

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <conio.h>
#define lim 100
//#define
/*****DECLARACION TIPOS DE
DATO*****/
typedef struct
{
    int codProd;
    char desc[51];
    int canti;
    float pUni;
    int st;
}producto;

typedef struct
{
    int nroPed;
    int cod;
    char tipoE[12];
    int zona;
    int cantP;
}pedido;

/*****/
int LeeControlCantidad(int,int,int,int);
int LECTURA_CATALOGOS(producto [], int, int);
void MOSTRAR(producto [], int);
int BUSQUEDA(producto [], int, int);
void ORDEN(producto [], float [], int);
void LISTAR(producto [], float [], int, int);
void GRABAR(producto [], int);
void LecturaArch(void);
void LecturaArch1(void);
void LecturaArch2(void);
/*****MAIN*****/
main()
{
    int puzzle, ce, pos, faltante, contDomicilio=0;
    float vRecXProd[lim]={0};
    producto V[lim];
    pedido info;
    FILE *pp, *pno, *pe;

    printf("\n          NEGOCIO DE PUZZLES          \n");
    printf(" *****          \n  ");

    printf("\n -----          ");
    printf("\n          SELECCION DE CANTIDAD DE PIEZAS          \n");
    printf(" -----          ");
    printf("\n Ingrese, por favor, una cantidad de piezas de puzzle(500-1000-1500-2000): ");
    puzzle=LeeControlCantidad(500,1000,1500,2000); //A
    ce=LECTURA_CATALOGOS(V, puzzle, lim); //B
    MOSTRAR(V, ce);

    printf("\n\n");
    system("pause");
    printf("\n -----          ");
    printf("\n          PROCESAMIENTO DE LOS PEDIDOS          \n");
    printf(" -----          ");

```

```

pp=fopen("PEDIDOS.dat","rb");
pno=fopen("NO_CORRESPONDE.dat","wb");
pe=fopen("PARA_ENTREGAR.dat","wb");

if ((pp == NULL) || (pno == NULL) || (pe == NULL))
{
    printf(" ERROR APERTURA EN ALGUN ARCHIVO!!!");
    getch();
    exit(1);
}

printf("\n La cantidad de piezas de puzzles previamente escogida fue de:%4d", puzzle);
fread(&info,sizeof(pedido),1,pp);
while(!feof(pp))
{
    pos=BUSQUEDA(V, info.cod, ce);
    if(pos != -1) //encontró
    {
        if(V[pos].st>=info.cantP) //stock suficiente
        {
            V[pos].st-=info.cantP;
            vRecXProd[pos]+=info.cantP*V[pos].pUni;

            if(strcmpi(info.tipoE,"DOMICILIO") == 0) //tipo entrega es DOMICILIO
            {
                if(info.zona == 1)
                    contDomicilio+=0;
                else
                {
                    if(info.zona == 2)
                        contDomicilio+=100;
                    else
                    {
                        if(info.zona == 3)
                            contDomicilio+=500;
                    }
                }
                fwrite(&info,sizeof(pedido), 1, pe);
            }
        }
        else //stock insuficiente
        {
            printf("\n STOCK INSUFICIENTE del codigo de producto %-30s", V[pos].desc);
        }
    }
    else //no encontró
        fwrite(&info,sizeof(pedido),1,pno);

    fread(&info,sizeof(pedido),1,pp);
}
fclose(pp);
fclose(pno);
fclose(pe);

printf("\n\n");
system("pause");
printf("\n -----");
printf("\n      CARACTERISTICAS DEL PROCESAMIENTO DE PEDIDOS      \n");
printf("\n -----");

printf("\n Por entregas a domicilio se recaudo un total de: $%4d", contDomicilio);

ORDEN(V, vRecXProd, ce);
LISTAR(V, vRecXProd, puzzle, ce);
GRABAR(V, ce);

```

```

printf("\n\n");
system("pause");
//LA LECTURA DE LOS ARCHIVOS ES SOLO INFORMATIVA(NO SE PIDE EN EL ENUNCIADO)
printf("\n -----");
printf("\n      LECTURA DE ARCHIVOS      \n");
printf(" ----- \n");
LecturaArch(); //NO_CORRESPONDE.dat
printf("\n\n");
LecturaArch1(); //PARA_ENTREGAR.dat
printf("\n\n");
LecturaArch2(); //ACTU_PUZZLE.dat
printf("\n\n");

fclose(pp);
fclose(pno);
fclose(pe);
return 0;
}
/*****FUNCIONES*****/
int LeeControlCantidad(int Valor1, int Valor2, int Valor3, int Valor4) //PARA SER UNA
FUNCION GENERICA TENDRIA QUE ENVIAR LOS VALORES DE LAS PIEZAS COMO PARAMETRO
{
    int valor;
    do{
        scanf("%d", &valor);
        if(valor != Valor1 && valor != Valor2 && valor != Valor3 && valor != Valor4)
            printf(" Valor erroneo!!!\n");
    }while(valor != Valor1 && valor != Valor2 && valor != Valor3 && valor != Valor4);

    return (valor);
}
/*****/
int LECTURA_CATALOGOS(producto V[], int valor, int N)
{
    FILE *pf;
    producto info;
    int I=0;

    pf = fopen("CATALOGO.dat", "rb");
    if(pf==NULL)
    {
        printf(" No se pudo abrir el archivo CATALOGO.dat!!! \n");
        system("pause");
        exit(1);
    }
    fread(&info, sizeof(producto), 1, pf);
    while(!feof(pf) && I<N)
    {
        if(info.canti == valor && info.st > 0)
        {
            V[I] = info;
            I++;
        }
        fread(&info, sizeof(producto), 1, pf);
    }
    fclose(pf);

    return(I);
}
/*****/
void MOSTRAR(producto V[], int N)
{
    int I;
    printf("\n\n      INFORMACION DEL ARCHIVO CATALOGO.DAT EN MEMORIA");
    printf("\n      CODIGO PRODUCTO      DESCRIPCION      CANTIDAD DE PIEZAS");

```

