

Instrucciones

- Lea con detenimiento y desarrolle **individualmente** cada una de las actividades a realizar durante la experiencia.
- Cree un archivo con compresor **.rar** o **.zip** con lo desarrollado. El nombre del archivo debe tener el siguiente formato: **TEL102_C1_Nombre_Apellido.rar** sin incluir tildes. Este archivo debe incluir una imagen con formato **.png** o **.jpg** y un código **.cpp** siguiendo la misma nomenclatura: **TEL102_C1_Nombre_Apellido.png** y **TEL102_C1_Nombre_Apellido.cpp**. (Ej. TEL102_C1_Nicolas_Galvez.rar).
- Enviar el archivo a través de la página de aula del ramo, sección “Control 2” hasta las 11:45:00 del día de **hoy**, Martes 06/10/2020, Hora continental de Chile (UTC-3).
- Cada minuto de atraso tendrá un descuento siguiendo la serie de Fibonacci.
- Trate de utilizar herramientas conocidas o aprendidas en clases. **No copie literalmente de recursos online.**
- Sea riguroso con las instrucciones de desarrollo.

MC++

En **PlusPlus City**, la cadena de restaurantes de comida rápida **MC++**, está preparándose para abrir de forma presencial durante la Pandemia. Por requerimientos del Ministerio de Sanidad, solo podrá contar con cinco (5) mesas disponibles en cada una de sus sucursales.

MC++ lo contrata a ud. para poder ayudar a automatizar la asignación de mesas a los clientes. Para eso le entrega una porción de código, ya implementado, llamado **mesa.cpp**.

En este código, que inicializa las cinco mesas disponibles, existe una estructura llamada **mesa** que contiene dos atributos: **int personas** que indica la cantidad de personas que utilizan la mesa e **int used** que indica si la mesa está disponible (0) o usada (1). El resto es deducible al leer el código.

mesas.cpp

```
#include<iostream>

const int M=5;

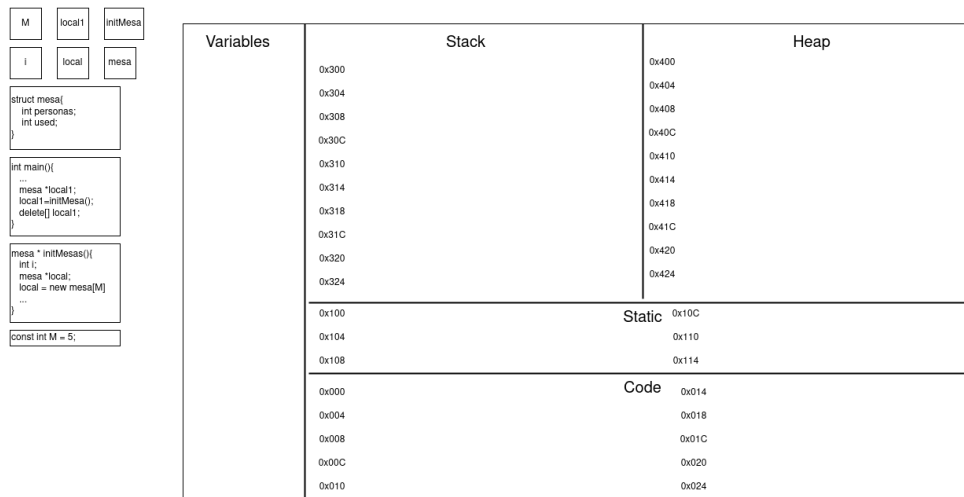
struct mesa{
    int personas;
    int used;
};

mesa * initMesas(){
    int i;
    mesa *local;
    local = new mesa[M];
    for(i=0;i<M;i++){
        local[i].personas=0;
        local[i].used=0;
    }
    return local;
}

int main(){
    mesa *local1;
    local1 = initMesas(); //dibujar este estado.
    delete[] local1; //liberar memoria
    return 0;
}
```

Su nuevo empleador le solicita:

1. Dibujar la asignación de memoria: código, estática, stack y heap; justo despues de ejecutar la linea `local1=initMesas();`. Use como base el siguiente esquema de memoria:



Tendrá dos opciones para realizar esto:

1. Podrá utilizar el archivo `memory-scheme.drawio`, disponible en AULA USM, y editarlo digitalmente en la plataforma <https://draw.io>.
2. Podrá dibujar a mano el esquema de memoria y digitalizarlo a través de una foto.

Deberá diferenciar claramente entre elementos que están siendo utilizados en memoria de elementos utilizados pero ya liberados. Para el primer tipo, debe usar **bordes y flechas continuas** y, para el segundo tipo, **bordes y flechas segmentadas**.

2. Implemente la función `mesa * checkMesa(mesa *local)`, la cual recibe como parámetro un arreglo dinámico del tipo `mesa` llamado `local` y debe retornar un puntero del tipo `mesa` a la primera posición en el arreglo `local` que represente a una mesa que **no está siendo usada**. En caso de no existir mesas sin uso, se debe retornar un puntero dirigido a `NULL`. Un ejemplo de salida, se muestra a continuación:

Salida (consola)

```
[elprofe@tel102 control2]$ ./MESAS
Mesas inicializadas
Mesa 0. Personas: 0. Usada: 0
Mesa 1. Personas: 0. Usada: 0
Mesa 2. Personas: 0. Usada: 0
Mesa 3. Personas: 0. Usada: 0
Mesa 4. Personas: 0. Usada: 0

Mesa disponible: 0x556223455eb0
```

3. Cree la función `void assignMesa(mesa *table)`, la cual recibe como parámetro un puntero a una `mesa` llamada `table` y no retorna valores. La función debe asignar valores a la mesa que representa la variable `table`. El número de personas debe ser solicitado por entrada estándar. Además, debe actualizar el atributo que indica que la mesa está siendo usada. Un ejemplo de salida, se muestra a continuación:

Salida (consola)

```
[elprofe@tel102 control2]$ ./MESAS
Mesas inicializadas
Mesa 0. Personas: 0. Usada: 0
Mesa 1. Personas: 0. Usada: 0
Mesa 2. Personas: 0. Usada: 0
Mesa 3. Personas: 0. Usada: 0
Mesa 4. Personas: 0. Usada: 0

Mesa disponible: 0x556223455eb0

Ingrese cantidad de personas: 5

Asignando Mesa 0x55ef77093eb0

Mesa 0. Personas: 5. Usada: 1
Mesa 1. Personas: 0. Usada: 0
Mesa 2. Personas: 0. Usada: 0
Mesa 3. Personas: 0. Usada: 0
Mesa 4. Personas: 0. Usada: 0
```