El departamento de Bienestar Estudiantil de la ESPOL ha implementado un nuevo servicio de rentas de bicicletas para los estudiantes que desean movilizarse internamente en el campus universitario.

El departamento de Bienestar Estudiantil está interesado en conocer la acogida del servicio por parte de los estudiantes, para ello cuenta con un archivo de datos en texto plano, el cual tiene la información de todas las rentas de bicicletas que se han dado hasta el momento. El archivo tiene el siguiente formato:

Matricula|Nombre|# de bicicleta|Hora Inicio|Hora Fin|Multa

A continuación se ofrece un ejemplo del contenido del archivo:

200501010|JUAN PEREZ|10|15|16|2.0  
200502020|LUIS CASTRO|24|14|15|3.50  
200502020|LUIS CASTRO|30|11|12|0.0  
200501010|JUAN PEREZ|9|9|10|1.50

Implemente la función **almacenarRentasEstudiantes(nombreArchivo, prefijoSalida, horaInicio, horaFinal)**, la cual recibe el nombre del archivo donde se encuentra la información de los alquileres y un rango de horas*. Para el rango de horas especificadas*, la función calcula las veces que cada estudiante ha prestado una bicicleta, la cantidad y monto total de multas y almacena los resultados de cada estudiante en archivos individuales asociados con cada uno de ellos. Los archivos tienen de salida por nombre:

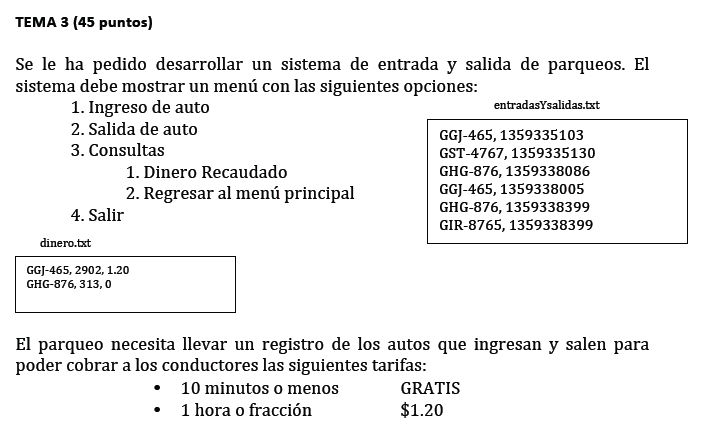
*prefijoSalida\_numeroMatricula.txt*

Ejemplo de la llamada a la funcion:

**almacenarRentasEstudiantes(“rentas.txt”, “reporte”, 11, 16)**

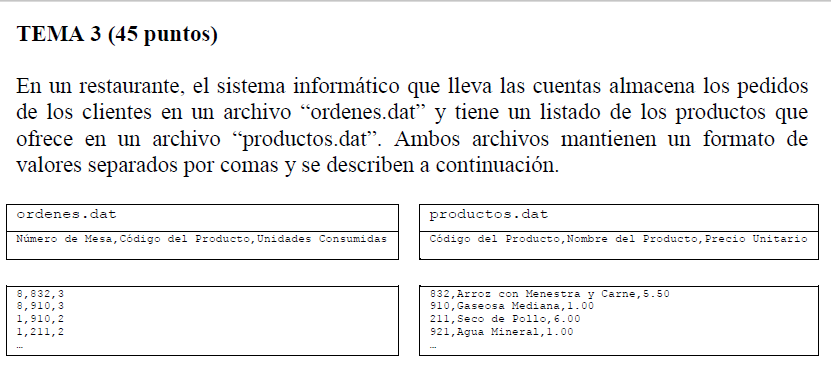
reporte\_200501010.txt  
1|1|2.0

reporte\_200502020.txt  
2|1|3.50



Implemente:

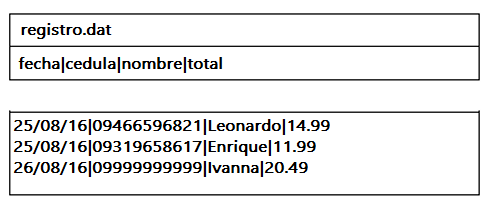
1. La función **ingresoSalida(filename, placa, segundos)** que recibe el nombre del archivo “entradasYsalidas.txt”, el numero de placa del auto que ingresa/sale y la hora de ingreso/salida que es la hora actual del calendario en segundos desde ene 1, 1970. Esta funcion escribe en el archivo la informacion de entrada y salida.
2. La función **pago(filename\_eys, placa, filename\_dinero)** que recibe el nombre de los dos archivos y el numero de placa del auto. La función calcula y retorna el valor que debe pagar el conductor del auto por el tiempo que ha estado en el parqueo. También escribe en el archivo “dinero.txt” la placa del vehículo, el tiempo que estuvo parqueado en segundos, y el valor cobrado.
3. El programa principal que muestre el menú del sistema. Permita al usuario registrar el ingreso de un vehículo, y al registrar la salida cobrar la tarifa. Mostrar el total de dinero recaudado. Use los procedimientos y funciones implementados en los literales anteriores.

