

```
#include <iostream>
#include <stdio>
#include <stdlib>
#include <cstring>

using namespace std;

class Fecha{
private:
    int dia, mes, anio;
public:
    int getDia(){return dia;}
    int getMes(){return mes;}
    int getAnio(){return anio;}
};

class Vendedor{
private:
    char codigoVendedor[5];
    char nombre[30];
    int provincia,calificacion;
public:
    const char *getCodigoVendedor(){return codigoVendedor;}
    const char *getNombreVendedor(){return nombre;}
    int getProvincia(){return provincia;}
    bool leerDeDisco(int);
};

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

class Usuario{
private:
    char codigoUsuario[5];
    char nombre[30];
    int DNI;
    Fecha registro;
public:
    bool leerDeDisco(int);
    Fecha getFechaRegistro(){return registro;}
};

bool Usuario::leerDeDisco(int pos){
    FILE *p;
    p=fopen("usuarios.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 Venta{
private:
    int numeroCompra;
    char codigoVendedor[5];
    char codigoUsuario[5];
    float importe;
    Fecha fechaVenta;
    int tipoPago;
public:
    const char *getCodigoVendedor(){return codigoVendedor;}
    Fecha getFechaVenta(){return fechaVenta;}
    bool leerDeDisco(int);
};

bool Venta::leerDeDisco(int pos){
    FILE *p;
    p=fopen("ventas.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 Inactivo{
private:
    char codigoVendedor[5];
    char nombre[30];
public:
    void setCodigoVendedor(const char *cv){strcpy(codigoVendedor,cv);}
    void setNombreVendedor(const char *n){strcpy(nombre,n);}
    bool grabarEnDisco();
};

bool Inactivo::grabarEnDisco(){
    FILE *p;
    p=fopen("inactivos.dat", "ab");
    if(p==NULL)exit(1);
    bool grabo=fread(this, sizeof *this,1,p);
    fclose(p);
    return grabo;
}

void puntoA();
void puntoB();
void puntoC();

int main() {
    puntoA();//ver como evitar duplicaci⬢n registros
    puntoB();
    puntoC();
    return 0;
}

bool buscarVentas(const char *cv, int anio){
    Venta reg;
    int pos=0;
    while(reg.leerDeDisco(pos++){
        if(reg.getFechaVenta().getAnio()==anio
            if(strcmp(reg.getCodigoVendedor(),cv)==0)
                return true;
        }
    }
    return false;
}

void puntoA(){
    Inactivo aux;
    Vendedor reg;
    int pos=0;
    /**while(reg.leerDeDisco(pos++){
        if(buscarVentas(reg.getCodigoVendedor(),2020)==false){
            aux.setCodigoVendedor(reg.getCodigoVendedor());
            aux.setNombreVendedor(reg.getNombreVendedor());
            aux.grabarEnDisco();
        }
    }*/
    FILE *p;
    p=fopen("ventas.dat", "rb");
    if(p==NULL)exit(1);
    while(fread(reg, sizeof reg,1,p)==1){
        if(buscarVentas(reg.getCodigoVendedor(),2020)==false){
            aux.setCodigoVendedor(reg.getCodigoVendedor());
            aux.setNombreVendedor(reg.getNombreVendedor());
            aux.grabarEnDisco();
        }
    }
    fclose(p);
}

void mostrarVector(int *v){
}

void puntoB(){
    Usuario reg;
    int pos=0;
    int v2015[12]={0};
    while(reg.leerDeDisco(pos++){
        if(reg.getFechaRegistro().getAnio()==2015)
            v2015[reg.getFechaRegistro().getMes()-1]++;

        /**if(reg.getAnio()==2015) si se decide usar 3 propiedades para la fecha
            v2015[reg.getMes()-1]++;*/
    }
    mostrarVector(v2015);
}

int buscarProvincia(const char *cv){
}

void puntoC(){
    Venta reg;
    int pos=0, cant=0, prov;

    while(reg.leerDeDisco(pos++){
        if(reg.getFechaVenta().getAnio()==2020 &&
            reg.getFechaVenta().getMes()<7 ){
            prov=buscarProvincia(reg.getCodigoVendedor());
            if(prov>=7 &&prov<=9)
                cant++;
        }
    }
    cout<<"LA CANTIDAD DE VENTAS DE LOS VENDEDORES DE LITORAL FUE: "<<<cant<<endl;
}
```