

Sistema de Control de Inventario de Depósito

Realizar un sistema de gestión de un depósito de productos mediante el uso exclusivo de matrices bidimensionales.

Cada posición del depósito representa un espacio físico y será modelada utilizando dos matrices:

Una matriz para almacenar los códigos de los productos.

Una matriz para almacenar la cantidad de stock disponible de cada producto.

La solución debe permitir agregar productos, retirar unidades, mover productos de un lugar a otro, buscar productos específicos y realizar reportes estadísticos de inventario.

Diseñar un programa robusto, modularizado, interactivo y validado, utilizando menús y funciones para organizar las operaciones.

Datos a Modelar

Se deben definir y trabajar con las siguientes matrices:

```
#define FILAS 10
```

```
#define COLUMNAS 10
```

```
int codigos[FILAS][COLUMNAS];
```

```
int stock[FILAS][COLUMNAS];
```

Notas importantes:

- Un valor de 0 en `codigos[i][j]` significa que el espacio está vacío.
- El `stock[i][j]` debe ser mayor o igual a cero. Si el stock es 0, se elimina el producto.

Funcionalidades requeridas

Los grupos deben implementar las siguientes funcionalidades mínimas:

1. Inicializar depósito

- Todas las posiciones de ambas matrices deben ser establecidas en cero.
- Deberá existir una opción en el menú para inicializar todo el depósito.

2. Agregar producto

- El sistema debe permitir que el usuario elija una posición libre (fila y columna).
- Se solicita al usuario ingresar:

- Código del producto (debe ser un número positivo distinto de cero).
 - Cantidad de stock inicial (mayor que cero).
- Validar que la posición elegida esté vacía.

3. Retirar stock

- Permitir que el usuario ingrese fila y columna de un producto existente.
- Solicitar la cantidad de unidades a retirar.
- Validar que la cantidad a retirar no supere el stock disponible.
- Si el stock llega a cero luego de la operación, **dar de baja** el producto (poner código a 0).

4. Mover producto

- Permitir trasladar un producto de una celda ocupada a otra celda vacía.
- Copiar tanto el código como el stock.
- Liberar la posición original (poner ambos valores a cero).

5. Buscar producto

- Permitir que el usuario ingrese un **código de producto**.
- Buscarlo en toda la matriz:
 - Informar la ubicación (fila y columna).
 - Informar el stock actual.
- Si el código no existe, notificarlo.

6. Mostrar reportes

Generar un reporte completo que incluya:

- **Mapa del depósito:** mostrar en pantalla la matriz de códigos (visualizando si hay productos o posiciones vacías).
- **Top 3 productos** con mayor stock disponible (ordenando manualmente usando algún algoritmo como burbuja o selección).
- **Cantidad total de espacios ocupados y libres.**

Requisitos

- Modularizar en funciones independientes:
- Una función por operación.
- Menú principal separado en función.

Validar todas las entradas:

- Posiciones dentro de los límites de la matriz.
- Cantidades

Entrega

Mail: mjvidal@ucel.edu.ar

Asunto del correo: Entrega Ejercicio Grupal – <Nombre del programa>

Mensaje:

Integrantes: listado de los integrantes

Adjunto:

Código fuente

Documentación - debe contener una descripción general, nombre de cada función y qué hace. Cómo están organizadas las bibliotecas, que funciones contienen y que responsabilidades contiene.