



Universidad Autónoma de Baja California
Facultad de Ingeniería Arquitectura y Diseño



Nombre
Gonzalez Ceseña Adan

Matricula
372799

Grupo
932

Materia
Lenguaje C

Practica 8

Ensenada, Baja California 22/11/2023

Practica 8 estructuras

Código

```
/*
    Nombre del progra : Practica_8_Estructuras_Gonzalez_Cesena_Adan
    Autor : Gonzale Cesena Adan
    Fecha 22/11/2023
    Descripcion Practica 8, creacion de un inventario con varias
funciones
*/

#include <stdio.h>
#include <string.h>

#include <stdlib.h>
#include <time.h>

#include "libad.h"

#define P 500 //Definimos un limite de registros manipulable

//-----Struct-----//
typedef struct producto
{
    char nombre[30];
    int precio;
    int cantidad;
}product;

//-----Prototipos-----//

void menu ();

void agregar (product prod[], int p);
void retirar (product prod[],int p);
void mostrar (product prod[],int p);
void total (product prod[],int p);

void letra (product prod[],int p);
```

```

void precio (product prod[],int p);
void cantidad (product prod[],int p);
//-----Main-----//

int main ()
{
    product reg[P];
    int op, p;
    p=0;
    do
    {
        system("CLS");
        menu();
        op = valid("Fuera de rang",0,6);
        switch(op)
        {
            case 1:
            {
                // Dependiendo si el inventario no esta lleno entrara o
no
                if(p < P)
                {
                    agregar(reg,p);
                    p++;
                }
                else
                {
                    printf("Inventario lleno\n");
                    getch();
                }
                break;
            }
            case 2:
            {
                // Dependiendo si hay datos en el inventario entrara o
no
                if(p > 0)
                {
                    retirar(reg,p);
                    p--;
                }
                else
                {

```

```

        printf("Inventario vacio\n");
        getch();
    }
    break;
}
case 3:
{
    // Dependiendo si hay datos en el inventario entrara o
no
    if(p > 0)
    {
        mostrar(reg,p);
    }
    else
    {
        printf("Inventario vacio\n");
        getch();
    }
    break;
}
case 4:
{
    // Dependiendo si hay datos en el inventario entrara o
no
    if(p > 0)
    {
        total(reg,p);
    }
    else
    {
        printf("Inventario vacio\n");
        getch();
    }
    break;
}
case 5:
{
    // Dependiendo si hay datos suficientes en el
inventario entrara o no
    if(p>1)
    {
        int oper, pos1, pos2,res;

```

```

        printf("Que operacion desea realizar\n1.-Suma
2.-Resta 3.-Multiplicacion 4.-Division\n");
        oper=valid("Fuera de rango\n",1,4);

        if(oper==1)
        {
            printf("Posicion 1\nQue posicion desea
sumar\nDesde 1 a %d\n",p);
            pos1=valid("Posicion fuera de rango",1,p);
            pos1--;

            printf("Posicion 2\nQue posicion desea
sumar\nDesde 1 a %d\n",p);
            pos2=valid("Posicion fuera de rango",1,p);
            pos2--;

            res = reg[pos1].precio + reg[pos2].precio;

            printf("Tu resultado es %d\n",res);
            getch();
        }
        else
        {
            if(oper==2)
            {
                printf("Posicion 1\nQue posicion desea
restar\nDesde 1 a %d\n",p);
                pos1=valid("Posicion fuera de rango",1,p);
                pos1--;

                printf("Posicion 2\nQue posicion desea
restar\nDesde 1 a %d\n",p);
                pos2=valid("Posicion fuera de rango",1,p);
                pos2--;

                res = reg[pos1].precio - reg[pos2].precio;
                printf("Tu resultado es %d\n",res);
                getch();
            }
            else
            {
                if(oper==3)
                {

```

```

printf("Posicion 1\nQue posicion desea
multiplicar\nDesde 1 a %d\n",p);
pos1=valid("Posicion fuera de
rango",1,p);

pos1--;

printf("Posicion 2\nQue posicion desea
multiplicar\nDesde 1 a %d\n",p);
pos2=valid("Posicion fuera de
rango",1,p);

pos2--;

res = reg[pos1].precio *
reg[pos2].precio;

printf("Tu resultado es %d\n",res);
getch();
}
else
{
    if(oper==4)
    {
        printf("Posicion 1\nQue posicion
desea dividir\nDesde 1 a %d\n",p);
pos1=valid("Posicion fuera de
rango",1,p);

pos1--;

printf("Posicion 2\nQue posicion
desea dividir\nDesde 1 a %d\n",p);
pos2=valid("Posicion fuera de
rango",1,p);

pos2--;

res = reg[pos1].precio /
reg[pos2].precio;

printf("Tu resultado es %d\n",res);
getch();
}
}
}
}
}
else

```

```

        {
            printf("datos insuficientes\n");
            getch();
        }
        break;
    }
    case 6:
    {
        if(p>1)
        {
            int ord;
            printf("Por que metodo desea
ordenar\n1.-Alfabeticamente 2.-Precio 3.-Cantidad\n");
            ord=valid("Fuera de rango",1,3);
            if(ord==1)
            {
                letra(reg,p);
            }
            else
            {
                if(ord==2)
                {
                    precio(reg,p);
                }
                else
                {
                    if(ord==3)
                    {
                        cantidad(reg,p);
                    }
                }
            }
        }
        else
        {
            printf("Inventario vacio o con solo 1 producto\n");
            getch();
        }
        break;
    }
    case 0:
    {

