**PROYECTO (individual)**

1. **Condiciones generales**

* **Fecha de entrega, lunes 7 de diciembre de 2020 a las 23:59**
* use ***lowerCamelCase.***
* Para la entrega, suba al Bb en un archivo comprimido con su nombre completo, los archivos (.c, .exe, y demás) que tienen la solución del proyecto.
* Haga uso de separación de archivos, cree un makefile y los archivos .h que considere necesarios.
* En alguna parte de su programa deberá hacer uso de enums
* Deberá hacer uso de structs.
* Su programa debe tener un menú con las opciones que operan el centro comercial y el parqueadero.
* Cree las funciones y procedimientos que considere necesarios para desarrollar los diferentes puntos del proyecto. Crear toda la solución de nuestro programa directamente en el main() no es una buena práctica de programación. Desacoplar las funcionalidades de sus programas no sólo mejoran la estética, también mejora el mantenimiento del código.
* La sustentación del proyecto la debe realizar en un dispositivo donde usted mismo pueda modificar, compilar y ejecutar su código. El uso de la cámara web es necesaria al momento de la sustentación

1. **EL CENTRO COMERCIAL**
2. Pida al usuario la cantidad de pisos del centro comercial (filas) y la cantidad de locales que tiene en cada piso (columnas).
3. Cada local del centro comercial deberá ser visto como un struct, los elementos mínimos que debe tener el local son los siguientes (nombre, código del local, piso en que se encuentra el local, estado del local, costo arrendamiento).
4. Los locales del centro comercial deberán manejar un estado (libre o ocupado) para conocer el estado de ocupación del centro comercial y gestionar el alquiler
5. Alquilar un local: el usuario podrá elegir el piso en que estará ubicado su local, de manera que usted asignará uno de los locales del piso (siempre y cuando esté libre) y el costo sea inferior a la capacidad de pago que tiene el cliente que alquilará el local.
6. Ordenar un piso: el usuario necesita reorganizar uno de los pisos con relación al costo del arrendamiento, de esa manera será más justo ya que los primeros locales son de mayor tamaño y mejor resultado de ventas. El usuario le indicará cual de los pisos desea ordenar, y usted deberá ordenarlo en orden descendente (por el costo) utilizando el algoritmo iterativo (selección o inserción, el que usted decida).
7. Cree un reporte que permita indicar el estado de ocupación del centro comercial, mostrando cuantos locales quedan disponibles en cada uno de los pisos.
8. Implemente una operación que permita modificar la información de uno o más locales, lo que usted prefiera.
9. Implemente una operación para desocupar uno o más locales, lo que usted prefiera. Desocupar un local hace referencia a cambiar el estado de ocupado a libre.
10. **EL PARQUEADERO DEL CENTRO COMERCIAL**
11. Cree un arreglo que almacenará parqueaderos (tantos como el usuario indique), este arreglo corresponderá a los lugares donde los clientes podrán dejar sus vehículos al visitar el centro comercial (el arreglo lo debe manejar con memoria dinámica ya que el centro comercial está con déficit de parqueaderos y es posible que deba incrementar puestos).
12. Cada parqueadero debe tener como mínimo los siguientes elementos (id del parqueadero, tipo {carro o moto}, estado {libre, ocupado})
13. Ingresar un vehículo, usted deberá hacer el proceso de ingresar un vehículo al parqueadero siempre y cuando haya disponibilidad (haya puesto libre) y coincida el vehículo con el puesto libre (carro o moto), luego de haber parqueado el vehículo se le debe informar al usuario en qué puesto quedó parqueado.
14. Cuando se parquee un vehículo se debe observar el porcentaje de ocupación, de manera que, si se excede el 80% de ocupación, se le advierta al usuario que se está llegando al límite del parqueadero.
15. Ampliar la capacidad del parqueadero: El centro comercial está terminando de remodelar el lugar, de manera que habrá unos nuevos espacios que se usarán para ampliar el parqueadero. El usuario le indicará cuantos espacios nuevos hay, y el programaba debe ampliarse para que pueda soportar el crecimiento en puestos del parqueadero.
16. Mostrar estado del parqueadero: el usuario podrá con esta opción ver cuáles parqueaderos están ocupados y cuáles están disponibles (ojo, dice cuáles no cuántos).
17. Si considera que necesita más funcionalidades para que su programa opere bien, puede crearlas.
18. **COMPONENTE DE CONSULTA**
19. Para realizar este punto usted deberá consultar sobre el manejo de archivos en lenguaje C, recomiendo que haga uso de archivos serializados o binarios ya que pueden facilitar su uso. El manejo de los archivos serializados tiene una gran similitud a cómo lo aprendieron en lenguaje Python.

* Usted deberá darle la opción al usuario de guardar la información del centro comercial, es decir, cuando el usuario utilice esta opción del menú, la información del centro comercial (sin incluir el parqueadero) quedará guardada en un archivo para poderla restablecer cuando sea necesario.
* Usted deberá, al momento de abrir el programa, preguntar al usuario si desea empezar un centro comercial desde cero o si desea cargar el centro comercial guardado, en caso de que el usuario decida cargarlo, usted deberá cargar la información desde el archivo serializado que realizó en el punto anterior.

**Rubrica de propiedad intelectual para la sustentación**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0.8 | 0.6 | 0.4 | 0.2 | 0 |
| **Sustentación** | Es evidente que el estudiante entiende el código que desarrolló lo explica con claridad y responde correctamente a las preguntas. | La sustentación es buena, pero se evidenció inseguridad del estudiante para explicar algunas partes del trabajo desarrollado o para responder algunas preguntas. | La sustentación es aceptable se evidencia que el estudiante desarrolló el código, pero le cuesta trabajo explicar aspectos del código. | La sustentación no es buena, hay inseguridad del estudiante para explicar gran parte del trabajo desarrollado o para responder muchas de las preguntas. Parece que el código no hubiera sido desarrollado por el estudiante. | El estudiante demuestra que entiende partes del código, pero no tiene claro cómo se relacionan con la funcionalidad solicitada. | Se evidencia que el estudiante no entiende el código desarrollado, no logra responder correctamente a las preguntas formuladas. |