



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Lenguaje C

Alumno: Vazquez Guzman Jorge Antonio

Matrícula: 372504

Maestro: Yulith Altamirano

Actividad No. : 8

Tema - Unidad : Estructuras

Ensenada Baja California a 22 de noviembre del 2023



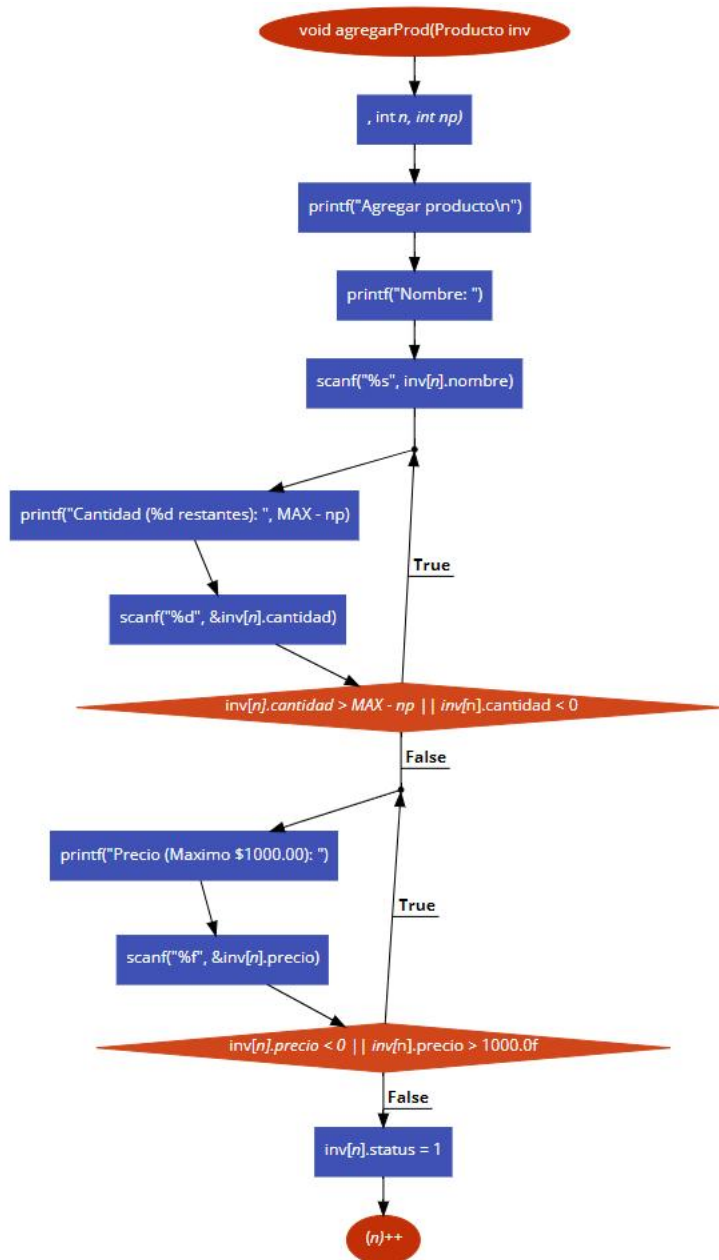
Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Link del repositorio:

