

Actividades de la Semana 9

Los alumnos del curso de “Programación para la Física y Astronomía” tienen 8 horas consideradas como trabajo personal establecidos en el programa de la asignatura.

Introducción

La sintaxis de un ciclo for es simple en C++, en realidad en la mayoría de los lenguajes de alto nivel es incluso muy similar, de hecho, con tan solo tener bien claros los 3 componentes del ciclo for (inicio, final y tamaño de paso) tenemos prácticamente todo hecho

<pre>for (int i = valor inicial; i <= valor final; i = i + paso) { Bloque de Instrucciones }</pre>	<pre>int i; for (i = 0; i < 2; i++) { cout << i; }</pre>
---	---

El inicio el ciclo for está asociado con una variable de tipo int que se asocia con el índice de una colección de datos ordenados por el índice, la estructura anterior es usando en algunas situaciones y debe ser conocida, también veremos las bondades del for en Python.

Introducción al Problema 01

Una técnica común utilizada por los desarrolladores de Python más nuevos para obtener el índice correspondiente cuando se trata de iterables es usar el `range(len(...))` o establecer e incrementar un contador.

<pre>semana = ["lunes", "martes", "miércoles", "jueves", "viernes", "sábado", "domingo"] for i in range(len(semana)): print("Dia {}: {}".format(i+1, dias[i]))</pre>
--

<pre>texto='Programando en Python' for i in range(len(texto)): print(texto[i])</pre>
--

Los elementos individuales de un arreglo en dos dimensiones se identifican con dos índices, para tener acceso a dichos elementos se requieren dos ciclos for anidados. La importación de la librería numpy será asignándole el nombre np en la importación, nos permitirá tener acceso a todas las funciones de la librería numpy sin modificar las funciones por defecto que carga Python.

<pre>import numpy as np a=np.array([[1, 2, 5], [7, 3, 2], [9, 7, 6], [3, 15, 11], [4, 6, 8]]) suma=0 for i in range(len(a)): for j in range(len(a[0])): suma+=a[i,j] print(suma)</pre>	<pre>import numpy as np a=np.array([[1, 2, 5], [7, 3, 2], [9, 7, 6], [3, 15, 11], [4, 6, 8]]) suma=0 for i in range(a.shape[0]): for j in range(a.shape[1]): suma+=a[i,j] print(suma)</pre>
---	--

Para la creación de matrices usando las funciones disponibles en numpy, establece la definición del tipo de datos, pero también podemos crear matrices con datos de tipo booleano

<pre>import numpy as np matriz_ceros = np.zeros((2,3), int) matriz_unos= np.ones((2,3), int) print(matriz_ceros) print(matriz_unos)</pre>	<pre>import numpy as np matriz_falso = np.zeros((2,3), bool) matriz_verdadero= np.ones((2,3), bool) print(matriz_falso) print(matriz_verdadero)</pre>
---	---

La matriz B se usará para resolver varios problemas

<pre>import numpy as np B=np.array([[2,6,2,14,10],[3,2,10,10,15],[8,6,9,5,2],[5,7,10,8,8],[8,2,14,13,15],[14,6,10,7,5],[7,2,8,11,3],[4,14,6,4,2],[5,10,15,12,2],[15,13,1,8,12],[11,14,3,4,9],[14,14,7,9,2],[5,5,7,7,5]])</pre>
--

La matriz D es la colección de datos

<pre>import numpy as np D=np.array([[12.2,110.1],[14.2,112.3],[15.8,117.3],[17.8,122.8],[19.8,127.3],[22.2,139],[23.8,140.4],[25.8,139],[28.3,147.8],[29.8,156.5],[32.3,160.7],[34.3,163],[36.2,167.6],[38.2,177.3],[39.8,177],[41.8,185.1],[44.2,190.4],[45.7,195.4],[47.8,197],[49.8,202.9],[52.3,207.3],[54.2,217.3],[56.3,217.3],[57.8,227.3],[60.2,228],[61.8,235.1],[63.8,237.3],[66.3,246.5],[67.8,246.5],[70.3,255.4],[72.2,260.7],[73.8,265.4],[76.3,272.3],[77.8,276.5],[79.8,280],[82.2,282],[83.8,284],[85.8,293],[87.8,297.6],[90.2,301.5],[92.2,304],[93.8,312],[96.3,317.8],[98.3,325],[100.3,330.1],[102.2,335.4]])</pre>

Los siguientes problemas desde 01 al 05 deben ser usando el ciclo for pero usando los índices establecidos en la introducción del problema 01

Problema 01

Obtenga la suma de todos los elementos cada fila y de cada columna de la matriz B, almacene sus resultados en dos vectores independientes, imprimir los resultados de los vectores. (use como ejemplo la matriz de la derecha, las celdas en celeste son los resultados esperados)

1	2	5	8
7	3	2	12
9	7	6	22
3	15	11	29
4	6	8	18
24	33	32	

Problema 02

Obtenga los valores máximos y mínimos de todos los elementos cada fila y de cada columna de la matriz B, almacene sus resultados en 4 vectores independientes, imprimir los resultados de los vectores.

Problema 03

Obtenga la suma de todos los elementos cada diagonal de la matriz B, almacene sus resultados en un vector independientes y muestre sus resultados. (use como ejemplo la matriz de la derecha, las celdas en color celeste son los resultados esperados)

1	2	5	
7	3	2	
9	7	6	
3	15	11	10
4	6	8	25
			32

Problema 04

Cuente y sume los elementos de la matriz B que son igual a 5 o 10.

