

Corrección Parcial #1_Parte 1-2_ Daniela Gaitan C

September 23, 2016

1 Corrección Parcial 1

Métodos Computacionales para Políticas Públicas - UROSARIO
[Daniela Gaitán Cotrino] [daniela.gaitanc@urosario.edu.co]

1.1 Parte 1

1.

```
x = 15  
x = x + 5  
print(x)
```

¿Qué resultado arroja el anterior código? Rta/ 20

2.

¿Qué entendemos por “código” en el contexto de este curso? Rta/ Una secuencia de instrucciones en un lenguaje de programación

3.

¿Una memoria USB es un ejemplo de qué componente de la arquitectura de un computador? Rta/ Secondary Memory

4.

```
print(94.2)
```

En el código anterior, ¿qué es 94.2? Rta/ Una constante

5.

```
x = 94.2
```

En el código anterior, ¿qué es 94.2? Rta/ Una variable

6.

¿Cuál de las siguientes no es una palabra reservada en Python? if break for else

Rta/ Ninguna de las anteriores

7.

```
a = a + 3
```

¿Qué hace el código anterior? Rta/ Toma el valor de "a", le suma 3, y asigna el resultado a "a"

8.

```
35 % 7
```

```
5 7 "35 % 7" 35
```

¿Cuál es el valor de la expresión anterior? Rta/ Ninguna de las anteriores

9.

```
x = 3 + 2 * 3 - 20 / 4 print(x)
```

¿Cuál es el resultado de la expresión anterior? Rta/ 4

10.

```
x = int(13.6)
```

¿Qué valor tiene "x" asignado si se ejecuta la anterior expresión? Rta/ 13

11.

¿Cómo se indica en Python el final de un bloque de código constitutivo de una función? Rta/
Volviendo al nivel de indentación de la palabra clave "def"

12.

```
def thing(): print("Hello") print("there")
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/ there

13.

```
def greet(lang): if lang == "es": return("Hola") elif lang == "fr": return("Bonjour") else:  
return("Hello")  
print(greet(fr), "Michael")
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/ Bonjour Michael

14.

```
def addtwo(a, b): added = a + b return a x = addtwo(2, 7) print(x)
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/ 2

15.

```
n = 5 while n > 0: print(n)
print("All done")
```

¿Qué está mal en el anterior código? Rta/ Este loop correrá por siempre

16.

```
tot = 0 for i in [5, 4, 3, 2, 1]: tot = tot + 1 print(tot)
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/ 5

17.

```
friends = [Joseph, Glenn, Sally] for friend in friends: print("Happy New Year:", friend)
print("Done!")
```

¿Cuál es el iterador en el anterior código? friends Joseph Glenn Sally

Rta/ Ninguna de las anteriores

18.

```
zork = 0 for thing in [9, 41, 12, 3, 74, 15]: zork = zork + thing print(zork)
```

¿Cuál es una buena descripción del anterior bloque de código? Rta/ Sume todos los elementos de una lista

19.

```
smallest_so_far = -1 for the_num in [9, 41, 12, 3, 74, 15]: if the_num < smallest_so_far: small-
est_so_far = the_num print("smallest_so_far")
```

¿Cuál es el resultado de ejecutar el anterior código? -1 3 74 9

Rta/ Ninguna de las anteriores. Sin embargo, el corrector del cuestionario dice que es -1

20.

```
str1 = "Hello" str2 = "there" bob = str1 + str2 print(bob)
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/ Hellothere

21.

```
x = "40" y = int(x) + 2 print(y)
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/ 42

22.

¿Cómo usaría el operador de indexación [] para imprimir la letra “q” dado el siguiente código?

```
x = "From marquard@uct.ac.za"
```

Rta/ `x[8]`

23.

¿Cómo usaría el operador de “slicing” [:] para imprimir el texto “uct” dado el siguiente código?

```
x = "From marquard@uct.ac.za"
```

Rta/ `x[14:17]`

24.

¿Cuál es el resultado de ejecutar el siguiente código? `print(len("banana") * 7)`

Rta/ 42

25.

¿Cómo se imprimiría el contenido de la siguiente variable en mayúsculas sostenidas? `greet = "Hello Bob"`

Rta/ `print(greet.upper())`

26.

```
data = "From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008" pos = data.find(".")  
print(data[pos: pos+3])
```

¿Cuál es el resultado de ejecutar el anterior código? Pista: el método “find” encuentra la primera ocurrencia de su argumento.

Rta/ `.ma`

27.

¿Qué indica el segundo parámetro de la función predefinida `open()`? Rta/ El modo (lectura o escritura) de apertura del archivo

28.

¿Para la siguiente lista, ¿cómo se imprimiría “Sally”? `friends = [Joseph, Glenn, Sally]`

Rta/ `print(friends[2])`

29.

```
fruit = "Banana" fruit[0] = "b" print(fruit)
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/ Ninguna de las anteriores

30.

```
a = [1, 2, 3] b = [4, 5, 6] c = a + b print(len(c))
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/ 6

31.

¿En qué se diferencian los diccionarios de las listas? Rta/ Las listas son indexadas usando enteros y los diccionarios pueden usar “strings” como índices

32.

```
stuff = { } print(stuff.get("candy", -1))
```

¿Cuál es el resultado de ejecutar el anterior código? Rta/-1

33.

¿Cuál es un uso común de diccionarios en Python? Rta/ Construir un histograma con la frecuencia de ocurrencia de determinado objeto

34.

```
if key in counts: counts[key] = counts[key] + 1 else: counts[key] = 1
```

¿Cuál de las siguientes líneas de código es equivalente al bloque arriba, asumiendo que “counts” es un diccionario? Rta/ counts[key] = counts.get(key, 0) + 1

1.2 Parte 2

```
In [57]: 1.
```

```
x = 7
y = 5.0
z = 10.0

w = x % 2 + y / z + z + y / (z + z)
# ¿Cuál es el valor de "w" después de ejecutar el anterior código?
```

```
In [58]: print(w)
```

11.75

```
In [59]: 2.
```

```
c = True
d = False
c = c and d
c = not c or d

# ¿Cuál es el valor de "c" después de ejecutar el anterior código?
```

```
In [60]: c
```

```
Out[60]: True
```

```
In [61]: 3.
```

```
d = 0
for p in range(0, 5):
    if p % 4 == 0:
        d = d + (p-1) * 25;
    else:
        d = d + 100;
print("$" + str(d) + "." + str(d % 100))

# ¿Qué resultado arroja el anterior código?
```

```
$350.50
```

```
In [13]: def mystery1(l):
        rv = []
        for x in l:
            rv = [x] + rv
        return rv

l = [0, 1, 2, 3, 4]
nl = mystery1(l)
print("nl:", nl)

# ¿Qué resultado arroja el anterior código?
```

```
nl: [4, 3, 2, 1, 0]
```

```
In [62]: 4.
```

```
def f(i):
    i = i + 2
    return i

def g(i):
    j = h(i * 4)
    return j

def h(i):
    return i - 1

def main():
    i = 6
```

```
j = k = l = 2
j = 2 * f(i)
k = f(j * 2)
l = g(i) / 2
return i, j, k, l
```

```
main()
```

```
# ¿Qué resultado arroja el anterior código?
```

```
Out[62]: (6, 16, 34, 11.5)
```