```
#include<iostream>
#include<conio.h>
#include<stdio.h>

#define DIM 100 // supongo 100 productos
using namespace std;

struct Producto {
    int codigo;
    int precioCompra;
    int precioVenta;
    int stockInicial;
    int stockActual;
    int stockMinimo;
};



void cargaProducto(Producto vecProductos[],int codigo, int &dim){
    int i=0;
    while(codigo!=0){
        vecProductos[i].codigo = codigo;
        cout<<"Ingrese el PRECIO DE COMPRA: ";
        cin>>vecProductos[i].precioCompra;
        cout<<"Ingrese el PRECIO DE VENTA: ";
        cin>>vecProductos[i].precioVenta;
        cout<<"Ingrese el STOCK INICIAL: ";
        cin>> vecProductos[i].stockInicial;
        cout<<"Ingrese el STOCK MINIMO: ";
        cin>>vecProductos[i].stockMinimo;

        vecProductos[i].stockActual = vecProductos[i].stockInicial;


        cout<<"Ingrese el CODIGO DE PRODUCTO (0 para fin):";
        cin>> codigo;

        i++;
    }
    dim =i;
}
int ingresarCodigo(Producto vecProductos[], int dim, int &codigo , int
&indiceCodigo){
    int i;

    do{
         i=0;

        cout<<"Ingrese el CODIGO DE PRODUCTO VENDIDO: ";
        cin>>codigo;

        while(vecProductos[i].codigo!= codigo && i<dim){
```

```
            i++;
        }

        if(i==dim){
            cout<<"ERROR! No se encontro el codigo. Ingrese un codigo
nuevamente."<<endl;
        }


    }while(i==dim);

    indiceCodigo = i;
}


void cargaVenta(Producto vecProductos[], int dim, int &resp ){
    int i,codigo,venta, indiceCodigo;
    while(resp==1){


        ingresarCodigo(vecProductos,dim, codigo, indiceCodigo);

        cout<<"Ingrese la CANTIDAD VENDIDA: ";
        cin>>venta;

        vecProductos[indiceCodigo].stockActual -= venta;




        cout<<"Quiere ingresar venta? (1: Si - 0: No): ";
        cin>> resp;

        while(resp!=1 && resp!=0){
            cout<<"ERROR! Ingrese respuesta valida. Quiere ingresar venta?
(1: Si - 0: No): ";
            cin>>resp;
        }
    }
}

int margenGanancia(int precioVenta, int precioCompra, int stock){
    int ganancia;

    ganancia = (precioVenta - precioCompra) * stock;

    return ganancia;

}

void ganancia(Producto vecProductos[], int dim){
    int i, acumMargen=0;
```

```
    for(i=0;i<dim;i++){
        acumMargen+= margenGanancia( vecProductos[i].precioVenta,
vecProductos[i].precioCompra, vecProductos[i].stockActual);

    }
    //cout<<"La ganancia del mes es de :"<<acumMargen<<endl;
}

void gananciaD(Producto vecProductos[], int dim, int &acum){
    int i;

    for(i=0;i<dim;i++){
        acum += (vecProductos[i].precioVenta -
vecProductos[i].precioCompra) * vecProductos[i].stockActual;
    }
}



bool pedir(Producto unProducto){
    return(unProducto.stockActual >= unProducto.stockMinimo);
}


int menorPrecio(Producto vecProductos[], int dim){
    int i, iMenorPrecio=0;

    for(i=1;i<dim;i++){
        if(vecProductos[i].precioVenta <
vecProductos[iMenorPrecio].precioVenta){
            iMenorPrecio =i;
        }
    }
    return iMenorPrecio;
}

int mayorPrecio(Producto vecProductos[], int dim){
    int i, iMayorPrecio=0;

    for(i=1;i<dim;i++){
        if(vecProductos[i].precioVenta >
vecProductos[iMayorPrecio].precioVenta){
            iMayorPrecio =i;
        }
    }
    return iMayorPrecio;
