

INSTRUCTOR



Rompecabeza01

a) Construir código flowgorithm para arreglos de dos dimensiones almacenados en un arreglo de una dimensión discriminando orden par e impar que almacene números enteros en la siguiente forma:

Matriz impares:

Orden 1

$a[0,0]=1$

Orden 3

$a[0,0]=1$ $a[0,1]=2$ $a[0,2]=3$

$a[1,0]=0$ $a[1,1]=4$ $a[1,2]=0$

$a[2,0]=5$ $a[2,1]=6$ $a[2,2]=7$

Orden 5

$a[0,0]=1$ $a[0,1]=2$ $a[0,2]=3$ $a[0,3]=4$ $a[0,4]=5$

$a[1,0]=0$ $a[1,1]=6$ $a[1,2]=7$ $a[1,3]=8$ $a[1,4]=0$

$a[2,0]=0$ $a[2,1]=0$ $a[2,2]=9$ $a[2,3]=0$ $a[2,4]=0$

$a[3,0]=0$ $a[3,1]=10$ $a[3,2]=11$ $a[3,3]=12$ $a[3,4]=0$

$a[4,0]=13$ $a[4,1]=14$ $a[4,2]=15$ $a[4,3]=16$ $a[4,4]=17$

:

:

etc

Matriz pares

Orden 2

$a[0,0]=1$ $a[0,1]=2$

$a[1,0]=3$ $a[1,1]=4$

Orden 4

$a[0,0]=1$ $a[0,1]=2$ $a[0,2]=3$ $a[0,3]=4$

$a[1,0]=0$ $a[1,1]=5$ $a[1,2]=6$ $a[1,3]=0$

$a[2,0]=0$ $a[2,1]=7$ $a[2,2]=8$ $a[2,3]=0$

$a[3,0]=9$ $a[3,1]=10$ $a[3,2]=11$ $a[3,3]=12$

Orden 6

$a[0,0]=1$ $a[0,1]=2$ $a[0,2]=3$ $a[0,3]=4$ $a[0,4]=5$ $a[0,5]=6$

$a[1,0]=0$ $a[1,1]=7$ $a[1,2]=8$ $a[1,3]=9$ $a[1,4]=10$ $a[1,5]=0$

$a[2,0]=0$ $a[2,1]=0$ $a[2,2]=11$ $a[2,3]=12$ $a[2,4]=0$ $a[2,5]=0$

a[3,0]=0 a[3,1]=0 a[3,2]=13 a[3,3]=14 a[3,4]=0 a[3,5]=0
a[4,0]=0 a[4,1]=15 a[4,2]=16 a[4,3]=17 a[4,4]=18 a[4,5]=0
a[5,0]=19 a[5,1]=20 a[5,2]=21 a[5,3]=22 a[5,4]=23 a[5,5]=24
:
:
etc.

b) Transpilar a Python el código de flowgorithm del inciso anterior y agregar las instrucciones necesarias con el módulo pyexcel_ods para almacenar las matrices generadas a LibreOffice Calc

Solicitar este ejercicio a Kimi y generar la solución tanto en flowgorithm como en Python



Ejercicio01: Rompecabeza01 – Generador de matrices "QR" con patrón par/impar

Autor: Kimi

Objetivo: Construir Flowgorithm que discrimine orden par/impar, genere matrices con patrón consecutivo y exporte a LibreOffice Calc mediante `pyexcel_ods` .

a) Código Flowgorithm (XML listo para importar)

