

2021-I_mcpp_taller_4_Valentina_Cuenca

February 27, 2021

1 Taller 4

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 27-feb-2021 11:59 PM

Valentina Cuenca norma.cuenca@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_matallana
- 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

1.2.1 True/False: 1-10

1. False
2. False
3. True

4. False
5. True
6. False
7. False
8. True
9. True
10. True

1.2.2 Multiple choice: 2, 3, 6, 7, 10

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

1.2.3 Programming Exercises: 1, 3, 4, 11, 12, 13

1.

```
[17]: def oldMacdonal():
    ↵
    ↪listaA=["vacas","mu","gatos","miau","perros","guau","gallinas","co-co","patos","Cuak"]
    for i in range(0,len(listaA),2):
        print("El viejo MacDonald tenía una granja, Ee-igh, Ee-igh, ¡Oh! Y en↵
    ↪esa granja tenía {0} , Ee-igh, Ee-igh, ¡Oh! Con un {1}, {1} aquí y un {1},↵
    ↪{1} allí. Aquí un {1}, hay un {1}, en todas partes un {1},{1} . ").
    ↪format(listaA[i], listaA[i+1]))
oldMacdonal()
```

El viejo MacDonald tenía una granja, Ee-igh, Ee-igh, ¡Oh! Y en esa granja tenía vacas , Ee-igh, Ee-igh, ¡Oh! Con un mu, mu aquí y un mu, mu allí. Aquí un mu, hay un mu, en todas partes un mu,mu .

El viejo MacDonald tenía una granja, Ee-igh, Ee-igh, ¡Oh! Y en esa granja tenía gatos , Ee-igh, Ee-igh, ¡Oh! Con un miau, miau aquí y un miau, miau allí. Aquí un miau, hay un miau, en todas partes un miau,miau .

El viejo MacDonald tenía una granja, Ee-igh, Ee-igh, ¡Oh! Y en esa granja tenía perros , Ee-igh, Ee-igh, ¡Oh! Con un guau, guau aquí y un guau, guau allí. Aquí un guau, hay un guau, en todas partes un guau,guau .

El viejo MacDonald tenía una granja, Ee-igh, Ee-igh, ¡Oh! Y en esa granja tenía gallinas , Ee-igh, Ee-igh, ¡Oh! Con un co-co, co-co aquí y un co-co, co-co allí. Aquí un co-co, hay un co-co, en todas partes un co-co,co-co .

El viejo MacDonald tenía una granja, Ee-igh, Ee-igh, ¡Oh! Y en esa granja tenía patos , Ee-igh, Ee-igh, ¡Oh! Con un Cuak, Cuak aquí y un Cuak, Cuak allí. Aquí un Cuak, hay un Cuak, en todas partes un Cuak,Cuak .

3.

```
[27]: import math
def SpheraArea(r):
```

```
pi= math.pi
area= 4 * pi * (r**2)
return area
SpheraArea(6)
```

[27]: 452.3893421169302

```
[26]: def SpheraVolume(r):
      pi= math.pi
      Volumen= (4 * pi * (r**3))/3
      return Volumen
      SpheraVolume(6)
```

[26]: 904.7786842338604

4.

```
[34]: def sumN(n):
      suma=0
      for i in range(1,n+1):
          suma=suma+i
      return suma
      def sumNCubes(n):
          suma=0
          for i in range(1,n+1):
              suma = suma + i**2
          return suma
```

```
[38]: n=int(input("Escriba un numero: "))
      sumN(n)
```

Escriba un numero: 4

[38]: 10

```
[39]: sumNCubes(n)
```

[39]: 30

11.

```
[41]: def SquareEach(nums):
      resultado=[]
      for i in nums:
          resultado.append(i**2)
      return resultado
      SquareEach([1,2,3,4])
```

[41]: [1, 4, 9, 16]

12.

```
[42]: def sumList(nums):  
        resultado=0  
        for i in nums:  
            resultado = i + resultado  
        return resultado  
sumList([1,2,3,4])
```

[42]: 10

13.

```
[46]: def toNumbers(strList):  
        for i in range(len(strList)):  
            strList[i]= i  
        return strList  
toNumbers(["hola","como","estas"])
```

[46]: [0, 1, 2]
