**Carrera: Ingeniería en Computadores**

**Curso: CE-1101 Introducción a la Programación, I Semestre 2017**

**Examen parcial I, grupos 4 y 6**

**Valor: 100 puntos, 20%**

**Fecha: 2 de abril de 2017**

**Hora: 9 a.m. a 11 a.m. (2 horas)**

**Profesor: Antonio González Torres**

**Instrucciones**

1. Las copias de exámenes se anulan automáticamente.
2. Debe utilizar los nombres de función **exactamente** como se indican en el enunciado.
3. Debe enviar **un solo** archivo .py con las soluciones a más tardar el 4 de abril a las 23:55 al email [antonio.gonzalez@tec.ac.cr](mailto:antonio.gonzalez@tec.ac.cr).
   1. Las soluciones deben ser copia fiel del examen entregado en formato físico al profesor el día 2 de abril de 2017.
   2. El alumno puede tomar fotos al examen para pasarlo a la computadora.
   3. El nombre del archivo debe ser enviado usando el siguiente formato: primera letra del nombre – primer apellido – parcial1, por ejemplo, para Antonio González el nombre del archivo sería **agonzalez-parcial1.**
   4. Después de la fecha y hora indicada no se recibirán las soluciones al examen, por lo que la revisión queda sujeta a la comprensión por parte del profesor de las respuestas entregadas en formato físico.
4. Utilice recursión de pila para responder todas las preguntas.
5. Para cada respuesta, utilice una función para validar los parámetros y restricciones de las entradas.

**Preguntas**

1. Escriba un programa para convertir números positivos en base 10, con valores decimales, a binario. Puede utilizar dos funciones para hacer las conversiones, una de las funciones podría calcular el valor binario de la parte entera del número, mientras que la otra función podría calcular la parte fraccional. El resultado se debe devolver usando dos listas y no puede usar funciones propias de Python para realizar ninguno de los cálculos u operaciones. Recuerde que el proceso para convertir la parte entera de un número en base 10 a binario consiste en divisiones sucesivas entre 2 y que los residuos que conforman el resultado, mientras que para obtener el valor binario de los valores decimales se realizan multiplicaciones sucesivas por 2 y la parte entera del resultado da origen al número binario.

>>> convertir(20.25)

([1,0,1,0,0],[0, 1])

>>> convertir(33,375)

([1,0,0,0,0,1],[0,1,1])

1. Programe una función que devuelva una lista con los divisores de un número entero. La lista está compuesta por todos los divisores del número, excepto por 1 y él mismo.

>>> divisores(10)

[2, 5]

>>> divisores(21)

[3, 7]

>>> divisores(12)

[2, 3, 4, 6]

1. Escribir una función denominada sumar(lista), que reciba como argumento una lista compuesta por elementos y sublistas y obtenga como resultado la suma de todos los elementos de la lista y de todas sus sublistas. No puede utilizar ninguna función de Python para manipular la lista ni para efectuar la suma.

>>> sumar([[1, 2, 4], 1, [1, 2, [3, 4]]])

18

>>> sumar([[1, [2, [3, [4]],[10]]]])

20

1. Escriba una función que recibe como entrada dos parámetros llamados digi y num: ambos números son enteros positivos. La función debe multiplicar cada dígito de num por digi y si el resultado es mayor a 9 debe descartar el resultado. El resultado final será el número que se forma con las multiplicaciones válidas. No puede utilizar ninguna función de Python para resolver el problema (ejercicio modificado del original).

>>> multid(3, 382)

96

>>> multid(5, 2118)

55

>>> multid(2, 2431)

4862