Guarda como `Rompecabeza01.fprg`

xml

📄 复制

```
<?xml version="1.0"?>
<flowgorithm fileversion="3.0">
  <!-- =====
    ROMPECABEZA01 - Matrices 2D con patrón par/impar
    Autor: Kimi
    ===== -->
  <function name="Main" type="None">
    <parameter/>
    <body>
      <!-- Solicitar orden -->
      <declare name="n" type="Integer" array="False"/>
      <output text="Orden de la matriz (impar o par): "/>
      <input variable="n"/>

      <!-- Validar par/impar -->
      <if expression="n mod 2 = 0">
        <then>
          <output text="Orden PAR detectado." newline="True"/>
          <call function="GenerarMatrizPar"><argument name="n"/></call>
        </then>
        <else>
          <output text="Orden IMPAR detectado." newline="True"/>
          <call function="GenerarMatrizImpar"><argument name="n"/></call>
        </else>
      </if>
    </body>
  </function>
```



```
<!-- ===== FUNCIÓN: Matriz IMPAR ===== -->
<function name="GenerarMatrizImpar" type="None">
  <parameter name="n" type="Integer" array="False"/>
  <body>
    <declare name="A" type="Integer" array="True" size="10000"/>
    <declare name="cont" type="Integer" array="False"/>
    <assign variable="cont" expression="1"/>

    <for variable="i" start="0" end="n - 1">
      <for variable="j" start="0" end="n - 1">
        <!-- Patrón: fila impar + 0, excepto esquinas -->
        <if expression="i mod 2 = 1 and j ≠ 0 and j ≠ n - 1">
          <then>
            <assign variable="A[i * n + j]" expression="0"/>
          </then>
          <else>
            <assign variable="A[i * n + j]" expression="cont"/>
            <assign variable="cont" expression="cont + 1"/>
          </else>
        </if>
      </for>
    </for>

    <output text="Matriz IMPAR generada:" newline="True"/>
    <call function="Mostrar2D"><argument name="A"/><argument name="n"/></call>
  </body>
</function>
```


b) Transpilación a Python + exportación con pyexcel_ods

Guarda como Rompecabeza01.py

Python

复制

```
# Rompecabeza01.py
# Genera matrices 2D con patrón par/impar y exporta a LibreOffice Calc (.ods)

import numpy as np
import pyexcel_ods as ods

MAX = 100 # tamaño máximo (igual que en Flowgorithm)

def generar_impar(n: int) -> np.ndarray:
    """Matriz impar con patrón: filas impares → 0 excepto esquinas"""
    A = np.zeros((n, n), dtype=int)
    cont = 1
    for i in range(n):
        for j in range(n):
            if i % 2 == 1 and j != 0 and j != n - 1:
                A[i, j] = 0
            else:
                A[i, j] = cont
                cont += 1
    return A
```

```
def generar_par(n: int) -> np.ndarray:
    """Matriz par con patrón: filas impares → 0 excepto esquinas"""
    A = np.zeros((n, n), dtype=int)
    cont = 1
    for i in range(n):
        for j in range(n):
            if i % 2 == 1 and j != 0 and j != n - 1:
                A[i, j] = 0
            else:
                A[i, j] = cont
                cont += 1
    return A

def exportar_ods(matriz: np.ndarray, nombre: str):
    """Exporta matriz a archivo .ods (LibreOffice Calc)"""
    data = [matriz[i].tolist() for i in range(matriz.shape[0])]
    ods.save_data(nombre, {"Hoja1": data})
    print(f"Archivo {nombre} generado.")
```

```
def main():
    n = int(input("Orden de la matriz (impar o par): "))
    if n <= 0 or n > MAX:
        print("Error: n debe estar entre 1 y", MAX)
        return

    if n % 2 == 1:
        M = generar_impar(n)
        exportar_ods(M, "matriz_impar.ods")
    else:
        M = generar_par(n)
        exportar_ods(M, "matriz_par.ods")

    print("Matriz generada:")
    print(M)

if __name__ == "__main__":
    main()
```

✓ Cómo usar el script Python

1. Instalar dependencia (una sola vez):

```
bash 📄 复制
pip install pyexcel-ods numpy
```

2. Ejecutar:

```
bash 📄 复制
python Rompecabeza01.py
```

3. Abrir el archivo `.ods` generado (`matriz_impar.ods` o `matriz_par.ods`) con LibreOffice Calc.

🚀 Conclusión

- Flowgorithm → lógica visual y validación de patrón par/impar.
- Python + pyexcel_ods → exportación directa a hojas de cálculo sin código adicional.
- Un solo arreglo 1D → simulación de matriz 2D con fórmula de stride (`i * cols + j`).

INSTRUCTOR



Escribir el código flowgorithm y Python de este ejercicio. Si existe algún error modificar el código para que funcione de manera correcta.

Almacenar el resultado de la matriz en LibreOffice Calc.