<https://github.com/JVPresti/LenguajeC/tree/main/Practica8>

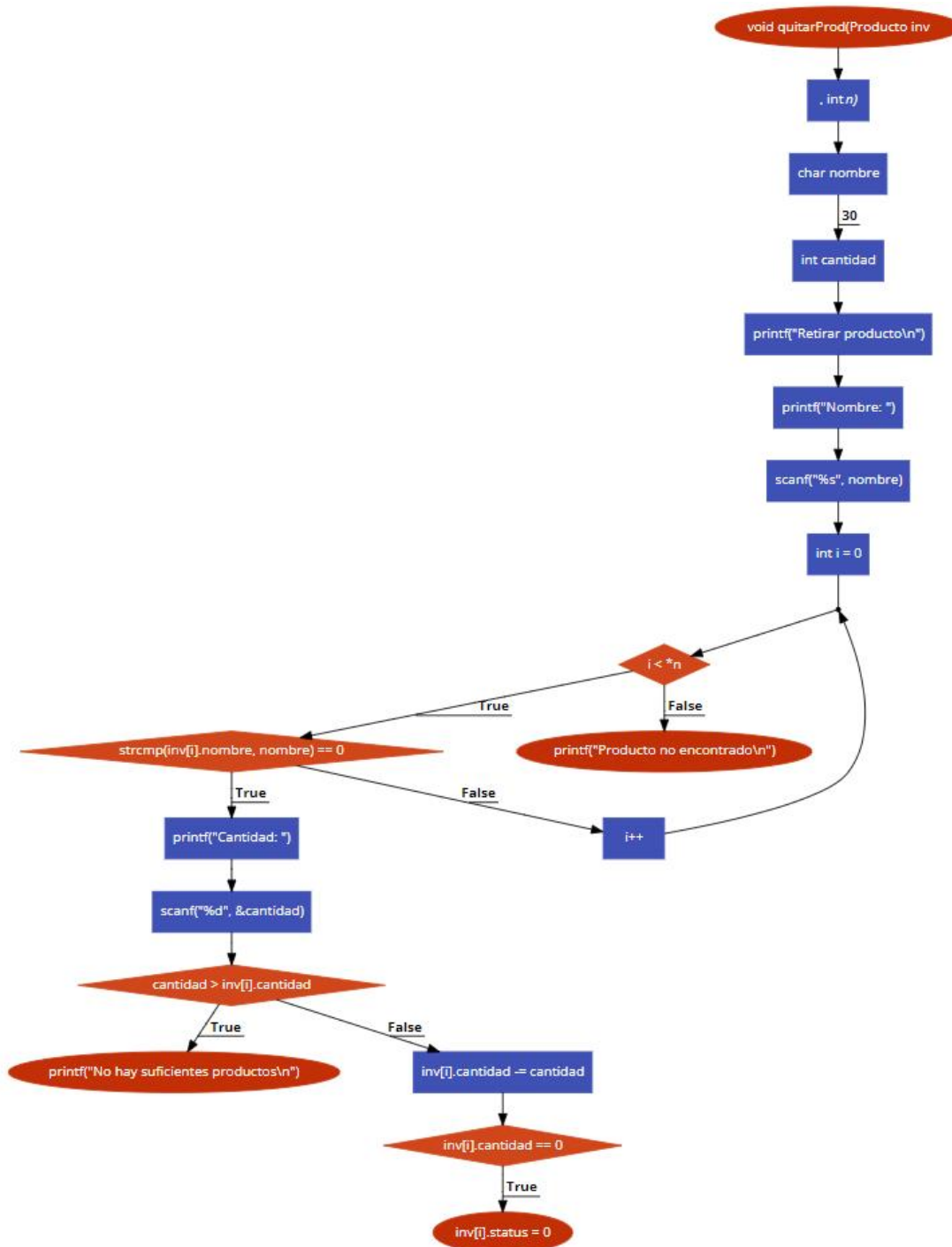
Diagramas de flujo de las funciones:





