Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

IE-0117 Programación Bajo Plataformas Abiertas

Ing. Marco Villalta Fallas - I Ciclo 2018

Laboratorio # 8

Ejercicios de programación en Python

Instrucciones Generales:
Los laboratorios se deben de realizar de manera individual.
El laboratorio debe de entregarse a más tardar el 11 de junio antes de las 23:59.
Entregue un archivo comprimido que incluya un directorio llamado `informe` con los archivos necesarios para generar el PDF del informe (.tex, imágenes, código, entre otros) y un directorio llamado `src` con los archivos de código fuente que lleven a la solución. Cualquier otro formato o entrega tardía no se revisará y el laboratorio tendrá una nota de cero.

Ejercicios

Parte 1

Nota: Resuelva cada ejercicio en un archivo diferente.

1. Escriba una función en Python que reciba tres argumentos y que imprima en consola cual es el mayor y cual es el menor. Compruebe su funcionalidad en un programa principal.

2. Escriba una función en Python que reciba una lista por argumento. La función debe regresar una lista nueva (una copia, sin alterar la lista fuente) que contenga únicamente el primer y último elemento de la lista original.

3. Escriba un script en Python que simule un juego sencillo de adivinanza. Implemente las siguientes funciones:

   - Función `random`. Regresa un número aleatorio entre 0 y 50 (inclusive).
   - Función `obtenerInt` . Solicita un número al usuario y lo regresa.
   - Función de respuesta, debe de contestar: el número secreto es mayor, el número secreto es menor, o adivinaste el número secreto según la respuesta del usuario (salida de la función `obtenerInt`).
   - Función `adivinanza`. Lógica que le de la funcionalidad al juego. El juego debe terminar hasta que el usuario adivine el número.

4. Realice el juego del punto 3, pero al revés. Ahora el usuario elige un número en el rango de 0 a 50, y la computadora dirá un número, y el usuario contestará si el número fue menor, mayor o igual.

5. Escriba una función en Python que reciba un número entero, y regrese una lista con todos los divisores del número.

6. Escriba una función en Python que ordene una lista de números. Recibe la lista como parámetro, y además una letra que es a para ordenar de manera ascendente y d de manera descendente.

7. Escriba una función en Python que imprima elemento por elemento una lista. Impleméntelo de manera recursiva. Explique en el informe la lógica que utilizó para la solución.

Parte 2

Implemente el siguiente ejercicio:

Solución de un Laberinto

Se tiene la siguiente representación de un laberinto:

```
####################################
#E#  ##  ######## # #      #     # #
# #   #             # #        #   #
#   # ##### ## ###### # #######  # #
### # ##    ##      # # #     #### #
#   #    #  #######   #   ###    #S#
####################################
```

En ella, el símbolo numeral (`#`) representa las paredes, mientras que la letra `E` y `S` representan entrada y salida respectivamente. Los espacios en blanco muestran los corredores que se pueden recorrer.

Para resolver el ejercicio, debe seguir los siguiente pasos:

1. Leer el laberinto desde un archivo de texto plano, y convertirlo a una representación de listas en Python en donde cada cuadrícula del laberinto representa una entrada de la lista bidimensional.
   - Si la cuadrícula es un corredor por el cual se puede transitar, su representación en la lista bidimensional debe ser un 0.
   - Si la cuadrícula es una pared, su representación en la lista bidimensional debe ser un 1.
   - Si la cuadrícula es la salida, su representación en la lista bidimensional debe ser un 2.
   - Si la cuadrícula ya ha sido visitada por el algoritmo, su representación en la lista bidimensional debe ser un 3.
2. Implemente la lógica de forma recursiva para recorrer el laberinto y encontrar una solución.
3. Finalmente, una ves solucionado el laberinto, convierta la representación de la lista bidimensional de nuevo a la forma original y escriba el contenido en un archivo de texto plano llamado `solucion.txt`.

   La solución debe de verse de la siguiente forma:

```
####################################
#S#  ##  ######## # #oooooo#  ooo# #
#o#ooo#    oooo     #o#   ooooo#ooo#
#ooo#o#####o##o######o# #######  #o#
### #o##oooo##oooooo#o# #     ####o#
#   #oooo#  #######ooo#   ###    #E#
####################################
```