

Parcial II Término 2003-2004. Diciembre 09, 2003

Tema 2 (20 puntos) Escriba un algoritmo que muestre por pantalla el resultado de la suma **S** de los términos de una progresión geométrica, de primer término **a** y razón **r**, con valores de **i** desde 0 hasta **n**.

La fórmula es:

El Programa debe solicitar al usuario los valores de las variables **a**, **n** y **r**, y validar que **r** es diferente de 1.

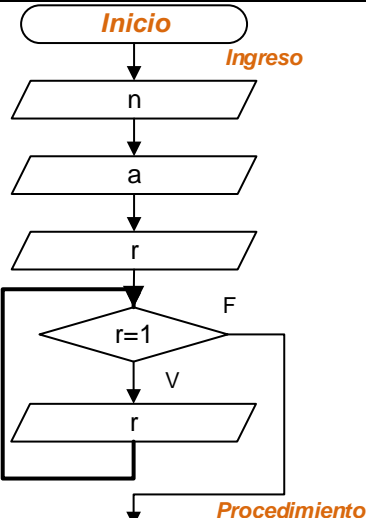
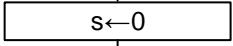
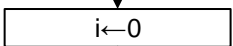
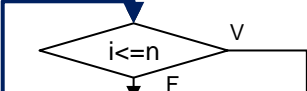
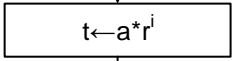
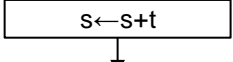
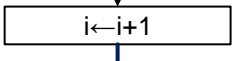
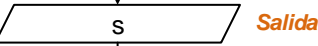

$$s = \sum_{i=0}^n ar^i = a + ar + ar^2 + ar^3 + \dots + ar^n$$

Rubrica: Ingreso y validación de datos (5 puntos), calculo de términos (5 puntos). Suma serie (5 puntos). Algoritmo integral (5 puntos)

Propuesta de Solución:

Ingresar la cantidad **n** de términos, el factor **a** y **r**, validando que **r** no sea 1. Iniciar con 0 las variables: **s** que acumula términos, **i** como el contador de términos que adicionalmente se usa como exponente. Como paso siguiente, calcular cada término y acumularlos en **s** tantas veces como sea necesario confirme a **n**. El resultado buscado se encontrará en **s**. Observe que el primer término es solo la constante **a**, con equivalente ar^0 .

Se propone usar las formas de lazo "mientras-repita" y "repita-hasta".

Descripción	DIAGRAMA DE FLUJO	Python
Inicio		# ICM00794-Fund. de Computación - FCNM-ESPOL # Parcial II Termin 2003. Tema 2 # Sumar términos de progresión geométrica # Propuesta de solución. edelros@espol.edu.ec. n=int(input('¿Valor de n?: ')) a=float(input('factor a: ')) r=float(input('factor r: '))
Ingresar el numero de términos		
Ingresar el factor a		
Ingresar el factor r		
Mientras r==1		while (r==1):
Ingresar el factor r		r=int(input('factor r debe ser diferente de 1: '))
Repita		
Suma de Serie s inicia con 0		s=0
Término i inicia con 0		i=0
Mientras i es menor o igual que n		while (i<=n):
Calcula términos t		t=a*(r**i)
Acumula términos en s		s=s+t
Siguiente término i		i=i+1
Repita		
Mostrar Resultado s		print(s)
Fin		

Ejecución del algoritmo: seriegeom.py

<pre>>>> ¿Valor de n?: 5 factor a: 0.5 factor r: 2 31.5 >>></pre>	<pre>>>> ¿Valor de n?: 3 factor a: 1 factor r: 2 15.0 >>></pre>
---	---

Luego se muestra el algoritmo usando el lazo “Repita-Hasta”.

Observe que la solución se muestra más corta en los lazos de validación de r y en el cálculo de los términos.

Diagrama de Flujo	Python
<pre> graph TD Inicio([Inicio]) --> n[/n/] n --> a[/a/] a --> r[/r/] r --> r_neq_1{r ≠ 1} r_neq_1 -- F --> r r_neq_1 -- V --> s_0[s ← 0] s_0 --> i_0[i ← 0] i_0 --> t_calc[t ← a * r^i] t_calc --> s_plus_t[s ← s + t] s_plus_t --> i_plus_1[i ← i + 1] i_plus_1 --> i_ge_n{i ≥ n} i_ge_n -- F --> t_calc i_ge_n -- V --> s_out[/s/] s_out --> Fin([Fin]) </pre>	<pre> # ICM00794-Fundamentos de Computación - FCNM-ESPOL # Parcial II Terminó 2003. Tema 2 # Sumar términos de progresión geométrica # Propuesta de solución. edelros@espol.edu.ec. n=int(input('¿Valor de n?: ')) a=float(input('factor a: ')) r=float(input('factor r: ')) # debe repetirse la pregunta while not(r!=1): r=int(input('factor r debe ser diferente de 1: ')) s=0 i=0 while not(i>n): t=a*(r**i) s=s+t i=i+1 print(s) </pre>

Ejecución del algoritmo: seriegeom_do.py

```

>>>
¿Valor de n?: 3
factor a: 1
factor r: 1
factor r debe ser diferente de 1: 2
15.0
>>>
        
```