Universidad Autónoma de Baja California

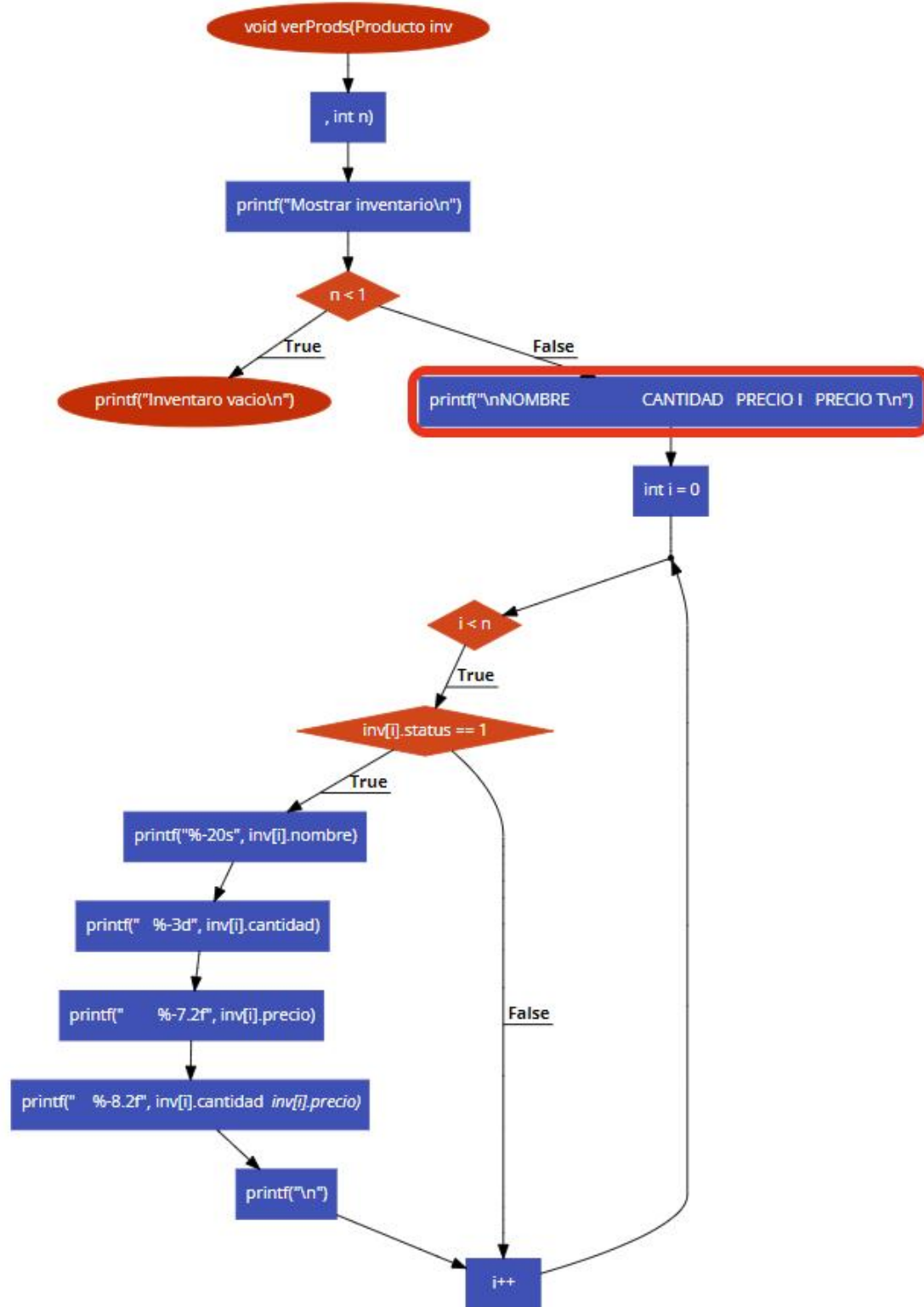
Facultad de Ingeniería Arquitectura y Diseño