Problema 05

Cree una matriz de las dimensiones B que almacene valores booleanos, donde los valores sean verdaderos si el elemento de la matriz B es mayor que 8, de lo contrario es falso.

Problema 06

Obtenga los valores máximos y mínimos de todos los elementos cada fila y de cada columna de la matriz B, almacene sus resultados en 4 vectores independientes, imprimir los resultados de los vectores.

Problema 07

Usando la matriz B identifique los elementos que tiene un valor igual a 2 y cambie ese valor por la suma del valor que tiene arriba (celda de la fila anterior) y abajo (celda de la fila siguiente). El caso especial es en la primera fila y la última donde no existe fila anterior o siguiente, en ese caso tome el valor buscado.

Problema 08

Usando la matriz D, calcule los valores de m y n de las siguientes expresiones. Donde x_i representa los elementos de la primera columna y y_i son los elementos de la segunda columna. $k=N$ es el número de elementos de la primera columna.

$m = \frac{k \sum (x_i y_i) - \sum x_i \sum y_i}{k \sum x_i^2 - (\sum x_i)^2}$	$n = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum (x_i y_i)}{k \sum x_i^2 - (\sum x_i)^2} = \frac{\sum y_i - m \sum x_i}{k}$
--	---

Donde

$\sum_{i=1}^{i=N} x_i = x_1 + x_2 + \dots + x_N$	$\sum_{i=1}^{i=N} x_i^2 = x_1^2 + x_2^2 + \dots + x_N^2$
$\sum_{i=1}^{i=N} x_i \cdot y_i = x_1 \cdot y_1 + x_2 \cdot y_2 + \dots + x_N \cdot y_N$	

Introducción al Problema 09

Los usuarios de Python no están obligados a usar los índices para recorrer una colección de objetos, los códigos siguientes no se utiliza un índice entero para numerar la colección de datos ordenados, el ciclo for asigna a la variable definida cada uno de los elementos de la colección de datos. Los ejemplos anteriores fueron reescritos.

```
semana = ["lunes", "martes", "miércoles", "jueves", "viernes", "sábado", "domingo"]
for dia in semana:
    print("Dia: {}".format(dia))
```

```
texto='Programando en Python'
for letra in texto:
    print(letra)
```

```
import numpy as np
a=np.array([[ 1, 2, 5],
            [ 7, 3, 2],
            [ 9, 7, 6],
            [ 3, 15, 11],
            [ 4, 6, 8]])
suma=0
for fila in a: # fila es una fila de M
```

```
for celda in fila:
    suma+=celda
print(suma)
```

Problema 09

Usando los conocimientos de la introducción de la pregunta 09, resuelva los problemas 01, 02, 04 y 08

Introducción al Problema 10

La librería numpy incorpora varias funciones que realizan los cálculos realizados en los problemas anteriores, pero los estudiantes que inician la programación deben ser capaces de resolver los problemas usando el ciclo for, ese trabajo personal los prepara para crear las soluciones a problemas que no tiene un código de programación escrito.

Revise los siguientes códigos.

Sumar matrices y

```
import numpy as np
B=np.array([[2,6,2,14,10],[3,2,10,10,15],[8,6,9,5,2]])
print(B)
print(B.sum(), np.sum(B))
print(B.sum(axis=0), B.sum(0))# suma columnas
print(B.sum(axis=1), B.sum(1))# suma filas
print(B[0:1,3:5].sum(1), np.sum(B[0:1,3:5]))
```

Combinando matrices

```
v1 = np.array([1, 2, 3])
v2 = np.array([4, 5, 6])
v3 = np.array([7, 8, 9])
M = np.concatenate((v1, v2, v3))
print(M)
```

Apilar, horizontal y verticalmente

```
M = np.vstack((v1, v2, v3))
print(M)
```

```
M = np.hstack((v1, v2, v3)) # El mismo resultado para este ejemplo que concatenate
print(M)
```

Mínimo y máximo

```
import numpy as np
B=np.array([[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15]])
print(B)
print(B.max(), np.max(B))
print(B.min(), np.min(B))
print(B.max(axis=0), B.max(0))# maximo por columnas
print(B.max(axis=1), B.max(1))# maximo por filas
print(B.min(axis=0), B.min(0))# minimo por columnas
print(B.min(axis=1), B.min(1))# minimo por filas
print(B[0:1,3:5].min(1), np.min(B[0:1,3:5]))
print(B[:, 2].min(), B[:, 2].max())
```

Seleccionar elementos en base a una condición en Numpy

Al aplicar un operador de igualdad o comparación a un objeto Numpy se obtiene un nuevo vector o matriz de valores booleanos, que puede ser utilizado en otra matriz.

```
import numpy as np
B=np.array([[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15]])
print(B)
print(B>=10)
print(B[B>=10])
print(B[(B>=3) & (B<7)])# & es y lógico
print(B[(B<3) | (B>=7)])# | es o lógico
print(B[~((B>=3) & (B<7))])# ~ es negación lógico
```

Problema 10

Usando los conocimientos de la introducción de la pregunta 10, resuelva los problemas 02, 04, 05 y 06