**Consigna**

La empresa CAUCHOS SA dedicada al rubro de la gomería desea analizar las ventas realizadas durante el día. En ella trabajan 5 gomeros y se comercializan 15 tipos de cubiertas distintas de las que cuenta con el siguiente stock informado como lote 1.

* Numero de cubierta (entero de 1 a 15)
* Unidades en Stock (entero de 1 a 5000)
* Stock mínimo (entero mayor que 0)

Al finalizar el horario laboral, se nos suministran diversas planillas informadas como lote 2 donde figura por cada venta de una determinada cubierta (sabiendo que al menos existe una venta): el número de gomero que realizó la venta, el número de cubierta y el importe de dicha venta.

LOTE 2

* Número de gomero (de 30 a 50 no consecutivos)
* Número de cubierta (de 1 a 15)
* Importe (real mayor que 0)
* Cantidad vendida (entero mayor que 0)

Dichos datos, se ingresarán por teclado en forma desordenada, hasta el ingreso de un número de gomero 0 (cero). Se puede repetir durante el día, la venta de una misma cubierta para un mismo gomero, y no hay necesariamente venta de todas las cubiertas.

Para poder probar el programa, tendrá que tener cargado previamente los datos del primer lote y la lista de gomeros que se usará para el lote 2.

Desarrollar el diagrama de lógica y su correspondiente codificación en lenguaje “C” completo y probado que resuelva los siguientes puntos siendo ud. quien decide el orden de resolución:

1. Construya una función llamada CargaInicial() que cargue dentro de ella en las estructuras convenientes, Nro de Cubierta, Unidades en stock y stock mínimo, preguntando dentro de la misma bajo la opción ‘S’ si se desea pedir por teclado dato por dato o ‘N’, si desea cargar directamente el lote de prueba suministrado al pie de la evaluación. Las dos formas deben estar programadas. No es necesario mandar como parámetro los límites de cada ítem para usar en la validación, pueden usarlos directamente dentro de la función para facilitar resolución. (Leer consigna 4), En el caso de la opción cargarse el lote automáticamente, puede expresarlo en el diagrama en un comentario donde corresponda cargarlo para facilitar la diagramación.
2. Construya una función llamada CargaGomero(), que cargue el listado de gomeros que usará posteriormente.
3. Ingresar la fecha del día y validarla desarrollando mínimamente las siguientes funciones anidadas:
   1. EsFechaValida() – Recibe todos los parámetros–retorna: correcta 1 / incorrecta 0.
   2. Bisiesto() Recibe el año y retorna 1 si lo es o 0 si no lo es.
4. Validar el ingreso del número de gomero, cubierta y stock en cada caso, a través de una única y misma función llamada ControlDatos().
5. En el segundo lote, validar que el gomero exista en los cargados previamente en el punto 2. (Utilizar función Busqueda())
6. Verificar en cada venta si hay stock suficiente para poder realizarla. Se deberá actualizar el stock correspondiente y si no alcanzan rechazar la venta con un mensaje aclaratorio.
7. Si la venta se concretó y el stock quedó por debajo del mínimo, informarlo
8. Informar el Número de cubierta que vendió la menor cantidad en una venta durante el día (hay un único mínimo).
9. Informar el Número de gomero que, en total, realizó la menor recaudación considerando todas las ventas (puede haber múltiples mínimos).
10. Informar los números de gomeros que no registraron ventas.
11. Informar los gomeros con más ventas (cantidad de ventas realizadas y NO importes NI cantidades) durante el día.
12. Informar en forma ordenada de mayor a menor los montos recaudados por cubierta. Para el ordenamiento, utilizar una función llamada Ordenamiento() con parámetros. Una vez ordenado, mostrar en el programa principal la lista con los Números de Cubiertas y Montos

**Ventas de fecha dd/mm/aaaa**

Cubierta Importe

10 12000.00

15 11200.00

11 9500.00

5 7532.00

TOTAL VENTAS: $40232.00

1. Informar las cubiertas que NO están por debajo del Stock mínimo.
2. Informar Importe total perdido por ventas rechazadas.

