

READ ME - centroComercial.exe

El siguiente escrito es el manual de usuario para las personas que visiten el ejecutable “centroComercial.exe”, el cual es un programa desarrollado en C, que hace uso de ciclos, ciclos anidados, condicionales, apuntadores, memoria dinámica, enums y demás funciones que en su contenido encontrará.

El programa representa un Centro Comercial, el cual cuenta con 6 funciones esenciales:

- Función crearNuevoLocal (void).
 - Entradas:
 - Doble apuntador de tipo struct definido como “**pMatriz”.
 - Doble apuntador de tipo struct definido como “**pEstado”.
 - Apuntador a una variable de tipo int definido como “*pPiso”.
 - Apuntador a una variable de tipo int definido como “*pLocalesPorPiso”.
 - Salidas: No retorna ningún valor al programa, pero sí imprime diversos datos en la pantalla del sistema.
 - Conceptos usados:
 - printf. Se utiliza para imprimir en pantalla datos que el usuario puede ver y le son de ayuda para guiarse en el programa.
 - scanf. Me permite guardar un dato que el usuario ingresa por consola en una variable local de mi función.
 - fflush(stdin). Limpia el buffer de la consola, evitando así errores al guardar datos con scanf.
 - do{}while{}. Me permite ejecutar una acción reiteradas veces, hasta que cierta condición definida al interior del while se cumpla.
 - Condicionales (if, else). Me permiten tomar diferentes caminos o decisiones dependiendo de condiciones que son evaluadas por el programa.

- Paso de parámetros por referencia. En lugar de generar copias locales de una variable, se crean apuntadores que son utilizados para referirse a estos valores sin necesidad de llamarlos respectivamente.
- Función `mostrarInformacionLocal (void)`.
 - Entradas:
 - Doble apuntador de tipo struct definido como `***pMatriz`.
 - Doble apuntador de tipo struct definido como `***pEstado`.
 - Apuntador a una variable de tipo `int` definido como `*pPiso`.
 - Apuntador a una variable de tipo `int` definido como `*pLocalesPorPiso`.
 - Salidas: No retorna ningún valor al programa, pero sí imprime diversos datos en la pantalla del sistema.
 - Conceptos usados:
 - `printf`. Se utiliza para imprimir en pantalla datos que el usuario puede ver y le son de ayuda para guiarse en el programa.
 - `scanf`. Me permite guardar un dato que el usuario ingresa por consola en una variable local de mi función.
 - `fflush(stdin)`. Limpia el buffer de la consola, evitando así errores al guardar datos con `scanf`.
 - `strcmp`. Se utiliza para comparar dos cadenas de caracteres. En el programa, si ambas cadenas son iguales, `strcmp` retorna el valor 0.
 - Condicionales (`if`, `else`). Me permiten tomar diferentes caminos o decisiones dependiendo de condiciones que son evaluadas por el programa.
 - Paso de parámetros por referencia. En lugar de generar copias locales de una variable, se crean apuntadores que son utilizados para referirse a estos valores sin necesidad de llamarlos respectivamente.
- Función `modificarLocal (void)`.
 - Entradas:
 - Doble apuntador de tipo struct definido como `***pMatriz`.

- Doble apuntador de tipo struct definido como “**pEstado”.
 - Apuntador a una variable de tipo int definido como “*pPiso”.
 - Apuntador a una variable de tipo int definido como “*pLocalesPorPiso”.
- Salidas: No retorna ningún valor al programa, pero sí imprime diversos datos en la pantalla del sistema.
- Conceptos usados:
 - printf. Se utiliza para imprimir en pantalla datos que el usuario puede ver y le son de ayuda para guiarse en el programa.
 - scanf. Me permite guardar un dato que el usuario ingresa por consola en una variable local de mi función.
 - fflush(stdin). Limpia el buffer de la consola, evitando así errores al guardar datos con scanf.
 - strcmp. Se utiliza para comparar dos cadenas de caracteres. En el programa, si ambas cadenas son iguales, strcmp retorna el valor 0.
 - Condicionales (if, else). Me permiten tomar diferentes caminos o decisiones dependiendo de condiciones que son evaluadas por el programa.
 - Paso de parámetros por referencia. En lugar de generar copias locales de una variable, se crean apuntadores que son utilizados para referirse a estos valores sin necesidad de llamarlos respectivamente.
- Función agregarVisitas (void).
 - Entradas:
 - Doble apuntador de tipo struct definido como “**pMatriz”.
 - Doble apuntador de tipo struct definido como “**pEstado”.
 - Apuntador a una variable de tipo int definido como “*pPiso”.
 - Apuntador a una variable de tipo int definido como “*pLocalesPorPiso”.
 - Salidas: No retorna ningún valor al programa, pero sí imprime diversos datos en la pantalla del sistema.
 - Conceptos usados:

