

CC1002 Introducción a la Programación

Auxiliar 5

Prof. Benjamin Bustos
Auxs. Cristobal Sepúlveda, Franco Sepúlveda

Fecha: 6 de septiembre de 2017

Resolver los problemas usando Python, utilizando estrictamente la Receta de Diseño entregada a lo largo del curso. Use nombres apropiados para funciones y variables, y realice testing cada vez que sea posible.

1. Listas

Utilizando el módulo *lista*, implemente las siguientes funciones:

1. Cree una función llamada *quintoElemento*, que reciba una lista de *Strings* y devuelva, si existe, el quinto elemento de la lista. Si no existe, debe devolver **None**.
2. Cree una función de nombre *cuentaElementos* que devuelva el largo de una lista cualquiera.
3. Implemente la función *pares*, la cual recibe una lista de números enteros y devuelve una nueva lista, la cual sólo contiene los número pares de la lista recibida.
4. Implemente la función *perimetro*, la cual recibe una lista de *puntos3D* y devuelve la suma de las distancias entre cada par contiguo de la curva. Para esto, importe el módulo *estructura* y cree la estructura *punto3D* que almacene tres valores enteros referentes a las coordenadas (x,y,z) del punto. Además, implemente la función *distancia* que calcular la distancia entre dos puntos 3D. Recuerde que la curva es cerrada, es decir, el primer punto y el último punto de la lista son contiguos.

Recordatorio: El razonamiento recursivo es vital para trabajar sobre estructuras recursivas. Además, la estructura *lista*, cuenta con las funciones *cabeza*, *cola*, *crearLista* y *vacía*.

2. Adivinar un número

Escriba un programa interactivo que le permita jugar a adivinar un número. Primero, el computador pide al usuario un valor entero $n > 0$

(se debe repetir la pregunta mientras el dato ingresado por el usuario no sea válido). Luego, el computador debe generar aleatoriamente un número en el intervalo $[1, n]$ y preguntar al usuario para que lo adivine. Si el usuario adivina el número, se termina el juego, y se debe mostrar cuántos intentos requirió el usuario para adivinarlo. Si el usuario se equivoca, el computador debe mostrar en pantalla si el número a adivinar es menor o mayor que el que ingresó el usuario, y volver a pedirle un número. El usuario puede rendirse ingresando un 0, en cuyo caso el computador debe mostrar cuántos intentos hizo el usuario antes de rendirse.

Ejemplo:

```
Valor inicial? 100
Adivina el numero [1,100]: 50
El numero a adivinar es menor que 50
Adivina el numero [1,100]: 2
El numero a adivinar es mayor que 2
Adivina el numero [1,100]: 25
El numero a adivinar es mayor que 25
Adivina el numero [1,100]: 40
El numero a adivinar es menor que 40
Adivina el numero [1,100]: 30
El numero a adivinar es menor que 30
Adivina el numero [1,100]: 27
Has ganado! Lo adivinaste en 5 intentos.
```