**Lote de Prueba a Utilizar**

Gomero a utilizar {33,35,41,44,50}

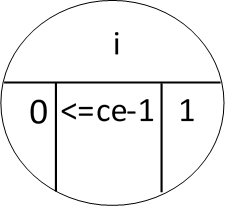
Cubiertas {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}

Unidades en Stock {100,5,55,5000,450,35,11,22,1872,10,4800,8,88,81,893}

Stock mínimo {25,2,22,2200,150,10,2,1,600,1,1000,2,2,22,60}

PonerEnCeroReal

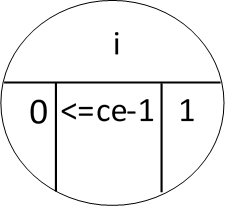
float v[], int ce void



v[i]=0

PonerEnCero

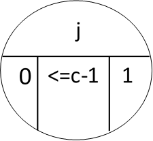
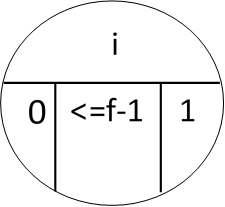
int v[], int ce void



v[i]=0

PonerMatEnCero

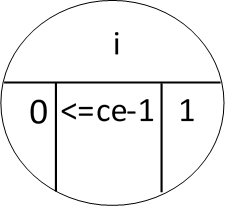
float m[][15], int f, int c void



m[i][j]=0

MostrarIgualQueVParalelos

float v[], int a[], int ce, int x void



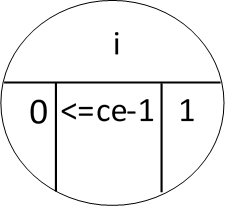
v[i]==x

a[i]

return s

SumaReal

float v[], int ce float



s+=v[i]

s=0

Bisiesto

int a int

a%4==0&&a%100!=0||a%400==0

return 1

return 0

esFechaValida

int d, int m, int a int

r=0

a>=1950&&a<=2020

m>=1&&m<=12

d>=1&&d<=CantidadDiasDelMes(m,a)

r=1

return r

CantidadDiasDelMes

int m, int a int

cdm=0

m==4||m==6||m==9||m==11

m==2

cdm=31

return cdm

cdm=30

cdm=28+Bisiesto(a)

p=-1&&i<ce

v[i]==x

p=i

i++

Busqueda

int v[], int ce, int x int

i=0

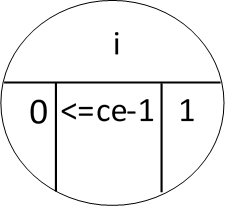
p=-1



return p

Listado

int a[], float b[] int ce void



“CUBIERTA IMPORTE”

b[i]>0

a[i],b[i]

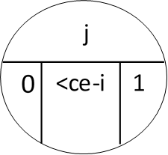
Ordenamiento

float a[], int b[], int ce void

i=0

hubocambio=0

a[j-1]<a[j]



aux=a[j-1]

a[j-1]=a[j]

a[j]=aux

aux2=b[j-1]

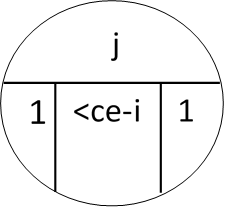
b[j-1]=b[j]

b[j]=aux2

hubocambio=1

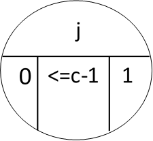
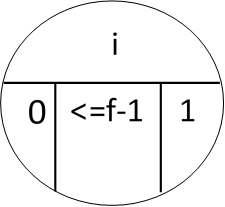
i++

hubocambio



SumarFilaMatEnV

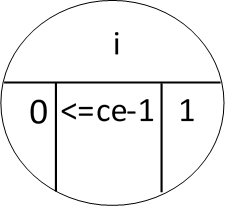
int m[][15], float v[], int f, int c void



v[i]+=m[i][j]

