



503202/503203 Programación Programación usando Arreglos 2

EQUIPO COORDINACIÓN DE PROGRAMACIÓN

15 de junio de 2025

- 1.- Escriba un programa Python que tenga una función `crea_trid()` que, dado los valores n , a , b y c , devuelva la matriz tridiagonal de orden n de la forma:

$$\begin{pmatrix} a & b & 0 & \dots & 0 & 0 \\ c & a & b & \dots & 0 & 0 \\ 0 & c & a & b & \dots & 0 \\ 0 & 0 & c & a & b & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a \end{pmatrix} \quad (1)$$

Entradas: La entrada a este programa incluye 4 números enteros, el primero mayor que cero correspondiente a n . Los tres últimos números enteros deben ser distintos de cero.

Salidas: La única salida del programa incluye sólo una matriz de $n \times n$ con las tres diagonales principales ocupadas por los valores a , b y c .

Ejemplo de entrada: $n=5$, $a=7$, $b=-1$, $c=3$

Ejemplo de salida:

$$\begin{pmatrix} 7 & -1 & 0 & 0 & 0 \\ 3 & 7 & -1 & 0 & 0 \\ 0 & 3 & 7 & -1 & 0 \\ 0 & 0 & 3 & 7 & -1 \\ 0 & 0 & 0 & 3 & 7 \end{pmatrix} \quad (2)$$

- 2.- Una matriz de Hilbert se define por

$$\mathbf{H}_n = (h_{ij}^{(n)}) \in R^{n \times n}, \text{ con } h_{ij}^{(n)} = \frac{1}{i+j-1}; i, j = 1, 2, \dots, n$$

Escriba un programa en Python que lea el valor de n y genere una matriz de Hilbert.

Entradas: La única entrada al programa es un número entero positivo mayor a 1.

Salidas: La única salida del programa es una matriz de Hilbert de $n \times n$.

Ejemplo de entrada: $n=5$

Ejemplo de salida:

$$\begin{pmatrix} 1.00 & 0.50 & 0.33 & 0.25 & 0.20 \\ 0.50 & 0.33 & 0.25 & 0.20 & 0.17 \\ 0.33 & 0.25 & 0.20 & 0.17 & 0.14 \\ 0.25 & 0.20 & 0.17 & 0.14 & 0.13 \\ 0.20 & 0.17 & 0.14 & 0.13 & 0.11 \end{pmatrix} \quad (3)$$

- 3.- SUDOKU es un juego matemático japonés cuyo objetivo es rellenar una cuadrícula de 9×9 celdas (81 casillas) dividida en subcuadrículas de 3×3 con las cifras del 1 al 9. Una solución válida de un SUDOKU es correcta solamente si no se repiten números en una misma fila, columna y subcuadrícula. Escriba un programa Python que lea la solución de un SUDOKU y despliegue si la solución es correcta o incorrecta.

3	4	9	8	1	6	5	7	2
2	7	5	9	3	4	8	6	1
8	1	6	2	7	5	9	3	4
9	6	2	7	8	1	4	5	3
4	5	7	6	9	3	2	1	8
1	8	3	4	5	2	6	9	7
5	9	4	1	2	7	3	8	6
6	3	1	5	4	8	7	2	9
7	2	8	3	6	9	1	4	5

- 4.- La descendencia de una colonia de seres vivos se puede modelar mediante un programa computacional, el cual maneja una matriz que corresponde al territorio donde habitan estos seres. Cada celda de la matriz contendrá un cero o un uno indicando respectivamente la ausencia o presencia de un ser vivo en el territorio.

Las reglas para determinar la evolución de las generaciones de estos seres son las siguientes:

- un habitante morirá por falta de alimento si en una generación está rodeado por tres o más seres vivos,
- un habitante morirá de soledad si está sólo o solamente está rodeado por un ser vivo,
- un habitante rodeado por dos seres vivos permanece vivo en la siguiente generación,
- si una celda vacía está rodeada por dos o tres seres vivos en la siguiente generación aparecerá ocupada por la reproducción de los habitantes de las celdas vecinas.

Se pide:

- Construir una función que permita ingresar una matriz cuadrada inicial de mínimo 3 y máximo 10 filas y columnas. La dimensión de la matriz se debe leer y se debe validar que el rango sea el indicado anteriormente. Los datos ingresados a la matriz también deben ser validados, es decir, el programa sólo puede aceptar valores 0 y 1 para cada posición.
- Construir una función que despliegue los elementos de la matriz.
- Construir una función que calcule el paso de una generación a otra.

El programa debe funcionar mostrando desde la primera generación cada evolución que ocurra con la colonia de seres vivos.

En la siguiente figura se muestra un ejemplo de evolución de una colonia de seres vivos:

0	1	2	3
1 1 0 0 1	0 0 1 0 0	0 1 1 0 0	0 0 0 1 0
1 1 0 0 0	0 0 1 1 0	0 0 0 0 1	0 0 0 0 0
0 0 1 0 0	1 0 1 0 1	0 0 1 0 0	0 0 0 0 1
0 0 0 1 1	1 0 0 0 0	0 0 0 1 0	0 0 1 0 0
1 1 0 1 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
4	5	6	7
0 0 0 0 0	0 0 0 1 1	0 0 0 0 0	0 0 0 0 0
0 0 0 1 1	0 0 1 1 1	0 0 1 0 0	0 1 0 0 0
0 0 0 1 0	0 0 0 0 0	0 1 0 0 0	0 0 1 1 0
0 0 0 1 0	0 0 1 0 1	0 0 0 1 0	0 0 1 0 1
0 0 0 0 0	0 0 0 0 0	0 0 0 1 0	0 0 1 0 1