



Nombre del alumno:

NOTA: 9 (nueve)

**Aclaraciones:**  
**Leer detenidamente los enunciados.**  
**Pegar la resolución al final del enunciado de cada punto.**

1) Una empresa que realiza instalaciones sanitarias tiene, entre otros, los siguientes archivos en su sistema:

**Tecnicos.dat:** DNI (int), nombre y apellido (char[50]), puesto (1 a 10), sueldo, fecha de ingreso.

**Tecnicos\_obras.dat:** Número de obra, DNI del técnico (int), fecha de inicio(día, mes y año), tipo (1 a 5).

El archivo tiene registros desde el año 2020 a la fecha; puede haber muchos registros para cada obra, y un técnico puede participar de muchas obras. Se pide:

a) Generar un archivo nuevo con el siguiente formato de registro:

DNI, nombre y apellido de los técnicos que hayan participado de alguna obra este año. (3 puntos).

b) El mes del 2022 en el que se registraron más ingresos de técnicos (3 puntos).

/\*

NOMBRE:

LEGAJO:

\*/

```
#include <iostream>
```

```
#include <cstdlib>
```

```
#include <cstring>
```

```
using namespace std;
```

```
class Fecha{
```

```
private:
```

```
    int dia,mes, anio;
```

```
public:
```

```
    void Mostrar(){
```

```
        cout<<dia<<"/"<<mes<<"/"<<anio<<endl;
```

```
    }
```

```
    int getDia(){return dia;}
```

```
    int getMes(){return mes;}
```

```
    int getAnio(){return anio;}
```

```
    void setDia(int d){dia=d;}
```

```
    void setMes(int m){mes=m;}
```

```
    void setAnio(int a){anio=a;}
```

```
};
```

```
class Tecnico{
```

```
private:
```

```
    char nombre[50];
```

```
    char apellido[50];
```

```
    int puesto;
```

```
    int DNI;
```

```
    float sueldo;
```

```
    Fecha ingreso;
```

```
    bool activo;

public:

    void setDNI(int i);

    void setNombre(const char*n);

    void setApellido(const char*n);

    void setPuesto(int i);

    void setSueldo(float i);

    void setFecha(Fecha i);

    void setActivo(bool i);


    int getDNI();

    const char* getNombre();

    const char* getApellido();

    int getPuesto();

    float getSueldo();

    Fecha getFecha();

    void getActivo();

};
```

```
class ArchivoTecnico{

private:

    char nombre[40];

public:

    ArchivoTecnico(const char*n);

    int contarRegistros();

    Tecnico leerRegistro(int i);

};
```

```
class Tecnico_Obras{  
  
private:  
  
    int numero;  
  
    int DNI;  
  
    Fecha inicio;  
  
    int tipo;  
  
    bool activo;  
  
public:  
  
    void setDNI(int i);  
  
    void setNumero(int i);  
  
    void setFecha(Fecha i);  
  
    void setTipo(int i);  
  
    void setActivo(bool i);  
  
  
    int getDNI();  
  
    int getNumero();  
  
    Fecha getFecha();  
  
    int getTipo();  
  
    void getActivo();  
  
  
};
```

```
class ArchivoTecnicoObras{  
  
private:  
  
    char nombre[40];
```

public:

ArchivoTecnicoObras(const char\*n);

int contarRegistros();

Tecnico\_Obras leerRegistro(int i);

};

class FINAL{

private:

int DNI;

char nombre[50];

char apellido[50];

bool estado;

public:

//SETS

void setDNI(int i){ DNI = i; }

void setNombre(const char\* n){ strcpy(nombre,n); }

void setApellido(const char\* n){ strcpy(apellido,n); }

void setEstado(bool n){estado = n;}

//GETS

int getDNI(){ return DNI; }

const char\* getNombre(){ return nombre; }

const char\* getApellido(){ return apellido; }

bool getEstado(){return estado;}

```

//OTROS

void Mostrar(){

    if( estado ){

        cout << nombre << endl;

        cout << apellido << endl;

        cout << DNI << endl;

    }

}

void operator=(Tecnico &obj){

    DNI = obj.getDNI();

}

};

```

```

class ArchivoFINAL{

private:

    char nombre[30];

public:

    ArchivoFINAL(const char* n){

        strcpy(nombre,n);

    }

    FINAL leerRegistro(int pos){

        FINAL reg;

        FILE *p=fopen(nombre, "rb");

        if(p==NULL) return reg;
    }
}

