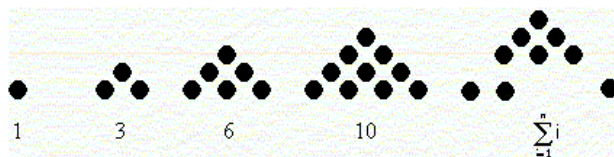


Parcial III Término 2003-2004. Abril 02, 2004

Tema 2. (25 puntos) Considere la secuencia de números triangulares 1, 3, 6, 10,... cuyo nombre refleja su ley de formación:

Escriba un algoritmo en pseudocódigo que indique si un número natural t , ingresado por teclado, es triangular, esto es, si es de la forma:

$$t = \sum_{i=1}^n i \quad \text{para algún número natural } n.$$



Rúbrica: Uso de acumuladores y contadores (5 puntos), lazo para calculo de serie (5 puntos), selección de respuesta apropiada (10 puntos). Solución integral y estructurada (5 puntos)

Propuesta de Solución:

El problema planteado es similar a la construcción de una pirámide, en la que se disponen de solo t bloques y se requiere saber si el número de bloques es exacto para formar una pirámide. Se observa que el número de bloques coincide con el número de piso a construir. Ejemplo: Piso 1 tiene 1 bloque, piso 2 tiene 2 bloques, etc. Dado un número t de bloques, se calcula la secuencia de números triangulares mientras los bloques usados sean menores que los t disponibles. En caso que el número “usados” de la secuencia es igual a t , se considera al número t como un número triangular.

Descripción	DIAGRAMA DE FLUJO	Python
<p>Inicio</p> <p>Ingresar el numero t</p> <p>El número de piso indica el número de bloques para construir el piso</p> <p>Al inicio usados es cero</p> <p>Mientras los bloques usados sean menores que los t disponibles</p> <p> acumule los bloques usados para construir otro piso</p> <p> siguiente piso,</p> <p>repita</p> <p>Si los bloques usados = t</p> <p>El numero t es triangular, representado por $z=1$</p> <p>Si no, $z=0$</p> <p>Mostrar resultado de z</p>	<pre> graph TD Inicio([Inicio]) -- Ingreso --> t[/t/] t -- Procedimiento --> Piso[piso ← 1] Piso --> Usados[usados ← 0] Usados --> Cond1{usados < t} Cond1 -- V --> Suma[usados ← usados + piso] Suma --> PisoInc[piso ← piso + 1] PisoInc --> Cond1 Cond1 -- F --> Cond2{usados = t} Cond2 -- V --> Z1[z ← 1] Cond2 -- F --> Z0[z ← 0] Z1 --> Salida[/z/] Z0 --> Salida Salida -- Salida --> Fin([Fin]) </pre> <p>Mientas Repita</p>	<pre> # ICM00794-Fund. de Computación - FCNM-ESPOL # Parcial III Término 2003.Tema 2. # Números triangulares # Propuesta de solución. edelros@espol.edu.ec. t=int(input("Verificar si es triangular: ")) piso=1 usados=0 while (usados<t): usados=usados+piso piso=piso+1 if usados==t: z=1 else: z=0 print(z) </pre>

Ejecución del algoritmo: triangular.py

<pre> >>> Verificar si es triangular: 10 1 >>> </pre>	<pre> >>> Verificar si es triangular: 11 0 >>> </pre>
---	---

Una forma alterna de solución usando el lazo “Repita- Hasta” se muestra en el siguiente ejemplo.

Observe que solo cambia la forma del lazo para repetir.

Ésta forma de lazo no es soportada por matlab, sin embargo en otros lenguajes se puede escribir como “Repita – Hasta”.

Diagrama de Flujo	Python
<pre> graph TD Inicio([Inicio]) --> Ingreso[/t/] Ingreso --> Piso[Piso=bloques] Piso --> p1[piso <- 1] p1 --> u1[usados <- 0] u1 --> u2[usados <- usados + piso] u2 --> p2[piso <- piso + 1] p2 --> D1{usados >= t} D1 -- F --> u2 D1 -- V --> D2{usados == t} D2 -- F --> z0[z <- 0] D2 -- V --> z1[z <- 1] z0 --> Salida[/z/] z1 --> Salida Salida --> Fin([Fin]) </pre>	<pre> # ICM00794-Fundamentos de Computación - FCNM-ESPOL # Parcial III Término 2003.Tema 2. Números triangulares # Propuesta de solución. edelros@espol.edu.ec. t=int(input('Verificar si es triangular: ')) piso=1 usados=0 while not(usados>=t): usados=usados+piso piso=piso+1 if usados==t: z=1 else: z=0 print(z) </pre>