Implementar:

La función **facturar(mesa, file\_ordenes, file\_productos)**

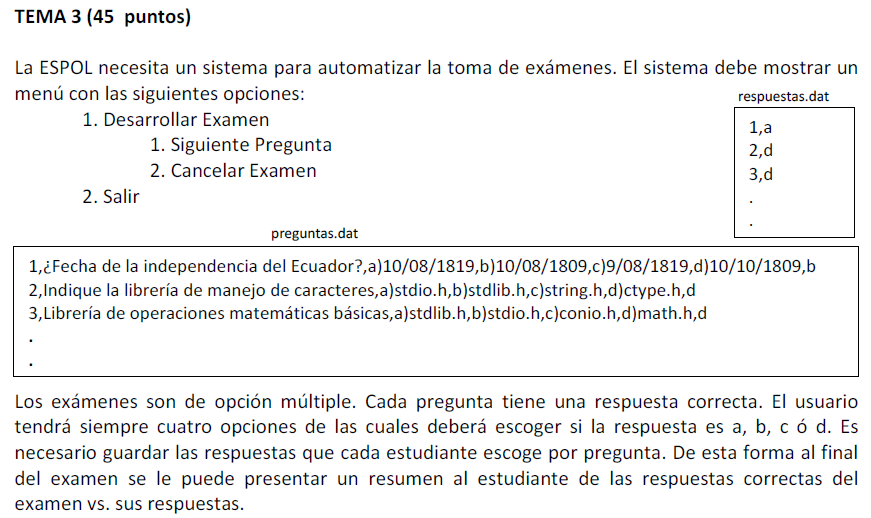
* Recibe: el número de mesa del cliente que desea pagar su orden y los nombres de los archivos ordenes.dat y productos.dat
* La función debe mostrar por pantalla todos los productos consumidos por dicha mesa, indicando: cantidad, nombre del producto, precio unitario y el subtotal por producto (precio unitario x cantidad).
* Al finalizar se debe retornar el total consumido para que pueda ser utilizado por el sistema.

La función **guardarOrden(filename, cedCliente, nombreCliente, fecha, total)**, que recibe el nombre al archivo “registro.dat”, la cedula del cliente, el nombre del cliente, la fecha en que se realizo la compra, y el valor total que cancelo el cliente. Esta función escribe en el archivo “registro.dat” la orden del cliente.



Un programa principal que:

1. Solicite los datos del cliente (nombre, cedula, dirección y teléfono).
2. Pregunte al usuario la mesa correspondiente.
3. Imprima la cabecera de la factura (nombre, cedula, dirección, teléfono y fecha). Asuma que tiene una función llamada imprimirFecha() que imprime por pantalla la fecha actual.
4. Llame a la función facturar que se encargara de imprimir el cuerpo de la factura y calculara el total consumido a utilizarse en el siguiente punto.
5. Calcule los valores que debe añadir por servicio (10%), y por IVA (14% del subtotal de la orden incluido servicio). Muestre estos valores y el total a pagar.



A partir del archivo preguntas.dat implemente:

**1)** La funcion **guardarRespuestas(FILE, num, respuesta)**, que recibe el nombre del

archivo “respuestas.dat”, el número de la pregunta que el usuario ha respondido y su respuesta.

**2)** La función **respuestaPregunta(num, respuesta, FILE)**, que recibe el número de la

pregunta, la respuesta del usuario y el nombre del archivo “preguntas.dat” para comprobar su

validez. El archivo contiene el número de pregunta, pregunta, opción a), opción b), opción c),

opción d), y respuesta correcta. La función retorna 1 si la pregunta fue respondida de forma

correcta, de lo contrario retorna 0.

**3)** El programa principal, que muestre el menú del sistema. El examen termina cuando el usuario

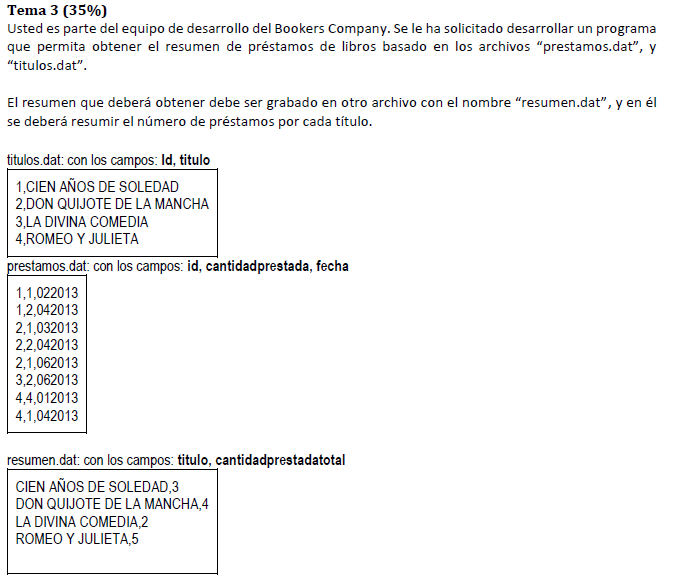
ha respondido a todas las preguntas o escoge la opción Cancelar Examen. Luego de esto se

muestra el número de respuestas correctas, el número de respuestas incorrectas y un resumen de

las respuestas correctas del examen vs. sus respuestas. Luego de esto, el programa deberá

regresar al menú principal. Use los procedimientos y funciones implementados en los literales

anteriores.



Se le solicita implementar en **python** lo siguiente:

1. La funcion **escribirResumen(ftitulos, fprestamos, fresumen)** que recibe los nombres de los archivos de títulos, de préstamos, y el nuevo archivo a ser generado. Esta función debe generar el archivo resumen como se explica anteriormente.