

CC1002 Introducción a la Programación

Auxiliar 1

Prof. Benjamin Bustos
Auxs. Cristóbal Sepúlveda, Franco Sepúlveda

Fecha: 9 de agosto de 2017

Todos los problemas deben ser resueltos en 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.

Problema 1

1. Implemente la función `sumarUno`, que recibe un número entero y devuelve el número incrementado en uno.
2. Utilizando la función `sumarUno`, implemente un programa interactivo que siga el siguiente diálogo:

```
Ingresa un numero: 123
124 te gane!
```

Problema 2

1. Implemente la función `def sumarFracciones(numerador1, denominador1, numerador2, denominador2)`, que recibe los numeradores y denominadores (valores enteros) de dos fracciones y retorna la suma expresada como un número real.
2. Utilizando la función `sumarFracciones`, implemente un programa interactivo que siga el siguiente diálogo:

```
Suma de fracciones: a/b + c/d
a? 1
b? 2
c? 3
d? 4
suma = 1.25
```

Problema 3

El módulo `random` de Python provee algunas funciones útiles para generar números aleatorios. Por ejemplo, contiene las siguientes funciones (nota: `None` en los parámetros del contrato significa que la función no recibe ningún parámetro):

```
# random: None -> float
# devuelve un número aleatorio en el intervalo [0,1)
# ejemplo: random.random() retorna 0.8535574455410475
# ejemplo: random.random() retorna 0.568816500411932
def random():
    ...

# randint: int int -> int
# devuelve un numero entero aleatorio en el intervalo
# [a, b]. Si a > b se produce ValueError
# ejemplo: random.randint(10, 20) devuelve 20
# ejemplo: random.randint(10, 20) devuelve 15
def randint(a, b):
    ...
```

1. Implemente la función `def dado()`, que simule el lanzamiento de un dado y devuelva el valor obtenido.
2. Implemente la función `randint` a partir de la función `random`. Puede suponer que siempre se cumple que $a \leq b$.