



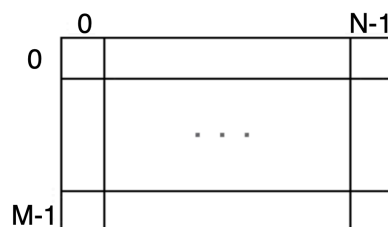
INTRODUCCIÓN A LA COMPUTACION

A continuación Ud. encontrará un problema que debe ser resuelto por equipos de no más de dos estudiantes, la solución final debe ser implementada en un programa escrito en lenguaje Python, utilizando funciones. La solución del problemas junto a su informe debe ser subida a la plataforma virtual del curso el día Jueves 06 de Diciembre hasta las 23:50 P.M., no se recibirán trabajos después de la fecha y hora indicada. La no presentación del trabajo será evaluada con 1.0.

PROBLEMA

Como es usual, el gato Tom persigue a Jerry por las habitaciones de una casa sin ningún éxito, pues el maldito roedor conoce todos los recovecos de la mansión que habitan. Harto de esta situación, Tom adquiere un PC con la intención de construir un programa capaz de encontrar el camino más corto hasta su ansiada presa sorteando todos los obstáculos (paredes, armarios, etc.) que pueda encontrarse.

Representaremos la casa por una superficie rectangular de M filas y N columnas.:



Supondremos que tanto Tom como Jerry ocupan una casilla dentro de esta superficie. Los obstáculos de la habitación serán rectángulos de coordenadas válidas dentro de la superficie, y se representarán con su vértice superior izquierdo y el inferior derecho. Existe la posibilidad que el rectángulo sea en realidad una recta (horizontal o vertical; por ejemplo, las coordenadas (3, 4) y (3, 6) definen una recta horizontal) o incluso un punto (cuando las dos coordenadas son iguales).

Se pide construir un programa que **procese** una entrada, que contenga la configuración inicial de la casa, y que la dibuje si es correcta.

Los casos de error que consideramos serán:

- a. Error (E0) M o N (o ambas) son igual a 0.
- b. Error (E1) Tom o Jerry no se encuentran en coordenadas válidas de la superficie de la casa.
- c. Error (E2) Tom y Jerry están en la misma casilla.
- d. Error (E3) Algún obstáculo no está situado en coordenadas válidas de la superficie.
- e. Error (E4) Los vértices que representan un obstáculo no cumplen una relación válida entre sí.
- f. Error (E5) Dos o más obstáculos se solapan.
- g. Error (E6) Tom o Jerry están en una casilla ocupada por un obstáculo.

ENTRADA

Los datos se encuentran en un archivo de texto llamado “casa.txt” y la descripción de su contenido es la siguiente:

La primera línea contiene dos números enteros positivos separados por un único espacio en blanco y corresponde a los valores de M y N respectivamente.

La segunda línea contiene cuatro números naturales separados por un único espacio en blanco. Los dos primeros valores representan la posición inicial de Tom (fila, columna) y los dos siguientes la de Jerry (fila, columna).

La tercera línea contiene solo un número (K) que representa la cantidad de obstáculo que hay en la casa.

Las siguientes K líneas, contienen cada una de ellas cuatro números naturales separados por un único espacio en blanco y que identifican las coordenadas de los obstáculos rectangulares. Los dos primeros valores representan el vértice superior izquierdo del obstáculo (fila, columna) y los dos restantes el vértice inferior derecho (fila, columna).

SALIDA

Debe ser generada en un archivo de texto llamado "error.txt" el cual en caso de detectar un error debe contener una sola línea con el mensaje "ERROR Ex", siendo la 'x' un número entre 0 y 6 que identifica el tipo de error que se encuentre en la entrada. Si hubiese más de un error a la vez, se escribiera aquel con código menor; por ejemplo, si el primer error lo provoca un obstáculo que se solapa con alguno de los anteriores y al mismo tiempo ocupa la casilla de Tom o de Jerry, el código a devolver sería E5.

En caso de no encontrar error, la salida contendrá M líneas de N caracteres cada una de ellas, siendo cada carácter el contenido de una casilla, que será 'o' para las casillas vacías, 'x' para las casillas que forman parte de un obstáculo, 'T' para la casilla ocupada por Tom y 'J' para la casilla ocupada por Jerry.

