Proyecto #1 - TankEscape

1. La historia

Hace mucho tiempo en unas batallas muy muy lejanas… (Primera y Segunda Guerra Mundial) hubo un general estadounidense llamado George Patton el cual fue caracterizado por personalidad extrovertida y valiente. Siempre iba de frente sin importar qué tan desesperanzadora se viese la situación. La mayoría de las batallas que dirigió terminaron en victorias, pero hubo una batalla en específico en que esto no ocurrió y gran parte de las tropas estadounidenses que ahí participaron, perecieron por su imposibilidad de escapar hacia zona segura. Luego de lo ocurrido el general Patton propuso la creación de un sistema automatizado para prevenir o por lo menos reducir situaciones adversas de tal magnitud. Pero en ese momento no se le dió solución al problema ya que no había tecnología lo suficientemente avanzada como para ello. Por eso hoy, **19 de noviembre del 2019,** la fuerza de inteligencia del ejército norteamericano le encarga a los estudiantes de Algoritmos y Estructuras de Datos implementar un programa que dé respuesta a la solicitud del fallecido general George Patton.

2. El pedido

El programa deberá consistir en procesar un mapa en específico y determinar cuántas posibilidades hay de sobrevivir en caso de que el vehículo se averíe en cualquier de las casillas posibles. Para que el tanque sobreviva, debe ser posible que llegue, desde una posición inicial (que no sea un muro), a la esquina superior derecha del mapa.

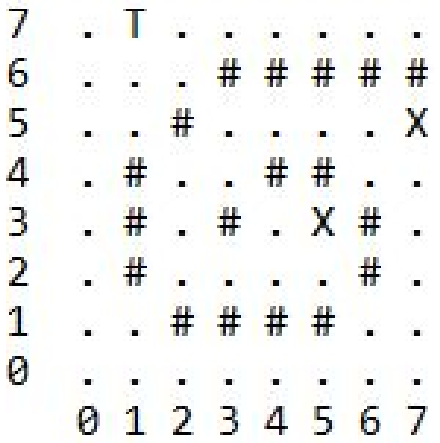
Representación en mapa

. Casilla vacía

T Tanque

Muro

X Punto de reparación



Como se puede observar, la columna 0 es la última, contrario a lo común, que es que sea la primera.

Además, como no todo puede ser tan fácil, se asume que, al inicio de cada comprobación, dicho tanque está averiado, por lo cual solamente tiene permitido desplazarse hacia arriba y hacia la derecha.

Existe la posibilidad que el tanque pueda ir hacia otras direcciones, pero esta requiere que el tanque pase por encima de una casilla marcada como "FixPoint". Para evitar ser descubierto, nadie (ni siquiera el mismo conductor del tanque) sabe dónde se encuentran los FixPoints hasta que están encima de dicha casilla. Por lo cual, no se debe de hacer ninguna especie de priorización a la hora de elegir en qué dirección ir.

Una vez que el tanque haya sido reparado, podrá devolverse por el camino que había recorrido previamente, para verificar si de esa forma puede hallar la vía de escape.

El criterio de decisión direccional del tanque es el siguiente:

Jerarquía	Significado	Representación en secuencia de pasos
1.	Arriba	W
2.	Derecha	D
3.	Abajo	S
4.	Izquierda	A
0.	FixPoint	X

Ya que el general Patton siempre fue extrovertido y adoraba los juegos, se pidió que las decisiones tomadas por el tanque fuesen representadas de esta manera, a forma de homenaje, debido a que son las teclas normalmente usadas para desplazarse en los videojuegos de **pc**.

Si es posible escapar desde una casilla [i] [j], entonces se almacena la secuencia de pasos en la estructura de datos idónea, para su posterior impresión. Si el tanque pasó encima de varios puntos de reparación en su camino hacia la salida, solo el primero deberá verse resaltado en la secuencia de pasos, ya que las reparaciones no son acumulativas y una vez reparado el tanque, los demás FixPoints que pueda pisar cumplen la misma función que las casillas vacías.

Por el contrario, cuando no sea posible escapar del mapa desde cualquier casilla [i] [j], se debe de tener en cuenta para que, al final del análisis de las demás casillas, se use para determinar la probabilidad de muerte segura en el campo de batalla.

3. Mensajes de error

Si alguna de las restricciones es infringida, se debe de mostrar el mensaje de error correspondiente a dicho error y, seguidamente, saltar al siguiente caso. (asuma que una vez encontrado el error, la siguiente línea siempre será la primera línea del caso siguiente, en caso de haberlo). Así pues, los mensajes permitidos son los siguientes:

- "El mapa posee dimensiones inapropiadas"
- "No puede haber un número negativo de muros"
- "Las coordenadas del muro N n son inválidas" (entiéndase por n el indicador de cuantos muros se han leído incluyendo el actual)
- "Se intentó construir un muro encima de otro"
- "Imposible escapar ya que hay un muro en la esquina superior derecha"
- "El número de fixPoints es menor que 0 o mayor que 4"
- "Las coordenadas del fixPoint N n son inválidas" (entiéndase por n el indicador de cuantos fixPoints se han leído incluyendo el actual)

4. La entrada

Todas las entradas serán leídas de un archivo “TankEscape.in”

La primera línea contendrá cuántos casos de prueba se evaluarán a lo largo de la ejecución del programa (N casos de prueba N > 0).

