

Primera entrega algoritmos y estructura de datos.

La serie de Padovan es una sucesión de números naturales definida por los valores iniciales $P(0)=P(1)=P(2)=1$, y la relación de recurrencia para $n \geq 3$:

$$P_n = P_{n-2} + P_{n-3}$$

Implementar en pseudocódigo o en un lenguaje como Python una función que calcule el término n -ésimo de la serie de Padovan.

Los tres primeros términos de la serie son fijos:

$$P(0)=1$$

$$P(1)=1$$

$$P(2)=1$$

La serie se calcula utilizando una relación de recurrencia, que define cómo obtener cada término posterior en función de los términos anteriores: $P(n)=P(n-2)+P(n-3)$

Cada término a partir del cuarto es la suma del término que está dos lugares antes y del que está tres lugares antes.

$$P(3)=P(1)+P(0)=1+1=2$$

$$P(4)=P(2)+P(1)=1+1=2$$

$$P(5)=P(3)+P(2)=2+1=3$$

$$P(6)=P(4)+P(3)=2+2=4$$

$$P(7)=P(5)+P(4)=3+2=5$$

$$P(8)=P(6)+P(5)=4+3=7$$

$$P(9)=P(7)+P(6)=5+4=9$$

$$P(10)=P(8)+P(7)=7+5=12$$

$$P(11)=P(9)+P(8)=9+7=16$$

Seudocódigo:

Inicializar $P(0) = 1$


```

1 #Solucion utilizando array
2 def padovan(N):
3     # Los tres primeros términos de la serie de Padovan son fijos: 1, 1, y 1
4     P = [1, 1, 1]
5
6     # Calculamos iterativamente desde el índice 3 hasta N (incluido)
7     for i in range(3, N):
8         P.append(P[i-2] + P[i-3]) # Cada nuevo término es la suma de los términos en i-2 y i-3
9     return P
10
11 while True:
12     try:
13         # Solicitamos al usuario que ingrese un número
14         N = int(input("Número de términos? "))
15
16         # Verificamos que el número sea positivo
17         if N <= 0:
18             print("Por favor, introduce un número entero positivo.")
19         elif N == 1:
20             print("Serie de Padovan:\n1")
21             break
22         elif N == 2:
23             print("Serie de Padovan:\n1 1")
24             break
25         elif N == 3:
26             print("Serie de Padovan:\n1 1 1")
27             break
28         else:
29             print("Serie de Padovan:")
30             P = padovan(N)
31             print(P)
32             break
33     # Si la entrada no es valida
34     except ValueError:
35         print("Entrada no válida. Por favor, introduce un número entero.")
36

```

```

1 def padovan(N):
2     # Los tres primeros términos de la serie de Padovan son fijos: 1, 1, y 1
3     a, b, c = 1, 1, 1
4
5     # Si N es 1, 2 o 3
6     if N >= 1:
7         print(a)
8     if N >= 2:
9         print(b)
10    if N >= 3:
11        print(c)
12
13    # Calculamos e imprimimos los términos de la serie desde el cuarto término.
14    for _ in range(3, N):
15        # El siguiente término es la suma de los términos a (P(n-3)) y b (P(n-2))
16        next_term = b + a
17        print(next_term)
18
19        # Actualizamos las variables para el siguiente ciclo
20        a, b, c = b, c, next_term
21
22 # Programa principal
23 while True:
24     try:
25         # Solicitamos al usuario que ingrese un número
26         N = int(input("Número de términos? "))
27
28         # Verificamos que el número sea positivo
29         if N <= 0:
30             print("Por favor, introduce un número entero positivo.")
31         elif N == 1:
32             print("Serie de Padovan:\n1")
33             break
34         elif N == 2:
35             print("Serie de Padovan:\n1\n1")
36             break
37         elif N == 3:
38             print("Serie de Padovan:\n1\n1\n1")
39             break
40         else:
41             print("Serie de Padovan:")
42             padovan(N)
43             break
44     # Si la entrada no es válida
45     except ValueError:
46         print("Entrada no válida. Por favor, introduce un número entero.")
47

```