

Solución – Parte 01



La primera solución propuesta por Kimi es incorrecta. Por este motivo, intentaremos encontrar la solución de otra forma apoyados por Kimi.

Análisis de lógica para matrices de orden PAR

Sea n orden par mayor que 0

si $n = 2$ generar el arreglo unidimensional de tamaño 2^2

$a[0]=1, a[1]=2,$ ningún cero $j=0, j=1$ 0
 $a[2]=3, a[3]=4$ ningún cero $j=0, j=1$ 0

Si $n=4$ generar el arreglo unidimensional de tamaño 4^4

$a[0]=1, a[1]=2, a[2]=3, a[3]=4,$ $j=0, j=3$ 0
 $a[4]=0, a[5]=5, a[6]=6, a[7]=0,$ $j=1, j=2$ 2
 $a[8]=0, a[9]=7, a[10]=8, a[11]=0$ $j=1, j=2$ 2
 $a[12]=9, a[13]=10, a[14]=11, a[15]=12$ $j=0, j=3$ 0

Si $n=6$ generar el arreglo unidimensional de tamaño 6^6

$a[0]=1, a[1]=2, a[2]=3, a[3]=4, a[4]=5, a[5]=6$ $j=0, j=5$ 0
 $a[6]=0, a[7]=7, a[8]=8, a[9]=9, a[10]=10, a[11]=0$ $j=1, j=4$ 2
 $a[12]=0, a[13]=0, a[14]=11, a[15]=12, a[16]=0, a[17]=0$ $j=2, j=3$ 4
 $a[18]=0, a[19]=0, a[20]=13, a[21]=14, a[22]=0, a[23]=0$ $j=2, j=3$ 4
 $a[24]=0, a[25]=15, a[26]=16, a[27]=17, a[28]=18, a[29]=0$ $j=1, j=4$ 2
 $a[30]=19, a[31]=20, a[32]=21, a[33]=22, a[34]=23, a[35]=24$ $j=0, j=5$ 0

Si $n=8$ generar el arreglo unidimensional de tamaño 8^8

$a[0]=1, a[1]=2, a[2]=3, a[3]=4, a[4]=5, a[5]=6, a[6]=7, a[7]=8$ $j=0, j=7$ 0
 $a[8]=0, a[9]=9, a[10]=10, a[11]=11, a[12]=12, a[13]=13, a[14]=14, a[15]=0$ $j=1, j=6$ 2
 $a[16]=0, a[17]=0, a[18]=15, a[19]=16, a[20]=17, a[21]=18, a[22]=0, a[23]=0$ $j=2, j=5$ 4
 $a[24]=0, a[25]=0, a[26]=0, a[27]=19, a[28]=20, a[29]=0, a[30]=0, a[31]=0$ $j=3, j=4$ 6
 $a[32]=0, a[33]=0, a[34]=0, a[35]=21, a[36]=22, a[37]=0, a[38]=0, a[39]=0$ $j=3, j=4$ 6
 $a[40]=0, a[41]=0, a[42]=23, a[43]=24, a[44]=25, a[45]=26, a[46]=0, a[47]=0$ $j=2, j=5$ 4
 $a[48]=0, a[49]=27, a[50]=28, a[51]=29, a[52]=30, a[53]=31, a[54]=32, a[55]=0$ $j=1, j=6$ 2
 $a[56]=33, a[57]=34, a[58]=35, a[59]=36, a[60]=37, a[61]=38, a[62]=39, a[63]=40$ $j=0, j=7$ 0

:

etc

INSTRUCTOR



Le solicitamos a Kimi: Construir código flowgorithm teniendo como entrada n siendo par y que genere lo siguiente:

Límites superior e inferior por cada renglón en donde NO SON cero

Para matriz orden n=2
limInf = 0 limSup=1
limInf = 0 limSup=1

Para matriz orden n=4
limInf=0 limSup=3
limInf=1 limSup=2
limInf=1 limSup=2
limInf=0 limSup=3

Para matriz orden n=6
limInf=0 limSup=5
limInf=1 limSup=4
limInf=2 limSup=3
limInf=2 limSup=3
limInf=1 limSup=4
limInf=0 limSup=5

Para matriz orden n=8
limInf=0 limSup=7
limInf=1 limSup=6
limInf=2 limSup=5
limInf=3 limSup=4
limInf=3 limSup=4
limInf=2 limSup=5
limInf=1 limSup=6
limInf=0 limSup=7

:

etc.



🔧 Generador de pares (limInf, limSup) para n par

Patrón:

- $n/2$ líneas
- Ascendente hasta la mitad, luego espejo.