}

void maximoyMinimo(Producto vecProductos[], int dim, int
&menorPrecioVenta, int &mayorPrecioVenta){
    menorPrecioVenta = vecProductos[menorPrecio (vecProductos,
dim)].precioVenta;
    mayorPrecioVenta = vecProductos[mayorPrecio (vecProductos,
dim)].precioVenta;
```

```
}


bool stockPorcentaje(Producto unProducto, int porcentaje){
    return(unProducto.stockActual >= ((unProducto.stockMinimo * porcentaje
/ 100) + unProducto.stockMinimo));

}

void sobreStock(Producto vecProductos[], int dim, Producto
vecProductosSobreStock[], int &dimSobreStock){
    int i, x=0;
    for(i=0;i<dim;i++){
        if(stockPorcentaje(vecProductos[i], 50)){
            vecProductosSobreStock[x]= vecProductos[i];
            x++;
        }
    }
    dimSobreStock =x;
}

void imprimir(Producto vecProductos[], int dim){
    int i;
    for(i=0;i<dim;i++){
        cout<<"******** PRUDUCTO "<<i+1<<" ********"<<endl;
        cout<<"CODIGO: "<<vecProductos[i].codigo<<endl;
        cout<<"PRECIO DE COMPRA: "<<vecProductos[i].precioCompra<<endl;
        cout<<"PRECIO DE VENTA: "<<vecProductos[i].precioVenta<<endl;
        cout<<"STOCK INICIAL: "<<vecProductos[i].stockInicial<<endl;
        cout<<"STOCK MINIMO: "<<vecProductos[i].stockMinimo<<endl;
        cout<<"STOCK ACTUAL: "<<vecProductos[i].stockActual<<endl;
    }
}

int alcanzanStockMinimo(Producto vecProductos[], int dim){
    int i,contProductosAlncanzanMinimo =0;
    for(i=0;i<dim;i++){
        if(pedir(vecProductos[i])){
            contProductosAlncanzanMinimo++;
        }
    }
}

int main(){
    int i,codigo,resp ,venta, codProducto, dim,
dimSobreStock,acumGananciaMes=0, mayorPrecioVenta, menorPrecioVenta;

    Producto vecProductos[DIM];
    Producto vecProductosSobreStock[DIM];

    cout<<"******** CARGA PRODUCTOS ********"<<endl;
    cout<<"Ingrese el CODIGO DE PRODUCTO (0 para fin):";
    cin>> codigo;
```

```
    //1)
    cargaProducto(vecProductos,codigo,dim);
    imprimir(vecProductos,dim);

    cout<<"******** CARGA VENTA ********"<<endl;
    cout<<"Quiere ingresar venta? (1: Si - 0: No): ";
    cin>> resp;

    while(resp!=1 && resp!=0){
        cout<<"ERROR! Ingrese respuesta valida. Quiere ingresar venta? (1:
Si - 0: No): ";
        cin>>resp;
    }

    cargaVenta(vecProductos, dim, resp);
    imprimir(vecProductos,dim);
    //2) ganancia del mes

    gananciaD(vecProductos, dim, acumGananciaMes);
    cout<<"Ganancia del mes: "<< acumGananciaMes<<endl;



    //3) cantidad de productos que alcanzan el stock minimo
    cout<<"Cantidad de productos que alcanzan el stock minimo: "
<<alcanzanStockMinimo(vecProductos, dim)<<endl;

    //4) Menor y Mayor precio de venta
    maximoyMinimo(vecProductos, dim, menorPrecioVenta, mayorPrecioVenta);
    cout<<"Menor precio de venta: "<< menorPrecioVenta<<endl;
    cout<<"Mayor precio de venta: "<< mayorPrecioVenta<<endl;

    //5) Cantidad de productos y datos de dichos productos  cuyo stock
actual supere en 50 %
    cout<<"Cantidad de productos y datos de dichos productos  cuyo stock
actual supere en 50 %"<<endl;
    sobreStock(vecProductos,dim,vecProductosSobreStock,dimSobreStock);
    cout<<"Cantidad : "<<dimSobreStock<<endl;
    imprimir(vecProductosSobreStock,dimSobreStock);


    getch();
    return 0;
}
```