que reciba el nombre de un archivo y un entero con su tamaño en bytes. La función debe crear un archivo binario del tamaño especificado y llenarlo con valores aleatorios (basura).

Escriba una sobrecarga de la función en la que se especifique el tamaño del archivo mediante un *std::string*, por ejemplo: "100Mb", "10Kb" ó "893b".

```
//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Autor: Prof. Gerardo Sas.
//sitio web: http://e-fich.unl.edu.ar
//Guia de estudios Nº 6 - Ejercicio Adicional Nº 1
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <iomanip>
#include <ctime>
#include <sstream>
using namespace std;

void createDummyFile(string nombre,int tamanio){
    //Declaro el objeto
    fstream archi(nombre.c_str(), ios::binary |ios::in|ios::out|ios::trunc);
    for (int c=0; c<=int(tamanio/sizeof(c)); c++){
        //Escribo el archivo con numeros al azar
        int num= rand()%256;
        archi.write(reinterpret_cast<char*>( &num), sizeof(num));
    }//El archivo ya fue creado
    int i;
    archi.seekg(0,ios::end);//Me posciono al final del archivo
    int largo= (archi.tellg()/sizeof(i));//Calculo cantidad de elementos
    archi.seekg(0,ios::beg);//llevo el puntero al inicio para comenzar la lectura
    cout<<"Lee archivo original: "<<endl;
    cout<<endl<<"Largo en cantidad de numeros: "<<largo<<endl;
    for (int c=0; c<largo; c++){
        archi.read(reinterpret_cast<char*>( &i), sizeof(i));
        //cout<<setw(8)<<i<<" ";
    }
    cout<<"El archivo ya fue creado"<<endl;
    archi.close();
}
//*****
void createDummyFile(string nombre,string tam){
    //Busco Mb
    cout<<"Entro con: "<<nombre<<" - "<<tam<<endl;
    int total;
    unsigned int loc ;
    std::stringstream aux;
    loc = tam.find("Mb",0);
```

```

        if( loc != string::npos ){
            string subcad = tam.substr(0,loc);
            aux<<subcad.c_str();
            aux>>total; total*= 1000000;
            cout<<"Mb - tamaño en Bytes: "<<total<<endl;
        }else{
            loc = tam.find("Kb",0);
            if( loc != string::npos ){
                string subcad = tam.substr(0,loc);
                aux<<subcad.c_str();
                aux>>total; total*= 1000;
                cout<<"Kb - tamaño en Bytes: "<<total<<endl;}
            else{
                loc = tam.find("b",0);
                if( loc != string::npos ){
                    string subcad = tam.substr(0,loc);
                    aux<<subcad.c_str();
                    aux>>total; cout<<"b - tamaño en Bytes: "<<total<<endl;
                }
                else
                    cout<<"No encuentro unidades"<<endl;
            }
        }
        createDummyFile(nombre, total);
    }

int main(int argc, char *argv[]) {
    string nombre;
    string bytess;
    char cad[20];
    srand( time(NULL) );//Para generar numeros distintos
    cout<<"Ingrese el nombre del archivo: //"; cin>>nombre;cin.ignore();
    cout<<"Ingrese la cantidad de Bytes (Mb - Kb - b): "; cin.getline(cad,20);
    bytess= cad;
    createDummyFile(nombre, bytess);
}

```

Ejercicio 2

Proponga una clase *Persona* para representar a una persona a partir de su nombre, apellido y fecha de nacimiento con día, mes y año. Diseñe una clase llamada *RecursosHumanos* que permita manejar los datos de un grupo de personas. La clase debe poseer métodos para:

- Guardar los datos de las personas en un archivo binario.
- Cargar los datos de las personas desde un archivo binario.
- Agregar una persona.
- Eliminar una persona a partir de su nombre o apellido.
- Indicar la cantidad de personas en la base de datos.

//sitio web: <http://e-fich.unl.edu.ar>

//Guia de estudios Nº 6 - Ejercicio Adicional Nº 2

/*clase Persona para representar a una persona a partir de su nombre, apellido y fecha de nacimiento con día, mes y año. Diseñe una clase llamada RecursosHumanos que permita manejar los datos de un grupo de personas. La clase debe poseer métodos para:

- a) Guardar los datos de las personas en un archivo binario.
- b) Cargar los datos de las personas desde un archivo binario.
- c) Agregar una persona.
- d) Eliminar una persona a partir de su nombre o apellido.
- e) Indicar la cantidad de personas en la base de datos.