Guarda como ParesLimite.fprg

xml

📄 复制

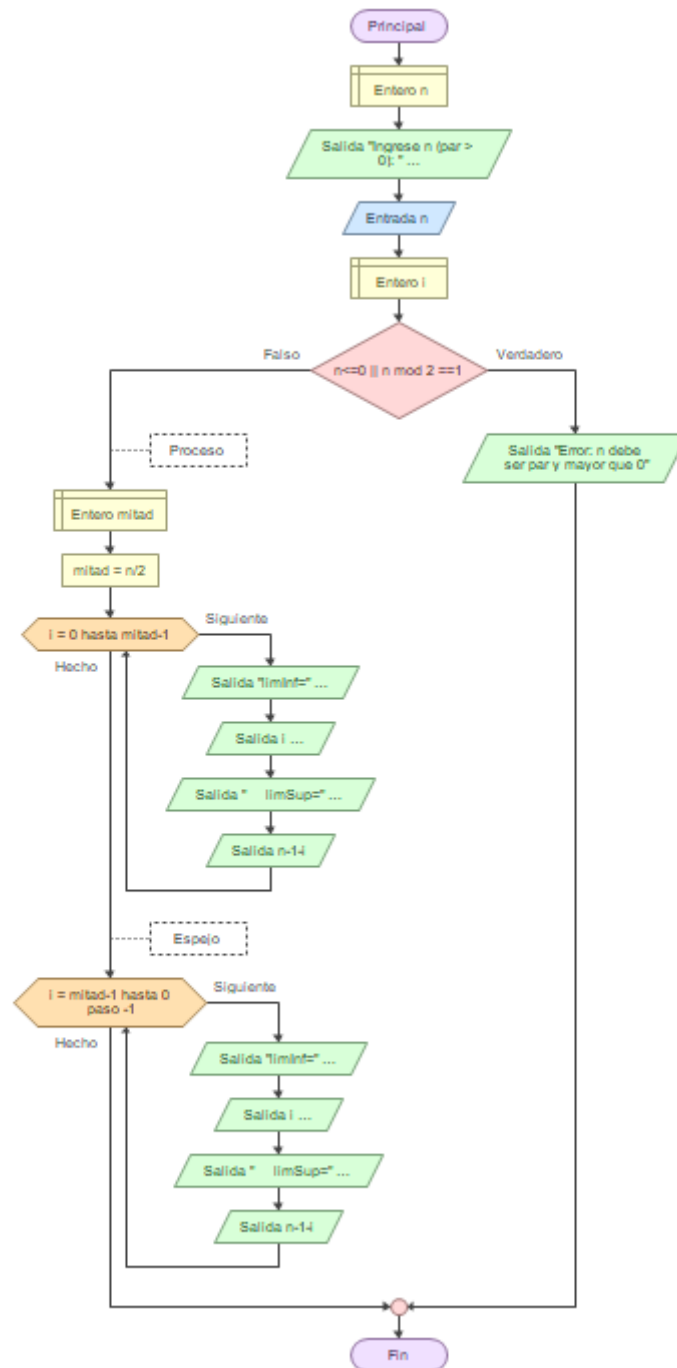
```
<!-- Entrada -->
<function name="Main" type="None">
  <parameter/>
  <body>
    <!-- Entrada -->
    <declare name="n" type="Integer" array="False"/>
    <output text="Ingrese n (par > 0): "/>
    <input variable="n"/>
    <if expression="n <= 0 or n mod 2 = 1">
      <then>
        <output text="Error: n debe ser par y mayor que 0" newline="True"/>
        <return/>
      </then>
    </if>

    <!-- Proceso -->
    <declare name="mitad" type="Integer" array="False"/>
    <assign variable="mitad" expression="n // 2"/>

    <for variable="i" start="0" end="mitad - 1">
      <output text="limInf=" newline="False"/>
      <output expression="i" newline="False"/>
      <output text="      limSup=" newline="False"/>
      <output expression="n - 1 - i" newline="True"/>
    </for>

    <!-- Espejo -->
    <for variable="i" start="mitad - 1" end="0" step="-1">
      <output text="limInf=" newline="False"/>
      <output expression="i" newline="False"/>
      <output text="      limSup=" newline="False"/>
      <output expression="n - 1 - i" newline="True"/>
    </for>
  </body>
</function>
</flowgorithm>
```





El resultado es correcto. Las corridas se muestran abajo

Ingrese n (par > 0):

2

limInf=0 limSup=1

limInf=0 limSup=1



Ingrese n (par > 0):

4

limInf=0 limSup=3

limInf=1 limSup=2

limInf=1 limSup=2

limInf=0 limSup=3

Ingrese n (par > 0):

limInf=0 limSup=5

limInf=1 limSup=4

limInf=2 limSup=3

limInf=2 limSup=3

limInf=1 limSup=4

limInf=0 limSup=5

6

Ejecutar

Ingrese n (par > 0):

limInf=0 limSup=7

limInf=1 limSup=6

limInf=2 limSup=5

limInf=3 limSup=4

limInf=3 limSup=4

limInf=2 limSup=5

limInf=1 limSup=6

limInf=0 limSup=7

8