

mcpp_taller4_camila_valencia

August 30, 2016

1 Taller 4

Métodos Computacionales para Políticas Públicas - UROSARIO

Entrega: viernes 2-sep-2016 11:59 PM

[Camila Valencia] [camila.valencia@urosario.edu.co]

1.1 Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller4_santiago_mataallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto “[Su nombre acá]” con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
 1. Descárguelo en PDF.
 2. Suba los dos archivos (.pdf y .ipynb) a su repositorio en GitHub antes de la fecha y hora límites.

(Todos los ejercicios tienen el mismo valor.)

1.2 Zelle, Exercises 6.8 (p. 159):

- True/False: 1-10
- Multiple choice: 2, 3, 6, 7, 10
- Programming Exercises: 1, 3, 4, 11, 12, 13

2 True/False

1. False
2. False
3. True
4. True
5. False
6. False
7. False
8. True
9. True
10. False

3 Multiple Choice

2. a
3. a
4. a
5. d
6. a

4 Programming Exercises

1. Write a program to print the lyrics of the song “Old MacDonald”. Your program should print the lyrics for five different animals.

```
In [1]: def sing (animal, sound):  
        print("Old MacDonald had a farm, Ee-igh, Ee-igh, Oh! \nAnd on that farm  
  
        animals=[ "chick", "duck", "cat", "dog", "sheep"]  
        sounds=[ "chick", "quack", "meow", "wow", "baa"]  
  
        for i in list(range(len(animals))):  
            cancion= sing(animals[i],sounds[i])
```

```
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!  
And on that farm he had a  chick , Ee-igh, Ee-igh, Oh!  
With a  chick , chick  here and a  chick , chick there.  
Here a  chick , there a chick , everywhere a  chick , chick .  
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!  
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!  
And on that farm he had a  duck , Ee-igh, Ee-igh, Oh!  
With a  quack , quack  here and a  quack , quack there.  
Here a  quack , there a quack , everywhere a  quack , quack .  
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!  
Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!  
And on that farm he had a  cat , Ee-igh, Ee-igh, Oh!
```

With a meow , meow here and a meow , meow there.
 Here a meow , there a meow , everywhere a meow , meow .
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 And on that farm he had a dog , Ee-igh, Ee-igh, Oh!
 With a wow , wow here and a wow , wow there.
 Here a wow , there a wow , everywhere a wow , wow .
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!
 And on that farm he had a sheep , Ee-igh, Ee-igh, Oh!
 With a baa , baa here and a baa , baa there.
 Here a baa , there a baa , everywhere a baa , baa .
 Old MacDonald had a farm, Ee-igh, Ee-igh, Oh!

3. Write definitions for these functions:

sphereArea(radius) Returns the surface area of a sphere having the given radius.

sphereVolume Returns the volumen of a sphere having the given radius.

Use your functions to solve programming excercise 1 from chapter 3.

Write a program to calculate the volume and surface area of a sphere from its radius, given as input.

```
In [9]: pi=3.14159265359
```

```
def sphereVolume(r):
    volume = (4/3)*pi*(r**3)
    return volume

def sphereArea(r):
    area = 4*pi*(r**2)
    return area
```

```
In [10]: sphereArea(1)
```

```
Out[10]: 12.56637061436
```

```
In [11]: sphereVolume(1)
```

```
Out[11]: 4.188790204786667
```

4. Write definitions for the following two functions:

sumN(n) returns the sum of the first n natural numbers

sumNcubes(n) returns the sum of cubes of the first n natural numbers

```
In [14]: def sumN(n):  
        m=0  
        for i in range (n+1):  
            m=m+i  
  
        return m
```

```
In [19]: sumN(3)
```

```
Out[19]: 6
```

```
In [17]: def sumNcubes(n):  
        m=0  
        for i in range (n+1):  
            m=m+(i**3)  
        return m
```

```
In [18]: sumNcubes(3)
```

```
Out[18]: 36
```

11. Write and test a function to meet this specification

squareEach(nums) nums is a list of numbers. Modifies the list by squaring each entry.

```
In [29]: def squareEach(nums):  
        for i in range(len(nums)):  
            nums[i]=nums[i]**2  
        return nums
```

```
In [30]: nums=[2,3,4,5,6,7]  
        squareEach(nums)
```

```
Out[30]: [4, 9, 16, 25, 36, 49]
```

12. Write and test a function to meet this specification

sumList(nums) nums is a list of numbers. Returns the sum of the numbers in the list.

```
In [31]: def sumList(nums):  
        x=0  
        for i in range(len(nums)):  
            x=x+nums[i]  
        return x
```

```
In [32]: nums=[2,3,4,5,6,7]  
        sumList(nums)
```

```
Out[32]: 27
```

13. Write and test a function to meet this specification

toNumbers(strList) strList is a list of strings, each of which represents a number. Modifies each entry in the list by converting it to a number.

```
In [33]: def toNumbers(strList):  
         for i in range(len(strList)):  
             strList[i]=ord(strList[i])  
         return strList
```

```
In [37]: lista = ["m", "p", "z", "a"]  
         toNumbers(lista)
```

```
Out[37]: [109, 112, 122, 97]
```