




Fund Programacion G28-29-30

-  Descripción
-  Editar
-  Ver entrega

Reto 1 - Semana 3 - Variante 3

Ficheros requeridos: punto07.py ([Descargar](#))

Tipo de trabajo: Individual

El siguiente código ha perdido algunas de sus líneas. Su trabajo es completar las líneas de código que faltan, las cuales están marcadas claramente con textos que describen lo que la línea debe hacer. Estas líneas faltantes son de color verde y empiezan por #.

Todo el programa realiza las siguientes acciones:

Se genera un entero entre 15 y 25; luego se construye un objeto de la clase vector (esta clase ya existe) tal como la definida en el curso. El tamaño del vector es el número generado inicialmente. Luego, se llena el vector con números enteros entre 1 y 99 generados aleatoriamente.

Con el vector creado se construye un nuevo vector, el cual debe contener el máximo común divisor y el mínimo común múltiplo de: el primer número y el segundo, el tercero y el cuarto, el quinto y el sexto y así sucesivamente.

Por ejemplo, si en el vector a procesar en las dos primeras posiciones están los números (posición 1)36 y (Posición 2) 48 entonces en las dos primeras posiciones del vector construido estarán 12(Máximo común divisor de 36 y 48) y 144(mínimo común múltiplo de 36 y 48). Si el tamaño del vector es impar, el número que se quede sin pareja no será tomado en cuenta.

Por el ejemplo:

El vector inicial se puede ver así: 91, 58, 12, 21, 19, 2, 54, 88, 73, 32, 73, 75, 74, 57, 44,

Y el vector resultante así: 1, 5278, 3, 84, 1, 38, 2, 2376, 1, 2336, 1, 5475, 1, 4218,

Ficheros requeridos

punto07.py



```
5 """Función que contiene la solución al problema que será calificado"""
6
7
8 def solucion():
9     # Completa la siguiente línea para generar un número entero aleatorio entre 15 y 25.
10    a = # Sugerencia, usa random.randint
11
12    """Creación del objeto vector con tamaño a"""
13    v = vector(a)
14
15    """Llenar el vector con números enteros aleatorios entre 1 y 99.
16    Recuerde que en el curso se definió que se debe llenar desde
17    la posición 1 en adelante, pues la posición cero guarda el número
18    de casillas ocupadas en el vector con números diferentes de cero"""
19    for i in range(1, a + 1):
20        # Completa el llenado de cada casilla, el número debe ser entero y
21        # aleatorio entre 1 y 99.
22        v.V[i] = # Sugerencia, usa random.randint
23
24    """Como el número es aleatorio entre 1 y 9999, habrá UNA (1) nueva casilla
25    ocupada, por lo tanto, se debe ir alterando en UNO (1) la posición 0 del vector
26    cada vez que se llene una casilla"""
27    v.V[0] += 1
28
29    """Salimos del ciclo de llenado y procedemos a crear un nuevo objeto vector con el mismo tamaño a"""
30    """Creación del nuevo objeto vector con tamaño a"""
31    vres = vector(a)
32
33    """Recorramos todas las casilla del vector desde la posición inicial (1) hasta la penúltima posición (a).
34    En este caso añadimos un tercer parámetro, que indicará la diferencia entre los números del rango,
35    en este caso, será de dos en dos.
36    Ejemplos: recorremos range(1, 6, 1), que es la secuencia de números (1,2,3,4,5) con un valor de incremento 1, que resultará en la secuencia
37    recorremos range(1, 6, 2), que es la secuencia de números (1,2,3,4,5) con un valor de incremento 2, que resultará en la secuencia
38    recorremos range(1, 6, 4), que es la secuencia de números (1,2,3,4,5) con un valor de incremento 4, que resultará en la secuencia
39    recorremos range(1, 6, 6), que es la secuencia de números (1,2,3,4,5) con un valor de incremento 6, que resultará en la secuencia
40    for i in range(1, a, 2):
41        # Completa la siguiente línea para obtener el máximo común divisor de la posición actual y la posición siguiente.
42        valormcd = # Sugerencia, usa math.gcd
43
44    """Almacenamos y hallamos el mínimo común múltiplo de la posición actual y la posición siguiente del vector"""
45    # Vamos a completar la función mcm, (vaya a la línea 80)
46    valormcm = mcm(v.V[i], v.V[i + 1])
47
48    """Agregamos el máximo común divisor obtenido al nuevo vector con el método agregarDato"""
49    agregarDato(vres, valormcd)
50
51    """Agregamos el mínimo común múltiplo obtenido al nuevo vector con el método agregarDato"""
52    agregarDato(vres, valormcm)
53
54    """El ejercicio ha terminado, presiona en evaluar :D"""
55    return v, vres
56
57
58 def agregarDato(vector, d):
59     """ Esta función agrega un valor al vector que tiene como primer parámetro"""
60
61     """Pregunta si el vector está lleno; si lo está, NO agregará ningún dato y 'retornará' nada"""
62     if esLleno(vector):
63         return
64
65     """Como se agregará un valor 'd', habrá UNA (1) nueva casilla
66     ocupada, por lo tanto, se debe ir alterando en UNO (1) la posición 0 del vector
67     cada vez que se agregue un valor 'd'"""
68     vector.V[0] = vector.V[0] + 1
69
70     """Finalmente, agregamos el valor 'd' a la última posición del vector"""
71     vector.V[vector.V[0]] = d
72
73
74 def esLleno(vector):
75     """ Esta función retorna True si el vector tiene todas las casillas ocupadas
76     De lo contrario, retorna False"""
77     return vector.V[0] == vector.n
78
79
80 def mcm(x, y):
81     """ Esta función retorna el valor del mínimo común múltiplo de dos números"""
82     # Completa la siguiente línea para obtener el mínimo común múltiplo de dos números
83     # Sugerencia, la fórmula para obtener el mínimo común múltiplo consiste en
84     # multiplicar los dos números y dividirlos por su máximo común divisor
85     # Ejemplo: el mcm entre 8 y 12 es 24, teniendo en cuenta que 8x12=96 y su máximo común divisor es 4.
86     valor = # Sugerencia, usa math.gcd
87     return valor
88
89
90 """Este procedimiento permite imprimir vectores en la consola"""
91
92
93 def imprimeVector(vector, mensaje="vector sin nombre: \t"):
94     print("\n", mensaje, end=" ")
95     for i in range(1, vector.V[0] + 1):
96         print(vector.V[i], end=" ")
97         if i % 30 == 0:
98             print("\n", end=" ")
99     print()
100
101
102 """Las siguientes líneas le permitirán probar su solución al presionar el botón de ejecutar"""
103 a, b = solucion()
104 imprimeVector(a, 'Original')
105 imprimeVector(b, 'Modificado')
```

[Servicio Web](#)

[VPL](#)

