

Disparo de la pistola laser

Para realizar este problema debe haberse realizado previamente el problema PR3.1 ya que se utilizan los tipos y funciones implementados en el problema.

En este problema se simulará el disparo de una pistola laser por una tortuga. La pistola dispara en la dirección de la tortuga y su objetivo es derretir un muro de hielo. Si el disparo encuentra un muro de hielo lo derrite y la casilla queda vacía, si encuentra cualquier otro objeto el disparo no tiene efecto. La pistola solo tiene fuerza para derretir un muro.

Para implementar el problema:

1. Declarar un tipo `tCoordenada` que represente una coordenada del tablero.
2. Modifica, si es necesario, el orden de los literales en el tipo enumerado `tDir` declarados en el problema anterior para que sea `UP`, `RIGHT`, `DOWN`, `LEFT`. De esta forma siguen el movimiento de las agujas del reloj.
3. Declarar dos vectores de incremento de filas y columnas que permiten manejar las cuatro direcciones del tablero. La primera debe ser la correspondiente a la dirección `UP`, la segunda la correspondiente a `RIGHT` etc.

Los alumnos del grupo DG deben declarar:

```
const std::vector<int> incF  {-1,0,1,0}
const std::vector<int> incC  {0,1,0,-1}
```

Observad que no son los mismos valores que los utilizados en el problema 6 del juez.

Los alumnos del grupo F debe declarar:

```
const int incF[] = {-1,0,1,0}
const int incC[] = {0,1,0,-1}
```

Los valores del vector `incF` indican el incremento que hay que aplicar a la fila dada para obtener los cuatro valores colindantes (arriba, derecha, abajo, izquierda) y los valores del vector `incC` indican los incrementos de la columna.

Estos vectores se utilizan para recorrer una fila o columna sin necesidad de diferenciar su dirección. Por ejemplo, si tenemos una tortuga en la posición `(x, y)` y que mira en la dirección `dir`, la casilla siguiente en esa dirección es: `tablero[x+incF[dir]][y+incC[dir]]`. Mediante un bucle podemos ir avanzando de una casilla en la siguiente utilizando la misma fórmula. De esta forma utilizamos menos instrucciones condicionales y hacemos el código mas legible.

4. Implementar una función `laser` que reciba un tablero, las coordenadas en que se encuentra la tortuga con la pistola laser y la dirección a la que apunta la tortuga y modifique el tablero de acuerdo con el resultado del disparo. La función debe utilizar los valores de los vectores `incF` y `incC` para recorrer las casillas del disparo. La función se puede implementar con un bucle, dos instrucciones condicionales y 5 asignaciones.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con dos valores, el primero indica el número de jugadores que quieren jugar, el segundo la tortuga que tiene la pistola laser. A continuación aparece la descripción de varios tableros, cada uno comienza con una línea en que se indica el número de tortugas que tiene. Después del número de tortugas aparecen 8 líneas con 8 valores cada una. Cada valor es un posible elemento del tablero. Cada caso acaba con una línea con tres guiones. El último caso se marca con un número de jugadores cero.

El tablero es de 8*8. El número de jugadores y de tortugas es mayor que cero y menor que 5. En todos los casos se garantiza que existe un tablero con un número de tortugas igual al número de jugadores. Los caracteres que representan cada elemento del juego son: `'#'`, es un muro de piedra, `'@'`, es una pared de hielo, `' '`, es una casilla vacía, `'$'`, es una joya, y `'C'` es una caja. Para representar una tortuga aparece una letra que indica la dirección en la que mira: `'U'`, `'D'`, `'R'`, `'L'` (`UP`, `DOWN`, `RIGHT`, `LEFT`).

Salida

Para cada caso de prueba se dibuja el tablero despues de disparar la pistola. Cada casilla se dibuja con dos caracteres. Se utilizan los siguientes símbolos para visualizar los elementos: "||", para los muros de piedra, "**", para los muros de hielo, "[]", para las cajas, "00", para las joyas; y para las tortugas que miran en las cuatro direcciones posibles usa: ">>", "<<", "^ ^", "v v"

Entrada de ejemplo

```
4 3
3
DC    CD
CC    CC
#0000#
@ $ @
#0$ $0#
0000
CC    CC
RC    CC
4
D@    @D
@@    @@
#0000#
@$$$@
#0$ $0#
0000
@@    @@
R      @U
---
2 2
2
DC    CD
CC    C
#0000#
@ $ @
#0 $0#
0000
CC    CC
CC    CC
---
0
```

Salida de ejemplo

```
VV**      **VV
****      ****
| |*****| |
  **0000**
| |**0000**| |
  ****
****      ****
>>      ^ ^

vv[]      []vv
[] []      []
| |*****| |
  **00  **
| |**  00**| |
  ****
[] []      [] []
[] []      [] []
```

Autor: Isabel Pita.