**Funciones**

*Nota: Todas las funciones excepto por* menú *son de tipo void. Por ello, la “salida” descrita en cada función describe los cambios hace la función dentro del programa, no el valor que retorna.*

**Función *inicializarCC*:** Inicializa la matriz 10x5 del centro comercial, asignando la disponibilidad de cada celda a *disp*, para poder ubicar los locales.

* Entradas: La matriz 10x5 **centroC**.
* Salidas: Settea todas las celdas de la matriz centroC a disponibles para poder agregar locales. Adicionalmente asigna el adecuado pisoLocal y numLocalxPiso para cada celda.
* Conceptos usados: For anidado, matrices.

**Función *menú:*** Display del menu principal que verá el usuario.

* Entradas: Ninguna
* Salidas: Print de todo el menu, legible para el usuario del programa.
* Conceptos usados: Print, scan.

**Función *agregarLocal*:** Pide al usuario la ubicacion y el nombre de un nuevo local para meterlo a la matriz.

* Entradas: La matriz 10x5 **centroC**.
* Salidas: Pide el piso y el numero de local al usuario. Verifica que este dentro del rango de la matriz, verifica que la celda mencionada esté disponible para asignarle un nuevo local, luego pide el nombre al usuario y lo asigna en su respectivo lugar en la matriz centroC
* Conceptos usados: Condicionales, operadores lógicos. Recursión (la función se llama a sí misma las veces que necesite mientras el usuario no ingrese un piso y/o número de local válido).

**Función dispLine:** Funcion usada por la funcion dispLocales para recursivamente hacer el display visual de la matriz.

* Entradas: La matriz 10x5 **centroC** y dos enteros, col y fil, para hacer la recursión.
* Salidas: Imprime “visualmente” un piso del centro comercial, poniendo una “O” en celdas disponibles y una “X” donde ya hay un local.
* Conceptos usados: Recursión, enums

**Función *dispLocales*:** Utiliza la función dispLine para imprimir cada columna de la matriz centroC.

* Entradas: La matriz 10x5 **centroC**.
* Salidas: Imprime todos los “pisos” del centro comercial
* Conceptos usados: Recursión

**Funcion *eliminarLocal*:** Pide al usuario la ubicación del local a borrar.

* Entradas: La matriz 10x5 **centroC**.
* Salidas: Settea la disponibilidad de dicha celda a “disp” para que se tome como vacía la celda y se pueda asignar un nuevo local en ese lugar.
* Conceptos usados: Condicionales, enums.

**Funcion *cambiarNombreLocal*:** Pide un nombre de un local existente para cambiarlo

* Entradas: La matriz 10x5 **centroC**.
* Salidas: Verifica que el local especificado exista dentro de la matriz. Donde sea asi, pide el nuevo nombre del local y directamente desde el scanf, lo asigna al nombre del local.
* Conceptos usados: For anidado, comparacion de strings (strcmp).

**ENTREGA 2**

Mejoras:

* *Se creó la matriz usando memoria dinámica (malloc) a diferencia de la primera entrega*
  + Con esto se modificaron todas las funciones originales para interactuar con la matriz de memoria dinámica
* *Se agregaron características (fields) adicionales al struct:*
  + Metros2: Tamaño del local
  + Rating: Calificación de los usuarios de 1 a 5 del local
  + Empleados: Cantidad de empleados que trabajan en el local. Por el coronavirus, se requiere que cada persona tenga almenos 4 metros cuadrados de espacio personal, lo cual se tiene en cuenta en este programa
  + Added: Un field único para cada local, el cual rastrear el orden en el cual los locales fueron agregados al centro comercial
* *Se agregaron funciones al programa:*
  + Rate
    - Entradas: la matriz de locales y las dimensiones de la misma.
    - Salidas: con input del usuario, modifica el rating (un field del struct) el cual promedia la experiencia de los clientes de dicho local.
    - Conceptos usados: ciclos, condicionales, array dentro de struct, conversión de tipos de dato.
  + orderByRating:
    - Entradas: la matriz de locales y las dimensiones de la misma.
    - Salidas: Usando selection sort, imprime los locales existentes en orden de rating (de menor a mayor)
    - Conceptos usados: Manejo de archivos, ciclos anidados, sorteo iterativo (SS)
  + orderByArrival:
    - Entradas: la matriz de locales y las dimensiones de la misma.
    - Salidas: Usando insertion sort, imprime los locales existentes en orden de llegada al centro comercial.
    - Conceptos usados: Manejo de archivos, ciclos anidados, sorteo iterativo (IS)
  + orderBySize:
    - Entradas: la matriz de locales y las dimensiones de la misma.
    - Salidas: Usando merge sort, imprime los locales existentes por el tamaño del local (de menor a mayor)
    - Conceptos usados: Manejo de archivos, ciclos anidados, sorteo recursivo
* *Funcionalidades de guardado:*
  + Save:
    - Entradas: la matriz de locales y las dimensiones de la misma, el nombre del archivo donde se guardará la matriz.
    - Salidas: El programa guarda los locales ocupados de la matriz dentro de un archivo binario para poder cargarlos al comienzo de otra ejecución. Tambien guarda las dimensiones de dicha matriz para asi cargarla, además del número total de locales.
    - Conceptos usados: Manejo de archivos, condicionales, manejo de apuntadores
  + Load:
    - Entradas: la matriz de locales y las dimensiones de la misma, el nombre del archivo donde se guardará la matriz.
    - Salidas: El programa al comienzo le da la opción al usuario de cargar una matriz vieja o comenzar con una nueva. En ambos casos inicializa una matriz vacía, pero al cargar una matriz vieja, usa la información previa de las dimensiones para crearla y luego inserta en sus respectivos lugares a los locales pre-existentes.
    - Conceptos usados: Manejo de archivos, condicionales, apuntadores, lectura de archivos.