Nombre del Estudiante \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Carnet \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Nota \_\_\_\_\_\_\_\_

Siga la sintaxis de Python y ajústese a lo que se le solicita. Para cada una de las preguntas que se le formula, construya las funciones que requiera para cumplir con el requerimiento solicitado. **DOCUMENTE SU SOLUCION.**

El examen debe ser construido en un archivo llamado Examen3\_SuNombre.py, por ejemplo Examen3\_Alejandro.py el cual deberá ser subido al tec-digital en la carpeta Tercer Parcial de la carpeta compartida de la comunidad para el curso de Introducción a la Programación. ***NO SE REVISARÁN PRUEBAS RECIBIDAS DESPUES DE LA HORA DETERMINADA COMO FINAL (12 medio día) AUNQUE SE REALICE LA ENTREGA.***

El estudiante deberá construir un archivo con las funciones que así requiera para dar solución a cada uno de los problemas planteados en esta prueba así como además debe definir una prueba que contenga la declaración de variables necesarias y la invocación a la función solicitada con el objetivo de ver el resultado de su ejecución.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pregunta 1  **suma-mitad-matriz** | Prueba  **Pregunta 1** | Pregunta 2  **sub-matriz** | Prueba  **Pregunta 2** | Pregunta 3  ***invertibles*** | Prueba  **Pregunta 3** | Pregunta 4  ***ceros*** | Prueba  **Pregunta 4** | Total |
| 22 puntos | 3 puntos | 22 puntos | 3 puntos | 22 puntos | 3 puntos | 22 puntos | 3 puntos | 100 puntos |
|  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| Pregunta extra  **Look-and-say** | Prueba  **Look-and-say** |
| 22 puntos | 3 puntos |
|  |  |

1. Escriba una función **iterativa** en PYTHONllamada ***suma-mitad-matriz*** que reciba una matriz de dimensión mXn m: filas y n: columnas que la sume las celdas de las dos mitades de la matriz e indique si esta suma es igual para ambas mitades.

En caso de que la matriz tenga m y n impares, la celda ubicada en la posición [m/2] [n/2] se ignora en ambas la suma. Considere los casos:

Caso: M impar, N impar

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 3 | 3  Se ignora la casilla porque es la que define las dos mitades de la matriz |
| 2 | 0 | 5 |

>>>suma-mitad-matriz(M)

True

Caso: M par, N impar

|  |  |  |
| --- | --- | --- |
| 1 | 3 | 4 |
| 2 | 3 | 4 |
| 2 | 1 | 3 |
| 4 | 4 | 3 |

>>>suma-mitad-matriz(M)

True

Caso: M impar, N par

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 3 | 4 | 5 |
| 2 | 3 | 5 | 4 |
| 1 | 5 | 3 | 2 |

>>>suma-mitad-matriz(M)

False

Caso: M par, N par

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 6 | 7 | 5 |
| 4 | 2 | 3 | 4 |
| 3 | 2 | 1 | 4 |
| 7 | 5 | 3 | 2 |

>>>suma-mitad-matriz(M)

False

1. Escriba una función ***iterativa*** ***sub-matriz***que reciba una matriz de tamaño mxn , una cantidad de filas y una cantidad de columnas y obtenga la submatriz que se forma a partir de las posiciones dadas (inclusive).

Matriz M

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 6 | 7 | 5 |
| 4 | 2 | 3 | 4 |
| 3 | 2 | 1 | 4 |
| 7 | 5 | 3 | 2 |

>>> sub-matriz (m, 2,2) =>

|  |  |
| --- | --- |
| 1 | 4 |
| 3 | 2 |

1. Construya una función **iterativa** en Python llamada ***invertibles(m1, m2)*** que recibe dos matrices m1 y m2 de dimensiones nXn ambas y retorne si ambas matrices se consideran invertibles.

Una matriz m1 es invertible si existe una matriz m2 tal que m1\*m2 (multiplicación de matrices) sea igual a la matriz identidad (formada por ceros excepto su diagonal que son unos). Ejemplo: (25 pts)

>>>m1 = [[2,5], [1,3]]

>>>m2 = [[3,-5], [-1,2]]

>>> invertibles(m1, m2)

>>>True

1. Escriba una función iterativa llamada ***ceros(m)*** que recibe una matriz de tamaño nxm y devuelve una lista con las posiciones de la matriz que contienen ceros. Debe indicar en que fila y que columna aparece cada cero encontrado.

Observe que las filas y las columnas inician en la posición 0.

>>>mat= [0,1,2,4], [9,8,0,0], [5,0,3,2]]

>>>ceros(mat)

>>>[ [0, 0] , [1, 2] [1, 3], [2, 1] ]

**Puntos extra: se calificarán únicamente si todo el examen es contestado.**

Escriba una función **iterativa** llamada **look-and-say** que reciba en un arreglo una serie de números y que genere la serie numérica llamada de la misma forma en la cual cada término describe al anterior.

[1] contiene un uno, o sea, 11.

[1, 1] contiene dos unos, o sea, 21.

[2, 1] contiene un dos y un uno, o sea, 1211.

[1, 2, 1, 1] contiene un uno, un dos y dos unos, o sea, 111221.

[1, 1, 1, 2, 2, 1] contiene tres unos, dos doses y un uno, o sea, 312211.