Universidad Autónoma de Baja California

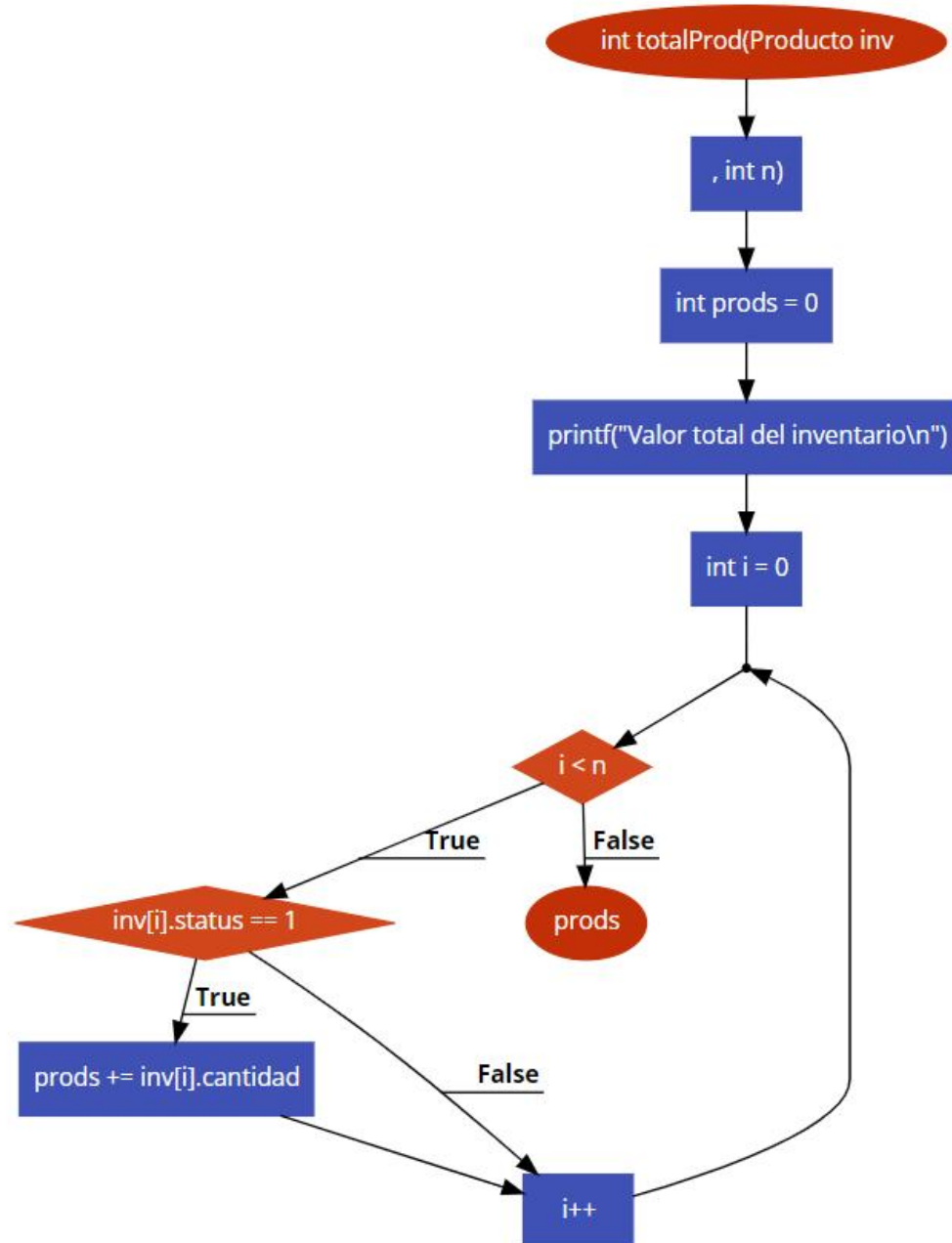
Facultad de Ingeniería Arquitectura y Diseño