MostrarIgualQueVParalelos2

int v[], int a[], int ce, int x void



v[i]==x

a[i]

opcion

!(opcion==’N’||opcion==’S’)

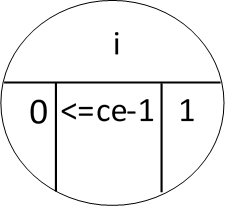
“1) Presione ‘S’ para cargar los datos manualmente o ‘N’ para cargar el lote de prueba: ”

CargaInicial

int a[], int b[], int ce void

opcion==’S’

Se ingresa/asigna automáticamente los valores de los lotes a utilizar



“Ingrese unidades de stock”:

ustock=ControlDatos(1,5000)

“Stock de la cubierta numero”,i+1

a[i]=ustock

stockmin

!(stockmin>0)

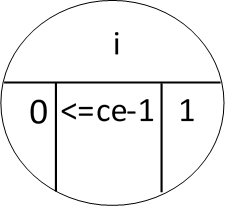
b[i]=stockmin

“Ingrese stock minimo”:

return menor

MenorRealPositivo

float v[], int ce float



menor=0

v[i]>0&&(menor==0||menor>v[i])

menor=v[i]

ControlDatos

int limi, int lims int

band=0

band==1

“Dato incorrecto, ingrese nuevamente: “

band=1

dato

!(dato>=limi&&dato<=lims)

return dato

esDatoVal

int x, int limi, int lims int

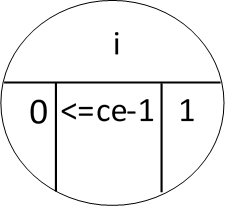
x>=limi&&x<=lims

return 1

return 0

MostrarMayorQue

int a[], int b[], int ce void



a[i]>b[i]

i+1

IngrDatoVal3

int lim1, int lim2, int lim3 int

band=0

band==1

“Dato incorrecto, ingrese nuevamente: “

band=1

dato

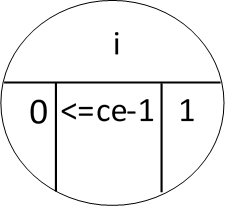
esDatoVal(dato,lim1,lim2)==0&&dato!=lim3

return dato

return mayor

Maximo

int v[], int ce int



mayor=0

mayor==0||v[i]>mayor

mayor=v[i]

CargaGomero

int a[], int b[], int ce void

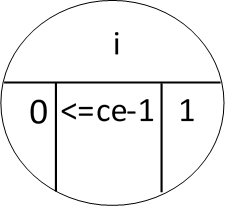
“2) CARGA DE LISTADO DE GOMEROS”

opcion

!(opcion==’N’||opcion==’S’)

“Presione ‘S’ para cargar los datos manualmente o ‘N’ para cargar el lote de prueba: ”

opcion==’S’



“Ingrese el numero de gomero (empleado numero”,i+1,”)”

pos=Busqueda(v,i,nrogomero)

pos!=-1

v[i]=nrogomero

nrogomero=ControlDatos(30,50)

pos!=-1

“Dato duplicado. Ingrese otro”:

v[0]=33

v[1]=35

v[2]=41

v[3]=44

v[4]=50

Ventas

band=0

cantmin=0

impVentasRechazadas=0

PonerEnCero(VCantVentasPorGomero,5)

PonerEnCeroReal(VVentasPorGomero,5)

PonerEnCeroReal(VVentasPorCubierta,15)

PonerMatEnCero(MVentas,5,15)

band==1

“Fecha no valida“

band=1

d,m,a

esFechaValida(d,m,a)==0

“3) Ingrese fecha actual en formato DD-MM-AAA: “

CargaInicial(VUniStock,VStockMin,15)

