

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
FUNDAMENTOS DE PROGRAMACIÓN - SEGUNDA EVALUACIÓN
II TÉRMINO 2012-2013

Nombre: _____ **Matrícula:** _____

TEMA 1 (25 puntos)

La **raíz digital** de un entero n se define como el resultado de la suma de sus dígitos repetidamente hasta que quede un solo dígito. Por ejemplo, la raíz digital de 1729 puede ser calculada utilizando los siguientes pasos:

$$\text{Paso 1: } 1 + 7 + 2 + 9 = 19$$

$$\text{Paso 2: } 1 + 9 = 10$$

$$\text{Paso 3: } 1 + 0 = 1$$

En vista que el total de la suma al final del paso 3 es el dígito 1, este es la raíz digital de 1729.

Usted deberá implementar:

1) La función recursiva **int sumarDigitos(int num)** que toma un entero no negativo y retorna la suma de sus dígitos. Por ejemplo, la llamada a `sumarDigitos(1729)` debe retornar $1+7+2+9$ que es 19.

2) La función recursiva **int obtenerRaizDigital(int num)** que toma un entero no negativo y retorna su raíz digital. Use la función `sumarDigitos`.

TEMA 2 (30 puntos)

Run-Length-Encoding es uno de los algoritmos más sencillos de codificación y compresión. Es especialmente útil para comprimir imágenes (no fotos), donde los colores se repiten uno tras otros por largos rangos.

Considere la siguiente línea, donde un pixel negro es representado por el caracter B y un pixel blanco es representado por W:

WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWWWWWB

Al comprimirlo con RLE se cuenta la cantidad de repeticiones consecutivas y se lo anota. Se prosigue a contar el siguiente set de repeticiones. El resultado es la siguiente cadena comprimida.

12,W,1,B,12,W,3,B,24,W,1,B

- a) Escriba la función **void comprimirRLE(FILE *original, FILE *rle)**, que recibe un archivo sin compresión y lo comprime hacia RLE. Considere que la longitud de la cadena original puede ser muy larga..
- b) Escriba un programa que:
 1. Pida el nombre de un archivo y verifique si este existe.
 2. Añada la extensión ".rle" al nombre de archivo original para generar el nombre del comprimido.
 3. Comprima el archivo.

TEMA 3 (45 puntos)

Se le ha pedido desarrollar un sistema de entrada y salida de parqueos. El sistema debe mostrar un menú con las siguientes opciones:

1. Ingreso de auto
2. Salida de auto
3. Consultas
 1. Dinero Recaudado
 2. Regresar al menú principal
4. Salir

entradasYsalidas.txt

```
GGJ-465, 1359335103
GST-4767, 1359335130
GHG-876, 1359338086
GGJ-465, 1359338005
GHG-876, 1359338399
GIR-8765, 1359338399
```

dinero.txt

```
GGJ-465, 2902, 1.20
GHG-876, 313, 0
```

El parqueo necesita llevar un registro de los autos que ingresan y salen para poder cobrar a los conductores las siguientes tarifas:

- 10 minutos o menos GRATIS
- 1 hora o fracción \$1.20

Implemente:

1) El procedimiento **ingresoSalida(FILE *eYs, char * placa, int segundos)**, que recibe un puntero al archivo "entradasYsalidas.txt", el número de la placa del auto que ingresa y la hora de ingreso que es la hora actual del calendario en segundos desde ene 1, 1970. (La hora puede ser obtenida utilizando la función time(NULL) de la librería time.h)

2) La función **float pago(FILE *eYs, char * placa, FILE *d)** que recibe un puntero al archivo "entradasYsalidas.txt", el número de la placa del auto y un puntero al archivo "dinero.txt". La función calcula y retorna el valor que debe pagar el conductor del mismo por el tiempo que ha estado en el parqueo. También escribe en el archivo "dinero.txt" la placa del vehículo, la diferencia 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 su salida cobrar la tarifa. Mostrar el total de dinero recaudado. Use los procedimientos y funciones implementados en los literales anteriores.

Nota: Asuma que un auto sólo puede ingresar una vez al parqueo en toda su vida.