

Taller de Programación

Certamen 1

13 de abril de 2019

Instrucciones:

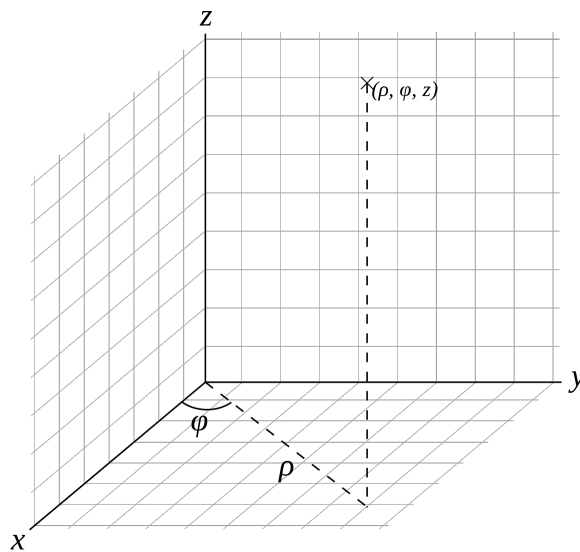
- Lea atentamente el enunciado de cada uno de los problemas.
- Elija solo **TRES** de los 4 problemas del certamen.
- Para cada problema cree un archivo .py distinto. El nombre del archivo debe ser el número del problema. Por ejemplo: uno.py, dos.py, tres.py o cuatro.py
- Suba sus respuestas como un archivo **ZIP** a la sección Evaluación en <http://canvas.udd.cl>. Solo tendrá una oportunidad para subir sus respuestas.
- Recuerde que usaremos un software de detección de plagio, confiamos en su honestidad.

(2 pts) 1. Coordenadas cilíndricas

Un punto en el espacio puede ser representado por conjunto de valores usando diferentes sistemas de coordenadas.

Escriba un programa que solicite tres números que representen las coordenadas cilíndricas (ρ, φ, z) , donde φ es un ángulo en **grados**, e imprima los valores de dicho punto en el sistema de coordenadas cartesianas en \mathbb{R}^3 . El programa debe guardar las coordenadas recibidas, e imprimir las coordenadas de salida en números **reales**.

Para convertir las coordenadas cilíndricas a cartesianas, utilice las expresiones de transformación: $x = \rho \cos \varphi$, $y = \rho \sin \varphi$, $z = z$ y las funciones trigonométricas $\sin()$, $\cos()$ y π de la librería `math` que reciben **radianes**. Recuerde que $2\pi = 360^\circ$.



Solución posible:

```
# Esta linea import sin, cos y pi desde la biblioteca math
from math import sin, cos, pi
rho = float(input('Ingrese rho: '))
phi = float(input('Ingrese phi: '))
z = float(input('Ingrese z: '))

phi = pi * phi/180
print(rho*cos(phi), rho*sin(phi), z)
```

- 0.5pts convierte input a float
- 0.5pts convierte a radianes
- 0.5pts utiliza las transformaciones correctamente
- 0.5pts imprime las coordenadas transformadas correctamente

(2 pts) 2. Calculando π

Escriba un programa en Python que estime el valor de π usando la siguiente sumatoria infinita:

$$\pi = 4 \times \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots \pm s_k \dots\right)$$

El programa debe entregar el valor de π cuando encuentre un término s_k de la sumatoria cuyo valor absoluto sea menor a 10^{-5} , es decir, $|s_k| < 10^{-5}$.

Solución posible:

```
s = 0
k = 0
resp = True
while resp==True:
    term = (-1)**(k) / (2 * k + 1)
    s += term
    k = k + 1
    if abs(term) < 0.00001:
        resp=False
print(4 * s)
```

- 0.5 crea variables necesarias para resolver el problema (acumulador, calculo del k-ésimo término, etc...)
- 0.8 usa correctamente el ciclo while (o un ciclo for, con un rango muy grande)
- 0.5 agrega condición de término del ciclo while (o for) cuando el término actual es menor a 10^{-5} .
- 0.2 imprime la aproximación de π

(2 pts) 3. Dígito verificador

Para prevenir errores de digitación es frecuente el uso de un dígito verificador. Un caso puntual es el dígito verificador módulo 11, utilizado en la cédula nacional de identidad, el cual se calcula usando los siguientes pasos:

- Se multiplican los dígitos del RUT por la serie 2, 3, 4, 5, 6 y 7, y se suman los productos. Esta operación se realiza de derecha a izquierda, y cuando quedan dígitos del rut por multiplicar se utilizan nuevamente los elementos de la serie 2, 3, 4, 5, 6, y 7.
- Una vez obtenida la suma total de los productos, esta se divide por 11 y se calcula el resto de esa división.
- Finalmente el dígito verificador se obtiene restando a 11 el resto de la división obtenida en el punto anterior.
 - Cuando el resultado del dígito verificador es igual a 10, al dígito verificador se le asigna la letra K.
 - Y si el dígito verificador es igual a 11 se le asigna el número 0.