```

PRECIO UNITARIO   STOCK ACTUAL");
    for (I=0; I<N; I++)
        printf("\n          %4d          %-30s   %4d          %10.2f          %4d",
V[I].codProd, V[I].desc, V[I].canti, V[I].pUni, V[I].st);

}
/*****/
int BUSQUEDA(producto V[], int dato, int ce)
{
    int pos=-1,i=0;

    while(i<ce && pos==-1)
    {
        if(V[i].codProd==dato)
            pos=i;
        else
            i++;
    }
    return pos;
}
/*****/
void ORDEN (producto V[], float VR[], int N)
{
    int I;
    int DESORDENADO;
    int COTA;
    float AUXFLOAT;

    COTA = N;
    DESORDENADO = 1;
    producto AUX;

    while (DESORDENADO != 0)
    {
        DESORDENADO = 0;
        for(I=1 ; I<COTA ; I++)
            if (V[I-1].st < V[I].st)
            {
                AUX= V[I-1];
                V[I-1]= V[I];
                V[I]= AUX;

                AUXFLOAT = VR[I-1];
                VR[I-1] = VR[I];
                VR[I] = AUXFLOAT;
                DESORDENADO = I;
            }
        COTA = DESORDENADO;
    }
}
/*****/
void LISTAR(producto V[], float VR[], int valor, int ce)
{
    int i;

    printf("\n\n CANTIDAD DE PIEZAS: %4d", valor);
    printf("\n DESCRIPCION DEL PRODUCTO   STOCK ACTUAL   RECAUDACION");
    for(i=0;i<ce;i++)
    {
        printf("\n %-30s   %4d   %10.2f", V[i].desc, V[i].st, VR[i]);
    }
}
/*****/
void GRABAR(producto V[], int n)
{

```

```

int i;
FILE *pf;

pf=fopen("ACTU_PUZZLE.dat","wb");
if (pf==NULL)
{
    printf (" ERROR archivo ACTU_PUZZLE.dat!!! \n");
    getch ();
    exit (1);
}

for(i=0;i<n;i++)
    fwrite(&V[i],sizeof(producto),1,pf);

fclose (pf);
}
/*****/
void LecturaArch(void)
{
    pedido al;
    FILE *pf;

    pf=fopen("NO_CORRESPONDE.dat", "rb");
    if(pf==NULL)
    {
        printf(" Error al entrar al archivo NO_CORRESPONDE.dat!!! \n");
        getch();
        exit(1);
    }
    printf ("\n DATOS GRABADOS: ARCHIVO NO_CORRESPONDE.dat ");
    printf("\n NRO. PEDIDO    CODIGO PRODUCTO    TIPO DE ENTREGA    ZONA DE RESIDENCIA\n");
    printf("CLIENTE    CANTIDAD PEDIDA");
    fread(&al,sizeof(pedido),1,pf);
    while(!feof(pf))
    {
        printf("\n %4d    %4d    %-20s %4d    %4d",
al.nroPed, al.cod, al.tipoE,al.zona, al.cantP);
        fread(&al,sizeof(pedido),1,pf);
    }
    fclose(pf);
}
/*****/
void LecturaArch1(void)
{
    pedido al;
    FILE *pf;

    pf=fopen("PARA_ENTREGAR.dat", "rb");
    if(pf==NULL)
    {
        printf(" Error al entrar al archivo PARA_ENTREGAR.dat!!! \n");
        getch();
        exit(1);
    }
    printf ("\n DATOS GRABADOS: ARCHIVO PARA_ENTREGAR.dat ");
    printf("\n NRO. PEDIDO    CODIGO PRODUCTO    TIPO DE ENTREGA    ZONA DE RESIDENCIA\n");
    printf("CLIENTE    CANTIDAD PEDIDA");
    fread(&al,sizeof(pedido),1,pf);
    while(!feof(pf))
    {
        printf("\n %4d    %4d    %-20s %4d    %4d",
al.nroPed, al.cod, al.tipoE,al.zona, al.cantP);
        fread(&al,sizeof(pedido),1,pf);
    }
    fclose(pf);
}

```

```

}
/*****/
void LecturaArch2(void)
{
    producto al;
    FILE *pf;

    pf=fopen("ACTU_PUZZLE.dat", "rb");
    if(pf==NULL)
    {
        printf(" Error al entrar al archivo ACTU_PUZZLE.dat!!! \n");
        getch();
        exit(1);
    }
    printf ("\n DATOS GRABADOS: ARCHIVO ACTU_PUZZLE.dat ");
    printf("\n          CODIGO PRODUCTO   DESCRIPCION                                CANTIDAD DE PIEZAS
PRECIO UNITARIO   STOCK ACTUAL");
    fread(&al,sizeof(producto),1,pf);
    while(!feof(pf))
    {
        printf("\n          %4d          %-30s   %4d          %6.2f          %4d",
al.codProd, al.desc, al.canti, al.pUni, al.st);
        fread(&al,sizeof(producto),1,pf);
    }
    fclose(pf);
}
/*****/

```