**Carrera: Ingeniería en Computadores**

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

**Examen parcial I, grupo 2**

**Valor: 100 puntos, 22%**

**Fecha: 8 de septiembre de 2017**

**Hora: 1 p.m. a 3 p.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 9 de septiembre 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 8 de septiembre 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. En todos los casos esta función tendrá un valor de 3 puntos.

**Preguntas (25 puntos cada una)**

1. Implemente un programa que tome como entrada una lista, multiplique cada valor de la lista por el índice de su posición y devuelva otra lista con el resultado.

>>> multiplicaciones ([5, 6, 2, 12, 15]) – Nota de ejecución: (0\*5)–(1\*6)-(2\*2)-(3\*12)-(4\*15)

[0, 6, 4, 36, 60]

1. Tome en cuenta la definición que se muestra a continuación. Desarrolle un programa en Python que recibe como entrada una lista de números de longitud n, los cuales son elevados a una potencia y sumados. Si la lista contiene sublistas como elementos, debe procesar cada elemento de estas, elevándolos a la potencia correspondiente y sumando el valor obtenido al resultado.

>>> suma ([5, [2,4], [3, [9,6],8]]) – Nota de ejecución:

2203011

1. El símbolo ∏ se usa para el producto de una serie de números, de forma análoga al uso de Σ para la suma de términos. La definición del uso de esta notación se muestra a continuación:

Considere el siguiente ejemplo, ,

y programe una función Python que recibe como parámetro el valor de n para resolver la siguiente operación:

>>> summ(3) – Nota de ejecución: (3\*1 - 1) + ((3\*1 - 1) \* (3\*4 - 2)) + ((3\*1 - 1) \* (3\*4 - 2) \* (3\*9 - 3))

502

>>>summ(4) – Nota de ejecución: (3\*1-1) + ((3\*1-1) \* (3\*4-2)) + ((3\*1-1) \* (3\*4-2) \* (3\*9-3)) +

((3\*1-1)\*(3\*4-2)\*(3\*9-3)\*(3\*16-4))

21622

1. Escriba un programa que recibe un número conformado por una cantidad de par de dígitos (e.g., 2, 4, 6 y 8), los cuales son procesados en parejas. Los dígitos que conforman cada pareja son intercambiados, como se muestra en los ejemplos de ejecución. El número no puede contener el dígito cero.

>>> intercambiar (185412) – Nota de ejecución: 185421 – 184521 - 814521

814521

>>> intercambiar (849865) – Nota de ejecución: 849856 – 848956 - 488956

488956

>>> intercambiar (21459887) – Nota de ejecución: 21459878–21458978–21548978-12548978

12548978