- printf. Se utiliza para imprimir en pantalla datos que el usuario puede ver y le son de ayuda para guiarse en el programa.
 - scanf. Me permite guardar un dato que el usuario ingresa por consola en una variable local de mi función.
 - fflush(stdin). Limpia el buffer de la consola, evitando así errores al guardar datos con scanf.
 - strcmp. Se utiliza para comparar dos cadenas de caracteres. En el programa, si ambas cadenas son iguales, strcmp retorna el valor 0.
 - Condicionales (if, else). Me permiten tomar diferentes caminos o decisiones dependiendo de condiciones que son evaluadas por el programa.
 - Paso de parámetros por referencia. En lugar de generar copias locales de una variable, se crean apuntadores que son utilizados para referirse a estos valores sin necesidad de llamarlos respectivamente.
 - Recursión. Llamado reiterado de una función a ella misma dentro de su procedimiento.
- Función recomendarLocales (void).
 - Entradas:
 - Doble apuntador de tipo struct definido como “**pMatriz”.
 - Doble apuntador de tipo struct definido como “**pEstado”.
 - Apuntador a una variable de tipo int definido como “*pPiso”.
 - Apuntador a una variable de tipo int definido como “*pLocalesPorPiso”.
 - Salidas: No retorna ningún valor al programa, pero sí imprime diversos datos en la pantalla del sistema.
 - Conceptos usados:
 - printf. Se utiliza para imprimir en pantalla datos que el usuario puede ver y le son de ayuda para guiarse en el programa.
 - scanf. Me permite guardar un dato que el usuario ingresa por consola en una variable local de mi función.

- fflush(stdin). Limpia el buffer de la consola, evitando así errores al guardar datos con scanf.
 - strcmp. Se utiliza para comparar dos cadenas de caracteres. En el programa, si ambas cadenas son iguales, strcmp retorna el valor 0.
 - Condicionales (if, else). Me permiten tomar diferentes caminos o decisiones dependiendo de condiciones que son evaluadas por el programa.
 - Paso de parámetros por referencia. En lugar de generar copias locales de una variable, se crean apuntadores que son utilizados para referirse a estos valores sin necesidad de llamarlos respectivamente.
- Función eliminarLocal (void).
 - Entradas:
 - Doble apuntador de tipo struct definido como “**pMatriz”.
 - Doble apuntador de tipo struct definido como “**pEstado”.
 - Apuntador a una variable de tipo int definido como “*pPiso”.
 - Apuntador a una variable de tipo int definido como “*pLocalesPorPiso”.
 - Salidas: No retorna ningún valor al programa, pero sí imprime diversos datos en la pantalla del sistema.
 - Conceptos usados:
 - printf. Se utiliza para imprimir en pantalla datos que el usuario puede ver y le son de ayuda para guiarse en el programa.
 - scanf. Me permite guardar un dato que el usuario ingresa por consola en una variable local de mi función.
 - fflush(stdin). Limpia el buffer de la consola, evitando así errores al guardar datos con scanf.
 - strcmp. Se utiliza para comparar dos cadenas de caracteres. En el programa, si ambas cadenas son iguales, strcmp retorna el valor 0.

- Condicionales (if, else). Me permiten tomar diferentes caminos o decisiones dependiendo de condiciones que son evaluadas por el programa.
- Paso de parámetros por referencia. En lugar de generar copias locales de una variable, se crean apuntadores que son utilizados para referirse a estos valores sin necesidad de llamarlos respectivamente.

El programa posee un manejo intuitivo, lo que permite al usuario gran libertad en su uso. Además, el programa cuenta con avisos en caso de que el usuario ingrese un dato o comando erróneo, y acompañada de esta advertencia, se muestran también instrucciones paso a paso de cómo desarrollar las funciones.

Por favor, lo invitamos a que abra el programa y experimente un poco con él. ¡Muchas Gracias!