```

```

        printf("Seguro que quiere salir?\n1.-Salir
2.-Repetir\n");
        op=valid("Fuera de rango",1,2);
        op--;
    }
}
} while (op != 0);
printf("Que tenga buen dia\n");
return 0;
}

//-----Funciones-----//
/*
    Funcion menu
    imprime las opciones que tendra el usuario para moverse por todo el
codigo
    No recibe parametros
    No retorna nada
*/
void menu ()
{
    printf("Bienvenido\nIngrese una opcion para continuar\n");
    printf("1.-Agregar al inventario\n");
    printf("2.-Eliminar inventario\n");
    printf("3.-Mostrar inventario\n");
    printf("4.-Sumatoria del inventario\n");
    printf("5.-Operacion\n");
    printf("6.-Ordenar\n");
    printf("0.-Salir\n");
}

/*
    Funcion : Agregar
    Agrega los datos basicos que ingrese el usuario a la estrucutra,
cada una con sus respectivas validaciones
    parametros :
        -prod[] = El registro/Estructura
        -p = La posicion en la que se encuentra el programa y en donde
almacenara los datos
    No retorna nada
*/
void agregar (product prod[], int p)
{

```



```

int v;
char cad[30];

do
{
    printf("Ingrese el nombre del producto\n");
    gets(cad);
    v=validCh(cad);
} while (v != 0);
minus(cad);
strcpy(prod[p].nombre, cad);

do
{
    printf("Ingrese la cantidad\n");
    scanf("%d", &prod[p].cantidad);
    if(prod[p].cantidad<1)
    {
        printf("No puede ser menor a 0\n");
    }
}
while(prod[p].cantidad<0);

printf("Ingrese el precio\n");
scanf("%d", &prod[p].precio);

printf("Elemento agregado con exito\n");
getch();
}

/*
Funcion retirar
elimina una posicion del inventario haciendo que se rempaze por la
de la siguiente posicion y asi hasta llegar al limite eliminando una
posicion usada
Parametros :
    -prod[] = El registro/Estructura
    -p = La posicion en la que se encuentra el programa y en donde
almacenara los datos
No retorna nada
*/
void retirar (product prod[],int p)
{

```

```

    int pos;
    printf("Que posicion desea eliminar\n");
    printf("Desde 1 a %d\n",p);
    pos = valid("Fuera de rango",1,p);
    pos--;

    for(pos;pos < p -1 ;pos++)
    {
        prod[pos] = prod[pos+1];
    }

    printf("Listo\n");
    getch();
}

/*
Funcion Mostrar
Imprime todos los registros que el usuario haya ingresado
Parametros :
    -prod[] = El registro/Estructura
    -p = La posicion en la que se encuentra el programa y en donde
almacenara los datos
    No retorna nada
*/
void mostrar (product prod[],int p)
{
    int i;
    for(i=0; i < p ;i++)
    {
        printf("Nombre %s\tCantidad %d\tPrecio
%d\n",prod[i].nombre,prod[i].cantidad,prod[i].precio);
    }
    getch();
}

/*
Funcion total
Hara una suma de todos los precios del inventario y lo imprimira
Parametros :
    -prod[] = El registro/Estructura
    -p = La posicion en la que se encuentra el programa y en donde
almacenara los datos
    No retorna nada
*/

```

```

*/
void total (product prod[],int p)
{
    int total = 0;
    for(int i=0;i<p;i++)
    {
        total = (prod[i].precio * prod[i].cantidad) + total;
    }
    printf("Total : %d\n",total);
    getch();
}

/*
Funcion letra
Ordenara el inventario de forma alfabeticamente
Parametros :
    -prod[] = El registro/Estructura
    -p = La posicion en la que se encuentra el programa y en donde
almacenara los datos
    No retorna nada
*/
void letra (product prod[], int p)
{
    int i,j;
    product temp;
    for (i = 0; i < p - 1; i++)
    {
        for (j = i + 1; j < p; j++)
        {
            if (strcmp(prod[i].nombre, prod[j].nombre) > 0)
            {
                temp = prod[i];
                prod[i] = prod[j];
                prod[j] = temp;
            }
        }
    }
    printf("Listo\n");
    getch();
}

/*
Funcion letra

```

```

    Ordenara el inventario de conforme el precio, de mayor a menor
    Parametros :
        -prod[] = El registro/Estructura
        -p = La posicion en la que se encuentra el programa y en donde
almacenara los datos
    No retorna nada
*/
void precio (product prod[], int p)
{
    int i,j;
    int temp;
    for(i = 0; i < p; i++)
    {
        for(j=i +1;j < p;j++)
        {
            if(prod[i].precio < prod[j].precio)
            {
                temp = prod[i].precio;
                prod[i].precio = prod[j].precio;
                prod[j].precio = temp;
            }
        }
    }
    printf("Listo\n");
    getch();
}

/*
Funcion letra
Ordenara el inventario de conforme la cantidad, de mayor a menor
Parametros :
    -prod[] = El registro/Estructura
    -p = La posicion en la que se encuentra el programa y en donde
almacenara los datos
    No retorna nada
*/
void cantidad (product prod[], int p)
{
    int i,j;
    int temp;
    for(i = 0; i < p; i++)
    {
        for(j=i +1;j < p;j++)

```

```
{  
    if(prod[i].cantidad < prod[j].cantidad)  
    {  
        temp = prod[i].cantidad;  
        prod[i].cantidad = prod[j].cantidad;  
        prod[j].cantidad = temp;  
    }  
}  
  
}  
printf("Listo\n");  
getch();  
}
```