*/

```
#include <iostream>
```

```
#include <cstring>
```

```
#include <fstream>
```

```
#include <cstdlib>
```

```
using namespace std;
```

```
typedef struct{  
    char apeynomb[50];  
    int diaNac, mesNac, anioNac;  
} persona;
```

```
class recursosHumanos{  
    char nombreArchi[50]; //Aqui se guardará el nombre físico del archivo  
    persona p[100]; //Este arreglo cargará todos los registros del disco a memoria  
    int cantidad; //Contador de registros
```

```
public:
```

```
    recursosHumanos(string nArch){strcpy(nombreArchi,nArch.c_str());} //Constructor  
    void carga(); //Metodo que permitira la carga de los registros al vecctor  
    persona buscar(string cad); //Buscara el nombre (cad) en el arreglo  
    void guardar(); //Permitira guardar el arreglo en disco;  
    bool eliminar(string cad); //Elimina un nombre del arreglo y guarda en disco el vector modificado  
    void agregar(persona per); //Agrega un registro en el arreglo y en el archivo en disco.  
    int ver_cantidad(){return cantidad;} //Devuelve la cantidad de registro o posiciones del arreglo  
    char * ver_apeynomb(int n){return p[n].apeynomb;} //Devuelve el nombre de la persona en la posicion n  
    int ver_diaNac(int n){return p[n].diaNac;} //Devuelve el dia de nacimiento en la posicion n.  
    int ver_mesNac(int n){return p[n].mesNac;} //Devuelve el mes de nacimiento en la posicion n.  
    int ver_anioNac(int n){return p[n].anioNac;} //Devuelve el anio de nacimiento en la posicion n.  
};
```

```
void recursosHumanos::carga(){  
    fstream archi_in(nombreArchi,ios::binary|ios::in); //Declaro el objeto archi_in para leer datos del archivo  
    if(!archi_in.is_open()){ //Si no se abrio informo del error  
        cout<<"Error en apertura de archivo"<<endl;  
        //Intento crearlo por si el error se genero porque no existia  
        //cout<<"Se creara un archivo vacio"<<endl;  
        getchar();  
        fstream archi(nombreArchi,ios::binary|ios::in|ios::out|ios::trunc);  
        archi.clear(); //Limpio  
        archi.close(); //Cierro el archivo  
    }else{  
        cantidad=0; //Pongo el contador de registros en 0  
        archi_in.seekg(0,ios::end); //Voy al final del archivo  
        int total = archi_in.tellg()/sizeof(persona); //Calculo el tamaño  
        archi_in.seekg(0,ios::beg); //Vuelvo al principio  
        while(cantidad < total){ //leo todos los registros y los cargo en el arreglo
```

```

        archi_in.read(reinterpret_cast<char*>(&p[cantidad]), sizeof(persona));
        //cout<<cantidad<<"- "<<p[cantidad].apeynomb<<endl;//Voy mostrando lo que leo
        cantidad++;//Incremento en 1 la cantidad
    }
    archi_in.close();//Cierro el archivo
}
}
persona recursosHumanos::buscar(string cad){
    persona aux;
    int c=0;
    while(c<cantidad){//Mientras c < cantidad no llega al final del arreglo
        if(strcmp(p[c].apeynomb,cad.c_str())==0){//Compara nombres y si lo encuentra
            aux= p[c];//Devuelvo el struct completo
            return aux;
        }
        c++;//Incrementa el contador de registros
    }//Si pasa es porque no lo encontro.
    strcpy(aux.apeynomb,"No Encontrado");//Asigno el mensaje y devuelvo el struct
    return aux;
}

void recursosHumanos::guardar(){
    //Abro el archivo para sobreescribirlo con el arreglo de struct
    fstream archi_out(nombreArchi,ios::binary|ios::out|ios::trunc);
    if(!archi_out.is_open()){
        cout<<"Error en apertura de archivo"<<endl;
        //getchar();
    }else{
        int c=0;
        archi_out.seekg(0,ios::beg);//Vuelvo al principio
        while(c< cantidad){//escribo todos los registros del arreglo
            archi_out.write(reinterpret_cast<char*>(&p[c]), sizeof(persona));
            //cout<<c<<"- "<<p[c].apeynomb<<endl;
            c++;//incremento el contador de registros grabados
        }
        archi_out.close();//Cierro el archivo
    }
}

bool recursosHumanos::eliminar(string cad){
    int c=0;//contador en cero
    while(c<cantidad){//Mientras no llegue al último
        if(strcmp(p[c].apeynomb,cad.c_str())==0){
            break;//Si lo encuentra sale del while
        }
        c++;//Incremento el contador
    }//fin del while
    if (c<cantidad){//Si el contador no llego al final, lo encontro.
        for(int k= cantidad-1; k > c;k--){
            p[k-1]=p[k];//Corro todos los registros 1 lugar para arriba
        }
        cantidad--;//Decremento cantidad en 1
        guardar();//Guardo en disco el arreglo modificado
    }
    return 0;
}
}