CargaGomero(VGomero,5)

“5) VENTAS DE LA EMPRESA CAUCHOS SA”

“Ingrese el numero de gomero cargado anteriormente (0 para terminar): ”

pos=Busqueda(VGomero,5,nrogomero)

pos==-1&&nrogomero!=0

nrogomero=IngrDatoVal3(30,50,0)

pos==-1&&nrogomero!=0

“El numero ingresado no aparece en la lista”

“Ingrese el numero de cubierta (entre 1 y 15): ”

nrogomero!=0

nrocubierta=ControlDatos(1,15)

importe

!(importe>0)

“Ingrese importe”:

cant

!(cant>0)

“Ingrese cantidad vendida: ”

venta=importe\*cant

VUniStock[nrocubierta-1]>=cant

“6) No se pudo realizar la venta (no hay stock suficiente)”

impVentasRechazadas+=venta

“Ingrese el numero de gomero cargado anteriormente (0 para terminar): ”

pos=Busqueda(VGomero,5,nrogomero)

pos==-1&&nrogomero!=0

nrogomero=IngrDatoVal3(30,50,0)

pos==-1 && nrogomero!=0

“El numero ingresado no aparece en la lista”

cantmin==0||cant<cantmin

canatmin=cant

nrocubiertamin=nrocubierta

MVentas[pos][nrocubierta-1]+=venta

VCantVentasPorGomero[pos]++

stockactual=VUniStock[nrocubierta-1]-cant

stockactual<VStockMin[nrocubierta-1]

“7) El stock quedo por debajo del minimo”

VUniStock[nrocubierta-1]=stockactual

VUniStock[nrocubierta-1]=stockactual

“Ingrese el numero de gomero cargado anteriormente (0 para terminar): ”

pos=Busqueda(VGomero,5,nrogomero)

pos==-1&&nrogomero!=0

nrogomero=IngrDatoVal3(30,50,0)

pos==-1&&nrogomero!=0

“El numero ingreasado no aparece en la lista”

“8) Numero de cubierta que vendio la menor cantidad en una venta durante el dia: “,nrocubiertamin

SumarFilaMatEnV(MVentas,VVentasPorGomero,5,15)

min=MenorRealPositivo(VVentasPorGomero,5)

“9) Numero/números de gomero/gomeros que realizo/realizaron la menor recaudación de ventas: “

MostrarIgualQueVParalelos(VVentasPorGomero,VGomero,5,min)

“10) Numero/números de gomero/gomeros que no registraron ventas: “

MostrarIgualQueVParalelos(VVentasPorGomero,VGomero,5,0)

max=Maximo(VCantVentasPorGomero,5)

“11) Gomeros con mas ventas”

MostrarIgualQueVParalelos2(VCantVentasPorGomero,VGomero,5,max)

Ordenamiento(VVentasPorGomero,VGomero,5)

“12) VENTAS DE FECHA “,d,”/”,m,”/”,a

Listado(VGomero,VVentasPorGomero,5)

“TOTAL DE VENTAS: $”,SumaReal(VVentasPorGomero,5)

“13) Cubiertas que no están por debajo del stock minimo: ”

MostarMayorQue(VUniStock,VStock,15)

“14) Importe total perdido por ventas rechazadas: $”,impVentasRechazadas

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

int esDatoVal(int,int,int);

int ControlDatos(int,int);

int IngrDatoVal3(int,int,int);

int esFechaValida(int,int,int);

int CantidadDiasDelMes(int,int);

int Bisiesto(int);

void PonerEnCero(int[],int);

void PonerEnCeroReal(float[],int);

void PonerMatEnCero(float[][15],int,int);

void CargaInicial(int[],int[],int);

void CargaGomero(int[],int);

int Busqueda(int[],int,int);

void Ordenamiento(float[],int[],int);

void SumarFilaMatEnV(float[][15],float[],int,int);

float SumaReal(float[],int);

float MenorRealPositivo(float[],int);

int Maximo(int[],int);

void MostrarIgualQueVParalelos(float[],int[],int,float);

void MostrarIgualQueVParalelos2(int[],int[],int,int);

void Listado(int[],float[],int);

void MostrarMayorQue(int[],int[],int);

int main()