```

```
fseek(p, sizeof reg*pos,0);  
  
fread(&reg, sizeof reg,1, p);  
  
fclose(p);  
  
return reg;  
}
```

```
int contarRegistros(){  
  
    FILE *p=fopen(nombre, "rb");  
  
    if(p==NULL) return -1;  
  
    fseek(p, 0,2);  
  
    int tam=ftell(p);  
  
    fclose(p);  
  
    return tam/sizeof(FINAL);  
}
```

```
bool Escribir(FINAL reg){  
  
    FILE *p=fopen(nombre, "ab");  
  
    if(p==NULL) return false;  
  
    bool escribio = fwrite(&reg, sizeof reg, 1, p);  
  
    fclose(p);  
  
    return escribio;  
}
```

```
};
```

```
void punto1();
```

```
void punto2();
```

```
int main(){
```

```
    punto1();
```

```
    system("pause");
```

```
    system("cls");
```

```
    punto2();
```

```
    return 0;
```

```
}
```

```
void punto1(){
```

```
/*
```

a) Generar un archivo nuevo con el siguiente formato de registro:

DNI, nombre y apellido de los técnicos que hayan participado de alguna obra este año

```
*/
```

```
/**TECNICO
```

```
    nombre: pepe
```

```
    apellido: espinoza
```

```
    puesto: 3
```

```
    dni: 12345678
```

```
    sueldo: 1000
```

```
    ingreso: 1 1 2023
```

```
*/
```

```
/** TECNICO OBRAS
```

```
    numero: 1
```



dni: 12345678

inicio: 2023

tipo: 4

\*/

//CREAR ARCHIVO

ArchivoTecnicoObras archTecObras("Tecnicos\_obras.dat");

ArchivoTecnico archTec("Tecnicos.dat");

//GUARDA CANTIDAD

int cantObras = archTecObras.contarRegistros();

int cantTec = archTec.contarRegistros();

//RECORRE TECNICOS

for( int i = 0 ; i < cantTec ; i++ ){

    Tecnico objTec = archTec.leerRegistro(i);

//RECORRE OBRAS

for( int j = 0 ; j < cantObras ; j++ ){

    Tecnico\_Obras objObra = archTecObras.leerRegistro(j);

//VERIFICA ANIO Y DNI

if(( objObra.getFecha().getAnio() == 2023 )&&( objObra.getDNI() == objTec.getDNI() )){

    //CREA OBJETO

    FINAL objFINAL;

    objFINAL.setApellido(objTec.getApellido());

    objFINAL.setDNI( objTec.getDNI());

```
objFINAL.setNombre(objTec.getNombre());
```

```
objFINAL.setEstado(true);
```

```
//CREA ARCHIVO Y GUARDA
```

```
ArchivoFINAL archFinal("Final.dat");
```

```
archFinal.Escribir(objFINAL);
```

```
//SALE DEL FOR OBRAS
```

```
j = cantObras;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
void punto2(){
```

```
/*
```

b) El mes del 2022 en el que se registraron más ingresos de técnicos (3 puntos).

```
*/
```

```
/**TECNICO
```

```
    nombre: pepe
```

```
    apellido: espinoza
```

```
    dni: 12345678
```

```
    puesto: 3
```

```
    sueldo: 1000
```

```
    ingreso: 1 1 2023
```

```
*/
```

```
//CREA ARCHIVO
```

```
ArchivoTecnico archTec("Tecnicos.dat");
```

```

//GUARDA CANTIDADES

int cantTec = archTec.contarRegistros();

int mes[12] = {0};


//RECORRE TECNICOS

for(int i = 0 ; i < cantTec; i++){

    Tecnico objTec = archTec.leerRegistro(i);


    //VERIFICA EL AÑO 2022

    if( objTec.getFecha().getAnio() == 2022){

        //AUMENTA EL CONTADOR MES

        mes[ objTec.getFecha().getMes()-1 ]++;

    }

}

//CREA MAYOR

int mayor = 0;


//RECORRE VECTOR MESES

for(int i = 0 ; i < 12 ;i++){


    //VERIFICA CANTIDAD DE MAYORES

    if (mes[i]>mes[mayor]) mayor = i;

}

```

```
cout << "El mes con mayor ingresos es " << mayor+1 << endl;
```

```
}
```

Desarrollar todas las clases que se consideren necesarias para la resolución del ejercicio. **No es obligatorio codificar de manera completa los métodos; se considera suficiente que se coloquen los prototipos dentro de las clases.**

2) Agregar una sobrecarga para algún operador de alguna de las clases del ejercicio 1, que reciba como parámetro un objeto de una clase distinta.

```
void operator=(Tecnico &obj){
```

```
    DNI = obj.getDNI();
```

```
}
```

```
void FINAL::operator= (Tecnico &obj){
```

```
    DNI = obj.getDNI();
```

```
}
```

3) ¿Es posible que una clase tenga varios constructores?. Justifique la respuesta

Una clase puede tener más de un constructor, debido a que en los parámetros del constructor se puede mandar varios parámetro, o ninguno, o uno solo. Dependiendo del caso en el que se quiera usar.