Universidad Simón Bolívar
Departamento de Computación y Tecnología de la información
CI-2691- Laboratorio de algoritmos I

Laboratorio 5

El objetivo de este laboratorio es la verificación y prueba de aserciones correspondientes a precondiciones, postcondiciones e instrucciones de asignación, secuenciación y selección, así como funciones de cota de una iteración.

Ejercicios a entregar

Para cada uno de los siguientes problemas proponga una solución a través de un programa, escriba la función de cota en cada ciclo utilizado. Adicionalmente, coloque las acciones de verificación de precondición y postcondición, tal como se explicó en el prelaboratorio 5, usando la instrucción `try`/`except`.

1. **Lab05Ejercicio1.py**: Los números naturales pueden ser perfecto, deficiente o abundante. Un número *perfecto*, cuando la suma de sus divisores propios positivos es igual al número dado (la unidad se considera divisor propio, pero no lo es el mismo número). En cambio, un número es *deficiente*, cuando la suma de sus divisores es menor que el número dado. En contraste a los números perfectos y deficientes, un número es *abundante*, cuando la suma de sus divisores propios es mayor que el número dado.

   Escriba un programa que reciba un número natural y determine si es perfecto, deficiente o abundante.

2. **Lab05Ejercicio2.py**: Verificar si una secuencia de enteros no negativos provista por el teclado está ordenada en forma creciente. La secuencia de entrada termina cuando se recibe el número **0** (que no se considera para la verificación del ordenamiento, sino como una marca de fin). No se admite secuencia vacía ni unitaria y la utilización de los métodos *sorted* y *sort* de las listas de Python.

   Nótese: que en este caso no se conoce el tamaño de la secuencia, por lo que para establecer una función de cota debe asumir un número máximo de iteraciones. Este número se asume muy grande para dar la posibilidad que el usuario introduzca una entrada adecuada sin producir que el programa aborte.

3. **Lab05Ejercicio3.py**: Hay una conjetura sobre secuencias de números naturales (enteros no negativos) que dice: “Si se toma cualquier número natural **x**, el siguiente de la secuencia se genera así: si el número **x** es par, el siguiente número será igual a la mitad **x**; si el número **x** es impar, el siguiente se obtiene al multiplicar por **3** y sumarle **1**. Si el proceso se repite, en algún número finito de pasos se llegará a la secuencia **4**, **2**, **1**”.

En consecuencia, se debe escribir un programa que reciba un número natural **x** y obtenga las secuencia necesarias para llegar al número **4**, en cada paso se debe mostrar el número de elementos de la secuencia y el valor actual de **x**.

Nota: Una vez verificado y finalizado su programa, realice las siguientes pruebas y observe lo que ocurre:

a) Ejecute los programas con datos de entrada incorrectos.
b) Modifique ligeramente algunas de las acciones (introduzcan errores adrede dentro del código), como por ejemplo, no incrementar la variable de control del ciclo, cambiar la condición de parada del ciclo, cambiar el cálculo del resultado principal del ciclo.
c) Corra sus programas con datos correctos.

Condiciones de entrega

Cree un archivo comprimido del tipo "tgz" llamado Lab5-X.tgz, donde X es su número de carné, que contenga los programas **Lab05Ejercicio1.py, Lab05Ejercicio1.py y Lab05Ejercicio1.py**. Debe subir el archivo en el aula virtual, en la sección del Laboratorio 5 antes de las 11:30 am del martes 18 de octubre de 2016.

Referencias

[1] **ForLoop, Documentación oficial de Python**. Disponible en la Web. https://wiki.python.org/moin/ForLoop

[2] **The Python Standard Library**, Document versión 3.3, capítulo 4. Disponible en la web: https://docs.python.org/3.3/tutorial/controlflow.html