Por ejemplo, el rut 30686957 posee el dígito verificador 4, esto es porque:

$$7 \times 2 = 14$$

$$5 \times 3 = 15$$

$$9 \times 4 = 36$$

$$6 \times 5 = 30$$

$$8 \times 6 = 48$$

$$6 \times 7 = 42$$

$$0 \times 2 = 0$$

$$3 \times 3 = 9$$

, la suma de los productos $14 + 15 + 36 + 30 + 48 + 42 + 0 + 9 = 194$, el resto de la división de $194 \% 11 = 7$, y la resta de $11 - 7 = 4$.

Realice un programa en Python que valide si el RUT de una persona fue ingresado correctamente. Considere que:

- El RUT se ingresará por teclado.
- El formato de ingreso del RUT es sin puntos, pero con guion, por ejemplo: 12345678-9
- Su programa debe funcionar con cualquier RUT, no solo el del ejemplo. ¡Pruebe con el suyo!

Solución posible:

```
r = input("Ingrese su rut: (Ej: 12345678-9) ")
x = []
```

```

for c in r[:-2]:
    x.append(int(c))

mult = [3,2,7,6,5,4,3,2]
suma = 0
for i in range(1,len(x)+1):
    suma += (x[-i])*(mult[-i])

d = 11-suma%11

if d == 10 and (r[-1]=="k" or r[-1]=="K"):
    print("El Rut esta correcto!!")
elif d==11 and r[-1]==0:
    print("El Rut esta correcto!!")
elif d==int(r[-1]):
    print("El Rut esta correcto!!")
else:
    print("El Rut NO esta correcto!!")

```

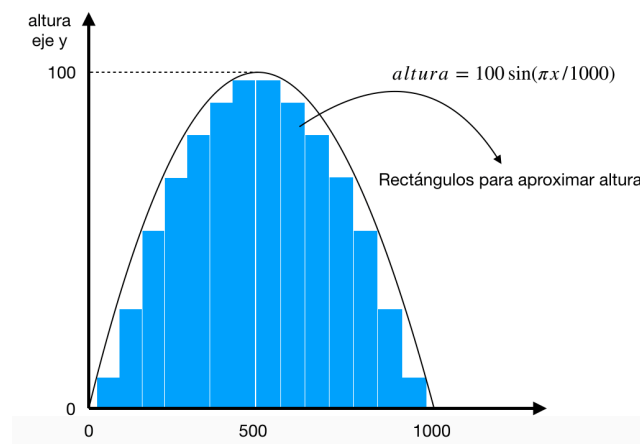
- 0.3 usar input para ingresar el RUT
- 0.7 por calcular la suma de los productos correctamente (podía hacerlo a mano o usando un ciclo)
- 0.3 calcular correctamente el dígito verificador
- 0.7 por validar el rut correctamente

Nota: la solución se aceptará como correcta aún cuando sólo funcione con un RUT de 8 dígitos.

(2 pts) 4. A Valparaíso en tren

Imagine por un momento que ya está en el último año de la carrera, y que por su buen desempeño en el área de tecnologías de información, ha sido invitada a participar en la construcción de la Línea de Trenes Santiago-Valparaíso. La empresa contratada para hacer el estudio es un poco anticuada, y siempre realiza los cálculos manualmente, sin utilizar ningún software. Su primera labor como Ingeniera en Práctica es estimar el costo para hacer el túnel. Para esto, debe desarrollar un programa que calcule el área del perfil del cerro donde se construirá el túnel que pasa por la Cuesta Lo Prado. Cálculos anteriores indican que la forma del cerro se puede aproximar usando $altura = 100 \sin(\pi x / 1000)$, donde x es la posición en la base del cerro y está en metros.

Para calcular el área de la sección del cerro, utilice la estrategia de suma de rectángulos de ancho 1 bajo la curva que describe el perfil del cerro.



Para este problema usted debe calcular el área del perfil del cerro sumando el área de los 1000 rectángulos bajo la curva.

Solución posible:

```
# esta linea import sin y pi desde la biblioteca math
from math import sin, pi

area = 0
for i in range(1000):
    y = 100 * sin(pi * i / 1000)
    area = area + y

print(area)
```

- 0.3 inicializar la variable acumuladora con cero
- 0.5 utilizar un ciclo for (o while) con las condiciones de inicio y fin correctas (para mil ciclos)
- 0.5 por calcular correctamente el área para cada rectángulo

- 0.5 actualizar la variable del área total por cada ciclo
- 0.2 por imprimir el área total