{

int VGomero[5],VUniStock[15],VStockMin[15],d,m,a,band=0,VCantVentasPorGomero[5],nrogomero,pos,nrocubierta,cant,cantmin=0,nrocubiertamin,stockactual,max;

float MVentas[5][15],importe,venta,impVentasRechazadas=0,VVentasPorCubierta[15],VVentasPorGomero[5],min;

PonerEnCero(VCantVentasPorGomero,5);

PonerEnCeroReal(VVentasPorCubierta,15);

PonerEnCeroReal(VVentasPorGomero,5);

PonerMatEnCero(MVentas,5,15);

do

{

if(band==1)

printf("\nFecha no valida\n");

band=1;

printf("\n3) Ingrese fecha actual en formato DD-MM-AAAA: ");

scanf("%d%d%d",&d,&m,&a);

}while(esFechaValida(d,m,a)==0);

system("cls");

CargaInicial(VUniStock,VStockMin,15);

system("cls");

CargaGomero(VGomero,5);

system("cls");

printf("\n5) VENTAS DE LA EMPRESA CAUCHOS SA\n");

do

{

printf("\nIngrese numero de gomero cargado anteriormente (0 para terminar): ");

nrogomero=IngrDatoVal3(30,50,0);

pos=Busqueda(VGomero,5,nrogomero);

if(pos==-1&&nrogomero!=0)

printf("\nEl numero ingresado no aparece en la lista\n");

}while(pos==-1&&nrogomero!=0);

while(nrogomero!=0)

{

printf("\nIngrese el numero de cubierta (entre 1 y 15): ");

nrocubierta=ControlDatos(1,15);

do

{

printf("\nIngrese importe: ");

scanf("%f",&importe);

}while(!(importe>0));

do

{

printf("\nIngrese cantidad vendida: ");

scanf("%d",&cant);

}while(!(cant>0));

venta=(float)importe\*cant;

if(VUniStock[nrocubierta-1]>=cant)

{

if(cantmin==0||cant<cantmin)

{

cantmin=cant;

nrocubiertamin=nrocubierta;

}

MVentas[pos][nrocubierta-1]+=venta;

VCantVentasPorGomero[pos]++;

stockactual=VUniStock[nrocubierta-1]-cant;

if(stockactual<VStockMin[nrocubierta-1])

{

printf("\n7) El stock quedo por debajo del minimo\n");

VUniStock[nrocubierta-1]=stockactual;

}

else

VUniStock[nrocubierta-1]=stockactual;

do

{

printf("\nIngrese numero de gomero cargado anteriormente (0 para terminar): ");

nrogomero=IngrDatoVal3(30,50,0);

pos=Busqueda(VGomero,5,nrogomero);

if(pos==-1&&nrogomero!=0)

printf("\nEl numero ingresado no aparece en la lista\n");

}while(pos==-1&&nrogomero!=0);

}

else

{

printf("\n6) No se pudo realizar la venta (no hay stock suficiente)\n");

impVentasRechazadas+=venta;

do

{

printf("\nIngrese numero de gomero cargado anteriormente (0 para terminar): ");

nrogomero=IngrDatoVal3(30,50,0);

pos=Busqueda(VGomero,5,nrogomero);

if(pos==-1&&nrogomero!=0)

printf("\nEl numero ingresado no aparece en la lista\n");

}while(pos==-1&&nrogomero!=0);

}

}

system("cls");

printf("\n8) Numero de cubierta que vendio la menor cantidad en una venta durante el dia: %d\n",nrocubiertamin);