Universidad Autónoma de Baja California

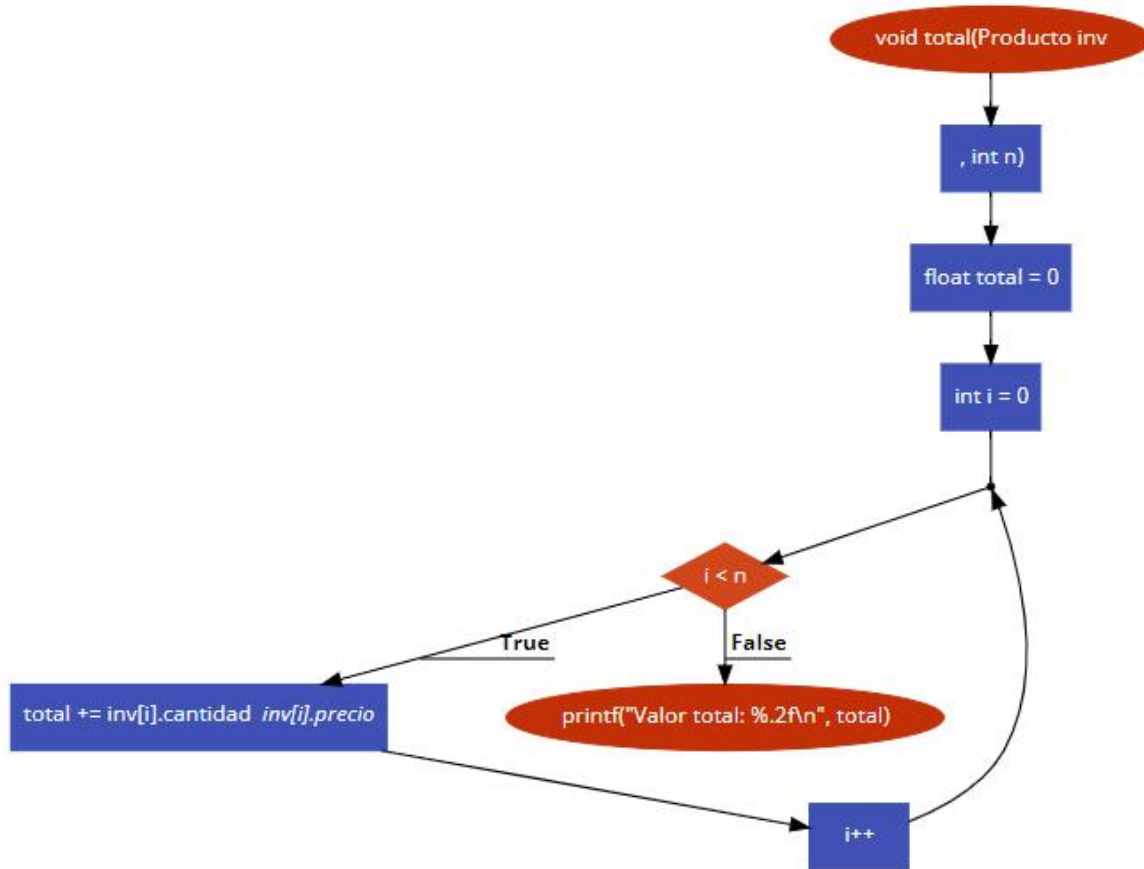
Facultad de Ingeniería Arquitectura y Diseño





Universidad Autónoma de Baja California

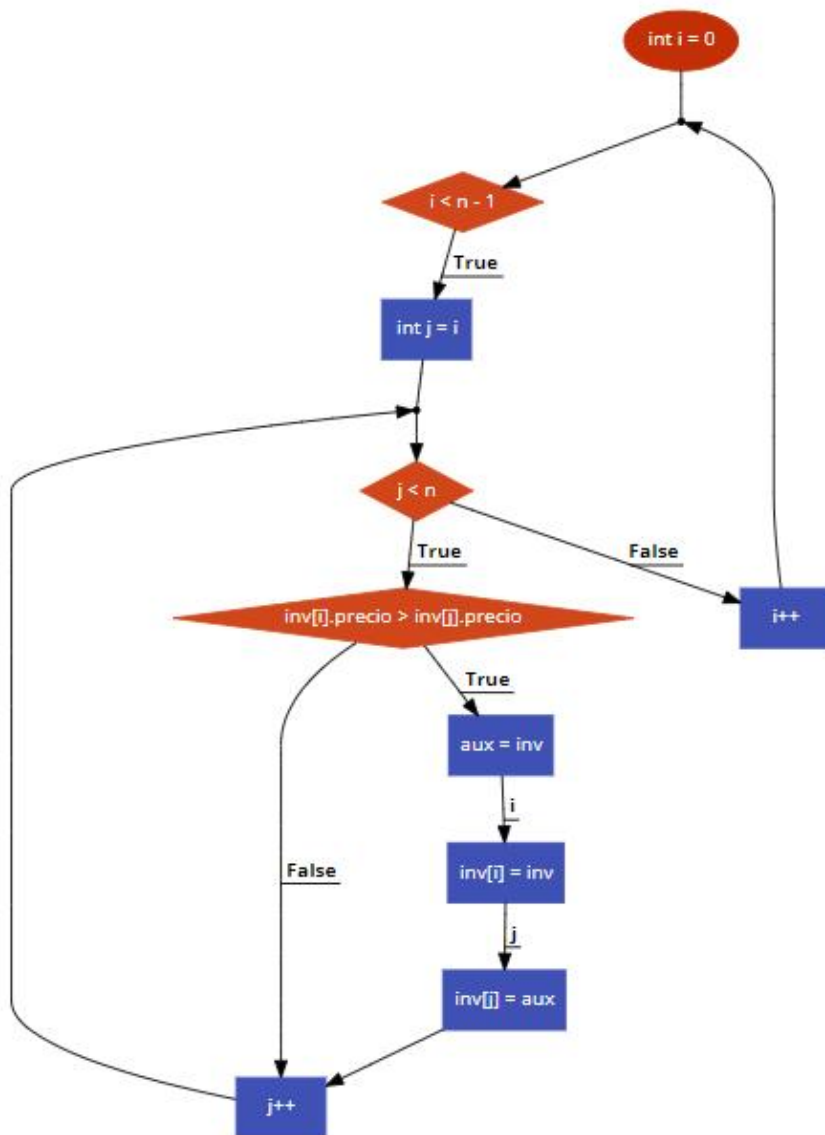
Facultad de Ingeniería Arquitectura y Diseño





Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño





Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Codigo del programa entero:

```
// Practica8_Estructuras._VazquezGuzman_Jorge
// Vazquez Guzman Jorge Antonio
// 22/11/2023
// Este archivo contiene el codigo de la actividad 8, el cual es un inventario
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100
```

```
typedef struct Producto
{
    char nombre[30];
    float precio;
    int cantidad;
    int status;
} Producto;
```

```
// Prototipos de funciones
int totalProd(Producto inv[], int n);
void agregarProd(Producto inv[], int *n, int np);
void quitarProd(Producto inv[], int *n);
void verProds(Producto inv[], int n);
void total(Producto inv[], int n);
void ordenar(Producto inv[], int n);
```

```
int main()
{
    Producto inv[MAX];
    int op, n = 0, np = 0;
```

```
do
{
    system("CLS");
    printf("\t MENU\n");
    printf("1.- Agregar producto\n");
    printf("2.- Retirar producto\n");
    printf("3.- Mostrar inventario\n");
    printf("4.- Valor total\n");
    printf("5.- Ordenar\n");
    printf("6.- Salir\n");
    printf("\nOpcion: ");
    scanf("%d", &op);
    switch (op)
    {
        case 1:
            if (np >= MAX)
            {
                printf("Inventario lleno\n");
                break;
            }