A continuación, se muestran algunos ejemplos de entrada errónea y el mensaje que debe aparecer en pantalla:

casa.txt	salida.txt
3 3 1 1 2 2 1 0 1 1 0	ERROR E4
3 3 1 1 2 1 1 0 1 1 2	ERROR E6
3 3 2 1 1 2 2 0 0 1 1 0 1 1 2	ERROR E5

En cambio, las entradas siguientes generan resultados correctos:

casa.txt	salida.txt
5 5 0 0 4 2 2 1 0 2 2 0 3 3 3	Tooxo xxxxx xxxxx oooxo ooJoo
7 5 0 1 6 0 5 1 0 1 1 1 3 1 3 3 1 3 4 5 0 5 1 5 3 5 3	oTooo xxoxo ooooo oxxxx ooooo xxoxo Joxxxx

Su trabajo será evaluado de acuerdo a la siguiente Rubrica:

	1	3	5	7	Total
Revisión del Código. (20%)	Código desordenado, sin uso de las normas de programación, no utiliza comentarios, nombres de variables inadecuados. No utiliza funciones.	Código Ordenado Usa identificadores de variables adecuado. Incorpora comentarios mínimos o existe sobresaturación de comentarios. Uso básico de funciones.	Código ordenado. Uso adecuado de variables (listas, string, etc.). Uso adecuado de estructuras algorítmicas Uso eficiente de los comentarios. Uso adecuado de funciones	Cumple con todas las normas de programación.	
Solución Computacional. (40%)	No se ejecuta. Errores de Ejecución. Programa cae en Loop.	Ejecuta pero con errores de lógica. Ejecuta pero no llega a la solución.	Llega a una solución parcial. Tarda más de un minuto en llegar a la solución. No cumple con el formato de salida.	Llega a la solución en menos de un minuto cumpliendo exactamente con el formato de salida.	
Contenido del informe (20%)	No cumple con todos los apartados del informe o estos no expresan lo solicitado.	Cumple con todos los apartado del informe, pero están desarrollados pobremente.	Cumple con todos los apartado del informe, están desarrollados de acuerdo a lo solicitado, pero hay	Cumple con todos los apartado del informe, están desarrollados de acuerdo a lo solicitado y se presenta de	

			desprolijidad en su presentación.	Manera prolija.	
Gramática y ortografía (20%)	El informe contiene 9 o más errores gramaticales u ortográficos.	El informe contiene entre 5 y 8 (inclusive) errores gramaticales u ortográficos.	El informe contiene entre 1 y 4 (inclusive) errores gramaticales u ortográficos.	El informe no contiene ningún error gramatical u ortográfico.	

CONTENIDO DEL INFORME

EL proyecto debe ser subido a la plataforma, en el hora y fecha indicada, en un archivo comprimido (zip o rar) que contenga sólo el código fuente del programa (formato .py) y el informe en formato PDF.

El informe debe considerar mínimo: la estrategia de solución empleada, la descripción de las estructuras de datos utilizadas, los alcances y limitaciones del programa. También debe tener un apartado con el manual de usuario.

El formato a seguir por el documento es la siguiente: Primera hoja, identificación de la universidad, departamento, carrera, asignatura, nombre de los profesores, nombre de los estudiantes y fecha de entrega.

A Continuación debe venir la introducción, diseño de la solución, detalles de la implementación, limitaciones y finalmente el manual de usuario.

Introducción: Contiene la descripción general del problema (no el enunciado), en este punto debe quedar claro cuál es el problema que se debe resolver.

Diseño de la solución: Descripción clara de la manera como se abordó el problema. Debe explicar cómo el mundo real ha sido modelado en la solución planteada. A continuación se deben describir los algoritmos y estructuras de datos principales que transforman el mundo en un modelo informático utilizado para alcanzar la solución.

Como usted ya sabe, un problema debe ser descompuesto en subproblemas, es en este punto donde se debe indicar cuales fueron los subproblemas planteados y como a partir de ellos se llega a la solución general del problema.

Detalles de implementación: Esto se refiere a la transformación de los algoritmos en programas escritos en lenguaje Python 3, se debe incluir cualquier detalle que usted considere relevante.

Limitaciones: Se deben indicar las limitaciones del programa, las cuales pueden ser propias del enunciado o propias de la solución planteada. Aquí deben quedar claramente registradas aquellas cosas que usted no resolvió y que son fuente de error de su programa. Como por ejemplo errores que no pudo corregir por falta de tiempo, en cuyo caso debe detallar cual sería la estrategia de solución.

Manual de Usuario: Permite que cualquier usuario interactúe con el programa, sin tener conocimiento del interior del programa. Debe contener todos los detalles que el usuario debe conocer para poder utilizar el programa. Generalmente se nutre de imágenes que muestran ejemplos de ejecución.