SumarFilaMatEnV(MVentas,VVentasPorGomero,5,15);

min=MenorRealPositivo(VVentasPorGomero,5);

printf("\n9) Numero/numeros de gomero/gomeros que realizo/realizaron la menor recaudacion de ventas: \n");

MostrarIgualQueVParalelos(VVentasPorGomero,VGomero,5,min);

printf("\n10) Numeros/numeros de gomero/gomeros que no registraron ventas: \n");

MostrarIgualQueVParalelos(VVentasPorGomero,VGomero,5,0);

max=Maximo(VCantVentasPorGomero,5);

printf("\n11) Gomeros con mas ventas\n");

MostrarIgualQueVParalelos2(VCantVentasPorGomero,VGomero,5,max);

Ordenamiento(VVentasPorGomero,VGomero,5);

printf("\n12) VENTAS DE FECHA %d/%d/%d\n",d,m,a);

Listado(VGomero,VVentasPorGomero,5);

printf("\nTOTAL DE VENTAS: $ %.2f\n",SumaReal(VVentasPorGomero,5));

printf("\n13) Cubiertas que no estan por debajo del stock minimo: \n");

MostrarMayorQue(VUniStock,VStockMin,15);

printf("\n14) Importe total perdido por ventas rechazadas: $ %.2f\n",impVentasRechazadas);

}

int esDatoVal(int x,int limi,int lims)

{

if(x>=limi&&x<=lims)

return 1;

else

return 0;

}

int IngrDatoVal3(int lim1,int lim2,int lim3)

{

int dato,band=0;

do

{

if(band==1)

printf("\nDato incorrecto, ingrese nuevamente: ");

band=1;

scanf("%d",&dato);

}while(esDatoVal(dato,lim1,lim2)==0&&dato!=lim3);

return dato;

}

int ControlDatos(int limi,int lims)

{

int dato,band=0;

do

{

if(band==1)

printf("\nDato incorrecto, ingrese nuevamente: ");

band=1;

scanf("%d",&dato);

}while(!(dato>=limi&&dato<=lims));

return dato;

}

int esFechaValida(int d,int m,int a)

{

int r=0;

if(a>=1950&&a<=2020)

if(m>=1&&m<=12)

if(d>=1&&d<=CantidadDiasDelMes(m,a))

r=1;

return r;

}

int CantidadDiasDelMes(int m,int a)

{

int cdm=0;

if(m==4||m==6||m==9||m==11)

cdm=30;

else

if(m==2)

cdm=28+Bisiesto(a);

else

cdm=31;

return cdm;

}

int Bisiesto(int a)

{

if(a%4==0&&a%100!=0||a%400==0)

return 1;

else

return 0;

}

void PonerEnCero(int v[],int ce)

{

int i;

for(i=0;i<=ce-1;i++)

v[i]=0;

}

void PonerEnCeroReal(float v[],int ce)

{

int i;

for(i=0;i<=ce-1;i++)

v[i]=0;

}

void PonerMatEnCero (float m[][15],int f,int c)

{

int i,j;

for(i=0;i<=f-1;i++)

for(j=0;j<=c-1;j++)

m[i][j]=0;

}

void CargaInicial (int a[],int b[],int ce)

{

int ustock,stockmin,i;

char opcion;

printf("\n1) CARGA LOTE 1\n");

do

{

printf("\nPresione 'S' para cargar los datos manualmente o 'N' para cargar lote de prueba: ");

fflush(stdin);

scanf("%c",&opcion);

}while(!(opcion=='N'||opcion=='S'));

if(opcion=='S')

{

for(i=0;i<=ce-1;i++)

{

printf("\nSTOCK DE LA CUBIERTA NUMERO %d\n",i+1);

printf("\nIngrese unidades en stock (entre 1 a 5000): ");

ustock=ControlDatos(1,5000);

a[i]=ustock;

do

{

printf("\nIngrese stock minimo: ");

scanf("%d",&stockmin);