La siguiente línea contendrá la cantidad de filas (F) , columnas (C) y muros horizontales (W) que contendrá el mapa (F > 0 , C > 0 , W > 0).

Las W líneas siguientes indicarán las coordenadas de los muros del mapa (tomando en cuenta que la numeración de las filas está invertida) de la siguiente forma: Fi (fila del muro), Col1 y Col2 (columna de inicio y finalización del muro). ($0 \leq Fi \leq F - 1$, $0 \leq Col1 \leq Col2 \leq C - 1$). Asuma que no puede haber un muro en la casilla de la esquina superior derecha, ni se puede colocar un muro encima de una casilla que ya esté marcada con muro.

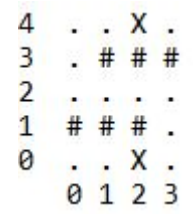
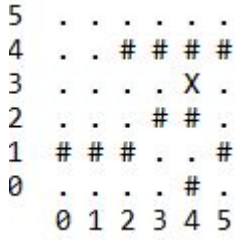
La siguiente línea X indica el número de puntos de reparación que habrá en el mapa ($0 \leq X \leq 4$).

Las X líneas siguientes indicarán las coordenadas de los FixPoints: fila (Row) y columna (Column) de los puntos de reparación ($0 \leq Row \leq F - 1$, $0 \leq Column \leq C - 1$)

5. La salida

La salida se guardará en un archivo llamado “TankEscape.out” el número de casos que se está por procesar. Una vez calculado el posible escape (o no) desde todas las casillas, se debe de emitir un informe que exponga la probabilidad de muerte, la cual será dada con la fórmula: cantidad_de_muertes / casillas _comprobadas *100. En las líneas siguientes, se debe escribir las secuencias de los pasos realizados por el tanque a partir de las casillas iniciales que sí le permitieron escapar.

6. Casos de prueba

Entrada	Mapa	Salida
3 2 2 1 0 1 0	No hay, puesto a que no se leen todos los datos	**Caso 1:** Las coordenadas del muro N 1 son inválidas
5 4 2 1 0 2 3 1 3 2 0 2 4 2		**Caso 2:** Porcentaje de muertes seguras: 35.714287 % Camino desde [4] [0]: DDXD Camino desde [4] [1]: DXD Camino desde [4] [2]: XD Camino desde [4] [3]: Camino desde [3] [0]: WDDXD Camino desde [2] [0]: WWDDXD Camino desde [0] [0]: DDXDWWAAAWWDDD Camino desde [0] [1]: DXDWWAAAWWDDD Camino desde [0] [2]: XDWWAAAWWDDD
6 6 5 0 4 4 1 0 2 1 5 5 2 3 4 4 2 5 1 3 4		**Caso 3:** Porcentaje de muertes seguras: 36.000000 % Camino desde [5] [0]: DDDDD Camino desde [5] [1]: DDDD Camino desde [5] [2]: DDD Camino desde [5] [3]: DD Camino desde [5] [4]: D Camino desde [5] [5]: Camino desde [4] [0]: WDDDDD Camino desde [4] [1]: WDDDD Camino desde [3] [0]: WWDDDDD Camino desde [3] [1]: WWDDDD Camino desde [3] [2]: DDXAASAWWWDDDD Camino desde [3] [3]: DXAASAWWWDDDD Camino desde [3] [4]: XAASAWWWDDDD Camino desde [2] [0]: WWWDDDDD Camino desde [2] [1]: WWWDDDD Camino desde [2] [2]: WDDXAASAWWWDDDD

7. Condiciones Generales

- El proyecto debe ser desarrollado en el lenguaje de programación C++.
- Será compilado y evaluado en el compilador g++ (Linux). Si el proyecto no compila por más de 5 errores o al grupo le toma más de 5 minutos hacerlo compilar, no será corregido.
- Los proyectos que no puedan ser ejecutados y las copias entre equipos tendrán la menor calificación. Aquellos que incumplan las condiciones generales tendrán una penalización de 10 puntos, sobre la nota final acumulada en el proyecto.
- Se pueden formar equipos de hasta máximo dos personas (alumnos de cualquier sección).
- La fecha de entrega queda pautada para el día **9 de Diciembre de 2019** hasta las 11:59 PM (GMT-4).
- La eficiencia del programa será tomada en cuenta a la hora de corregir.

8. Formato de entrega

- Destinatario: se debe enviar la solución al correo preparador, con copia al profesor de la sección correspondiente.

- Asunto: el asunto del correo a enviar con la solución del proyecto debe tener la siguiente estructura **[AYED] Proyecto #1 - Sección - Cédula _1_ Cédula _2**. ***Por ejemplo***: [AYED] Proyecto #1 - C1 - 11111111_22222222.

- Archivo adjunto: se debe adjuntar un archivo comprimido (.zip, .rar, entre otros) con el **nombre del asunto**. El archivo comprimido debe contener el/los archivo/s de código fuente en el lenguaje de programación indicado, junto a un **archivo PDF** que describa el análisis que llevó a la solución del problema. La intra documentación del código puede ayudar a la rapidez de la corrección, por lo cual se recomienda su realización.

- Si no se entrega el proyecto con este formato, acarreará una penalización en la nota.