Universidad del Desarrollo
Facultad de Ingeniería

Tecnologías de Información 2
Certamen 2 - Parte conceptual

12 de Noviembre de 2018

Instrucciones:

- Indique su nombre y sección.
- Lea atentamente el enunciado de cada uno de los problemas
- Escriba con letra legible la respuesta.
- Duración: 1h 30m

Nombre:

Sección:

(1.2pts) Verdadero o Falso. Escribe V o F en mayúsculas. Si la respuesta es falsa, debe justificarla para considerar correcta su respuesta.

a) __F__ Si a, b y c son listas de tamaño N, el código `c = a + b` es equivalente a `for i in range(N): c[i] = a[i] + b[i]`.

Justificación: Falso porque el operador + concatena dos listas.

b) __F__ Una vez creada una tupla a, e inicializada con 5 elementos, puedes recuperar el primer elemento de la tupla con el código `a[1]`.

Justificación: El primer elemento está en a[0].

c) __V__ Los diccionarios de Python permiten mezclar diferentes tipos de datos para los valores.

Justificación:

d) __F__ En un conjunto de Python s = {1,2,3}, es posible obtener el último elemento con s[-1]

Justificación: Los conjuntos no se pueden acceder por la posición de los elementos. Solo se puede recorrer el conjunto, o saber si un elemento está presente (también se puede añadir, eliminar y calcular otras operaciones entre conjuntos).

e) __V__ Si la clave de un diccionario es una lista, el programa entrega error.

Justificación:

f) __F__ Si a y b son conjuntos (set) el código a = b copia los elementos que están en b en la variable a.

Justificación: a = b crea un alias, es decir, la variable a es una alias del contenido del conjunto b.

(1pto) Funciones

Nuestro objetivo es desarrollar una función para computar el valor absolutos de cada uno de los elementos de una lista de valores enteros. Por ejemplo, para la lista [1, -6, 2, -1] el resultado deseado es [1, 6, 2, 1]. El código de abajo es la solución a este problema, con cinco fragmentos de código faltantes. Tu tarea es identificar las líneas código que completan este programa. Para esta tarea exploraremos dos estrategias distintas.

`def abs(x):`
`  N = len(x)`
`  UNO`
`  for i in range(N):`
`  if x[i] < 0:`
`  DOS`
`  TRES`
`  CUATRO`
	A. return x
	B. return z
	C. x[i] = -x[i]
	D. z = [0]*N
	E. z[i] = -x[i]
	F. else: z[i] = x[i]
	G. x[i] = x[i]
	H. else: z[i] = -x[i]
	I. No código para esta caja

(0.5pts) Primera estrategia: suponga que la estrategia es modificar la lista que se pasa por parámetro para reflejar el resultado. Usando la tabla de la derecha, complete cada caja con la letra correspondiente al código que es necesario para completar esta tarea. Puede usar más de una vez la misma alternativa.

UNO: **I** DOS: **C** TRES: **I o G** CUATRO: **I o A**

Nota: Si en UNO puso D y en TRES puso F, entonces también es una posible respuesta correcta.

(0.5pts) Segunda estrategia: suponga que la siguiente estrategia es retornar una nueva lista que refleje el resultado. Otra vez, use el código de las alternativas de la derecha para completar esta tarea. Puede usar más de una vez la misma alternativa.

UNO: **D** DOS: **E** TRES: **F** CUATRO: **B**

(1pto) Funciones recursivas

```
def r(n):
    if n == 0: return 0
    if n == 1: return 0
    if n == 2: return 1
    return r(n-1) + r(n-2) - r(n-3)
```

A. ¿Cuál es el valor de r(3)? **1**
B. ¿Cuál es el valor de r(0)? **0**
C. ¿Cuál es el valor de r(4)? **2**
D. ¿Cuál es el valor de r(7)? **3**
E. ¿Cuántas veces fue necesario ejecutar la función r() para obtener r(7)? **46 (incluyendo r(7)), o 45 (sin incluir r(7)).**

(1.2pts) Al lado de cada bloque de código, marca si el código genera un error, o genera un loop infinito, o no hace error. Si no hace error, escribe lo que imprime. Asume que A = False; B = True; C = False;

a)

```
while A = True:
    B = False
print(B)
```

X Genera un error
◯ Es un loop infinito
◯ No hace error, e imprime: _____

b)

```
while A == False:
    if B == False: A = False
    if C == False: B = False
print(B)
```

◯ Genera un error
X Es un loop infinito
◯ No hace error, e imprime: _____

c)

```
for i in range(100):
    temp = A
    A = B
    B = temp
print(A)
```

◯ Genera un error
◯ Es un loop infinito
X No hace error, e imprime: __False___

d)

```
d = dict()
d[A] = B
d[(A, B)] = B
print(d[(A, B)])
```

◯ Genera un error
◯ Es un loop infinito
X No hace error, e imprime: __True__

e)

```
s = set()
s.add([A, B])
if [A,B] in s: print('existe')
else: print('no existe')
```

X Genera un error
◯ Es un loop infinito
◯ No hace error, e imprime: _____

(1.5pts) Escriba en la siguiente caja, código que cree una variable A que represente una matriz de N x N, donde cada celda de la matriz contenga 3 valores. Cada elemento de la matriz debe estar inicializado con el valor 0. El input para la variable N es un número entero positivo. Por ejemplo, para N = 3, este es el contenido de la variable A:

```
[ [[0, 0, 0], [0, 0, 0], [0, 0, 0]],
  [[0, 0, 0], [0, 0, 0], [0, 0, 0]],
  [[0, 0, 0], [0, 0, 0], [0, 0, 0]] ]
```

1.	`N = int(input())`
2.	`A = [ ]`
3.	`for i in range(N):`
4.	`    A.append(list())`
5.	`    for j in range(N):`
6.	`        A[i].append([0,0,0])`
7.	
8.	
9.	
10.	`print(A)`

**Nota: no es necesario utilizar todas las líneas de código*