}while(!(stockmin>0));

b[i]=stockmin;

}

}

else

{

a[0]=100;a[1]=5;a[2]=55;a[3]=5000;a[4]=450;a[5]=35;a[6]=11;a[7]=22;a[8]=1872;a[9]=10;a[10]=4800;a[11]=8;a[12]=88;a[13]=81;a[14]=893;

b[0]=25;b[1]=2;b[2]=22;b[3]=2200;b[4]=150;b[5]=10;b[6]=2;b[7]=1;b[8]=600;b[9]=1;b[10]=1000;b[11]=2;b[12]=2;b[13]=22;b[14]=60;

}

}

void CargaGomero(int v[],int ce)

{

int i,nrogomero,pos;

char opcion;

printf("\n2) CARGA DEL LISTADO DE GOMEROS\n");

do

{

printf("\nPresione 'S' para cargar los datos manualmente o 'N' para cargar lote de prueba: ");

fflush(stdin);

scanf("%c",&opcion);

}while(!(opcion=='N'||opcion=='S'));

if(opcion=='S')

for(i=0;i<=ce-1;i++)

{

do

{

printf("\nIngrese el numero de gomero (empleado numero %d): ",i+1);

nrogomero=ControlDatos(30,50);

pos=Busqueda(v,i,nrogomero);

if(pos!=-1)

printf("\nDato duplicado. Ingrese otro\n");

}while(pos!=-1);

v[i]=nrogomero;

}

else

{

v[0]=33;v[1]=35;v[2]=41;v[3]=44;v[4]=50;

}

}

int Busqueda(int v[],int ce,int x)

{

int i=0,p=-1;

while(p==-1&&i<ce)

{

if(v[i]==x)

p=i;

else

i++;

}

return p;

}

void SumarFilaMatEnV (float m[][15],float v[],int f,int c)

{

int i,j;

for(i=0;i<=f-1;i++)

for(j=0;j<=c-1;j++)

v[i]+=m[i][j];

}

float MenorRealPositivo (float v[],int ce)

{

float menor=0;

int i;

for(i=0;i<=ce-1;i++)

if(v[i]>0&&(menor==0||menor>v[i]))

menor=v[i];

return menor;

}

void MostrarIgualQueVParalelos (float v[],int a[],int ce,float x)

{

int i;

for(i=0;i<=ce-1;i++)

if(v[i]==x)

printf("\n%d\n",a[i]);

}

int Maximo(int v[],int ce)

{

int i,mayor=0;

for(i=0;i<=ce-1;i++)

if(mayor==0||v[i]>mayor)

mayor=v[i];

return mayor;

}

void MostrarIgualQueVParalelos2(int v[],int a[],int ce,int x)

{

int i;

for(i=0;i<=ce-1;i++)

if(v[i]==x)

printf("\n%d\n",a[i]);

}

void Ordenamiento(float a[],int b[],int ce)

{

int i=0,hubocambio,j,aux2;

float aux;

do

{

hubocambio=0;

for(j=1;j<ce-i;j++)

if(a[j-1]<a[j])

{

aux=a[j-1];

a[j-1]=a[j];

a[j]=aux;

aux2=b[j-1];

b[j-1]=b[j];

b[j]=aux2;

hubocambio=1;

}

i++;

}while(hubocambio);

}

float SumaReal(float v[],int ce)

{

int i;

float s=0;

for(i=0;i<=ce-1;i++)

s+=v[i];

return s;

}

void Listado(int a[],float b[],int ce)

{

int i;

printf("\nCUBIERTA\tIMPORTE\n");

for(i=0;i<=ce-1;i++)

if(b[i]>0)

printf("\n%d\t\t$ %.2f\n",a[i],b[i]);

}

void MostrarMayorQue (int a[],int b[],int ce)

{

int i;

for(i=0;i<=ce-1;i++)

if(a[i]>b[i])

printf("\n%d\n",i+1);

}