```

```
            agregarProd(inv, &n, np);
            np = totalProd(inv, n);
            break;
        case 2:
            if (np < 1)
            {
                printf("Inventario vacio\n");
                break;
            }
            quitarProd(inv, &n);
            np = totalProd(inv, n);
            break;
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
case 3:
    verProds(inv, n);
    break;
case 4:
    total(inv, n);
    break;
case 5:
    ordenar(inv, n);
    break;
case 6:
    printf("Saliendo...\n");
default:
    printf("Opcion no valida\n");
    break;
}
system("PAUSE");
} while (op != 6);
```

```
return 0;
}
```

```
// Funcion agregarProd
// Esta funcion agrega un producto al inventario
// Como parametros recibe el inventario, el numero de productos y el numero de productos en el inventario
// Regresa el numero de productos en el inventario
void agregarProd(Producto inv[], int *n, int np)
{
    printf("Agregar producto\n");
    printf("Nombre: ");
    scanf("%s", inv[*n].nombre);
```

```
do
{
    printf("Cantidad (%d restantes): ", MAX - np);
    scanf("%d", &inv[*n].cantidad);
} while (inv[*n].cantidad > MAX - np || inv[*n].cantidad < 0);
```

```
do
{
    printf("Precio (Maximo $1000.00): ");
    scanf("%f", &inv[*n].precio);
} while (inv[*n].precio < 0 || inv[*n].precio > 1000.0f);
```

```
inv[*n].status = 1;
(*n)++;
}
```

```
// Funcion quitarProd
// Esta funcion quita un producto del inventario
// Como parametros recibe el inventario y el numero de productos
// Regresa el numero de productos en el inventario
void quitarProd(Producto inv[], int *n)
{
    char nombre[30];
    int cantidad;
    printf("Retirar producto\n");
    printf("Nombre: ");
    scanf("%s", nombre);
```

```
for (int i = 0; i < *n; i++)
{
    if (strcmp(inv[i].nombre, nombre) == 0)
    {
        printf("Cantidad: ");
        scanf("%d", &cantidad);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
if (cantidad > inv[i].cantidad)
{
    printf("No hay suficientes productos\n");
    return;
}
```

```
inv[i].cantidad -= cantidad;
if (inv[i].cantidad == 0)
{
    inv[i].status = 0;
}
```

```
return;
}
```

```
printf("Producto no encontrado\n");
}
```

```
// Funcion verProds
// Esta funcion muestra los productos en el inventario
// Como parametros recibe el inventario y el numero de productos
// Regresa el numero de productos en el inventario
void verProds(Producto inv[], int n)
{
    printf("Mostrar inventario\n");
    if (n < 1)
    {
        printf("Inventario vacio\n");
        return;
    }
}
```

```
printf("\nNOMBRE          CANTIDAD  PRECIO I  PRECIO T\n");
for (int i = 0; i < n; i++)
{
    if (inv[i].status == 1)
    {
        printf("%-20s", inv[i].nombre);
        printf("   %3d", inv[i].cantidad);
        printf("      %-7.2f", inv[i].precio);
        printf("    %-8.2f", inv[i].cantidad * inv[i].precio);
        printf("\n");
    }
}
```

```
// Funcion totalProd
// Esta funcion calcula el numero de productos en el inventario
// Como parametros recibe el inventario y el numero de productos
// Regresa el numero de productos en el inventario
int totalProd(Producto inv[], int n)
{
    int prods = 0;
```

```
printf("Valor total del inventario\n");
for (int i = 0; i < n; i++)
{
    if (inv[i].status == 1)
    {
        prods += inv[i].cantidad;
    }
}
```

```
return prods;
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
// Funcion total
// Esta funcion calcula el valor total del inventario
// Como parametros recibe el inventario y el numero de productos
// Regresa el numero de productos en el inventario
void total(Producto inv[], int n)
{
    float total = 0;
    for (int i = 0; i < n; i++)
    {
        total += inv[i].cantidad * inv[i].precio;
    }
}
```

```
printf("Valor total: %.2f\n", total);
}
```

```
// Funcion ordenar
// Esta funcion ordena el inventario
// Como parametros recibe el inventario y el numero de productos
// Regresa el numero de productos en el inventario
void ordenar(Producto inv[], int n)
{
    int op;
    Producto aux;
```

```
printf("Ordenar\n");
system("pause");
do
{
    system("CLS");
    printf("Ordenar por:\n");
    printf("1.- Precio\n");
    printf("2.- Cantidad\n");
    printf("3.- Nombre\n");
    printf("4.- Salir\n");
```

```
printf("Opcion: ");
scanf("%d", &op);
switch (op)
{
    case 1:
        for (int i = 0; i < n - 1; i++)
        {
            for (int j = i; j < n; j++)
            {
                if (inv[i].precio > inv[j].precio)
                {
                    aux = inv[i];
                    inv[i] = inv[j];
                    inv[j] = aux;
                }
            }
        }
        printf("\nOrdenado por precio\n");
        break;
    case 2:
        for (int i = 0; i < n - 1; i++)
        {
            for (int j = i; j < n; j++)
            {
                if (inv[i].cantidad > inv[j].cantidad)
                {
                    aux = inv[i];
                    inv[i] = inv[j];
                    inv[j] = aux;
                }
            }
        }
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
printf("\nOrdenado por cantidad\n");
break;
case 3:
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i; j < n; j++)
        {
            if (strcmp(inv[i].nombre, inv[j].nombre) > 0)
            {
                aux = inv[i];
                inv[i] = inv[j];
                inv[j] = aux;
            }
        }
    }
    printf("\nOrdenado por nombre\n");
    break;
case 4:
    return 0;
default:
    printf("Opcion no valida\n");
}
} while (op < 1 || op > 4);
return 0;
}
```