```

```

void recursosHumanos::agregar(persona per){
    fstream archi_out(nombreArchi,ios::binary|ios::out|ios::ate);
    if(!archi_out.is_open()){
        cout<<"Error en apertura de archivo"<<endl;
        //getchar();
    }else{
        //archi_out.seekp(0,ios::end);//al final
        archi_out.write(reinterpret_cast<char*>(&per), sizeof(per));
        p[cantidad]= per;
        cantidad++;
    }
    archi_out.close();
}

int menu(){
    int op;
    cout<<"Recursos Humanos"<<endl;
    cout<<"1) Salir"<<endl;
    cout<<"2) Ver archivo"<<endl;
    cout<<"3) Buscar Persona"<<endl;
    cout<<"4) Guardar datos en Archivo"<<endl;
    cout<<"5) Eliminar Persona"<<endl;
    cout<<"6) Ver cantidad de personas en el archivo"<<endl;
    cout<<"7) Agregar Persona"<<endl;
    cout<<endl<<"Elija una Opcion: ";cin>>op;
    return op;
}

int main(int argc, char *argv[]) {
    recursosHumanos A("archivo.bin");
    A.carga();
    int opcion=0;
    while(opcion != 1){
        opcion = menu();
        switch (opcion){
            case 1: {A.guardar(); return 0;};
                break;
            case 2:{int c=0;
                for(c=0;c < A.ver_cantidad(); c++){
                    cout<<"Nombre : "<<A.ver_apeynomb(c)<<" , Fecha Nac: "
                        <<A.ver_diaNac(c)<<"/"<<A.ver_mesNac(c)<<"/"<<A.ver_anioNac(c)<<endl;
                }
            };break;
            case 3:{persona dato,encontro;
                cout<<"Ingrese el nombre y apellido a buscar: "; cin>>dato.apeynomb;
                encontro= A.buscar(dato.apeynomb);
                if(strcmp(encontro.apeynomb, "No Encontrado") == 0){
                    cout<<"Nombre no encontrado."<<endl;
                }else{
                    cout<<"Nombre: "<<encontro.apeynomb<<endl;
                    cout<<"Fecha de Nacimiento: "<<encontro.diaNac<<"/"<<encontro.mesNac
                        <<"/"<<encontro.anioNac<<endl;
                };break;
            }
        }
    }
    //switch
    case 4:{A.guardar();};break;
}

```

```

case 5:{persona dato,encontro;
    cout<<"Ingrese el nombre y apellido a Eliminar: ";
    cin.ignore();gets(dato.apeynomb);
    encontro= A.buscar(dato.apeynomb);
    if(strcmp(encontro.apeynomb,"No Encontrado")== 0){
        cout<<"Nombre no encontrado."<<endl;
    }else{
        cout<<"Nombre: "<<encontro.apeynomb<<endl;
        cout<<"Fecha de Nacimiento: "<<encontro.diaNac<<"/"<<encontro.mesNac
            <<"/"<<encontro.anioNac<<endl;
        A.eliminar(encontro.apeynomb);
        cout<<"ELIMINADO"<<endl;
    };break;
case 6:{cout<<"Cantidad de personas en el archivo: "<<A.ver_cantidad()<<endl;
    };break;
case 7:{ persona dato;
        cin.ignore();
        cout<<"Ingrese Apellido y Nombre: "; cin.getline(dato.apeynomb,50);
        cout<<"Ingrese dia de Nacimiento: "; cin>>dato.diaNac;
        cout<<"Ingrese mes de Nacimiento: "; cin>>dato.mesNac;
        cout<<"Ingrese Anio de Nacimiento: "; cin>>dato.anioNac;
        A.agregar(dato);};break;

    }//del switch
} //del while
return 0;
}

```