

Los movimientos de la tortuga

Para realizar este problema deben haberse realizado previamente los problemas PR3.2 y PR3.3 ya que se utilizan los tipos y funciones implementados en esos problemas.

La tortuga puede realizar cuatro movimientos: **AVANZAR**, **GIROIZQUIERDA**, **GIRODERECHA** y **LASER**. Cuando realiza el primero se desplaza una casilla en la dirección que está mirando, siempre que los objetos del tablero se lo permitan (ver problema PR3.3), con la segunda se gira 90° en el sentido de las agujas del reloj, con la tercera se gira 90° en sentido contrario a las agujas del reloj, y la última le permite disparar la pistola laser (ver problema PR3.2).

En este problema se simulará el movimiento de la tortuga al realizar la secuencia de acciones que se le indique.

Para implementar el problema:

1. Implementa una función `avanzar`, que modifique el tablero desplazando el objeto que se encuentre en la posición indicada por las coordenadas a la casilla siguiente en la dirección dada. Esta función supone que el movimiento es posible y solo avanza una casilla.
2. Para los alumnos del grupo DG. Declara una constante `NUM_DIRECCIONES = 4` con el número de direcciones posibles. Sobrecarga los operadores de incremento y decremento para el tipo enumerado `tDir`. Los operadores de incremento y decremento pueden utilizarse en forma prefija o en forma posfija. Cada uno tiene un comportamiento diferente. La forma prefija primero incrementa el valor de la variable y a continuación utiliza el valor incrementado en la expresión en que aparece el operador. La forma posfija utiliza el valor sin incrementar en la expresión y lo incrementa posteriormente. Por ello, hay que implementar dos funciones diferentes. Como ambas reciben el mismo nombre, `operator++`, la forma de diferenciarlas es mediante los parámetros. Se añade un parámetro ficticio de tipo entero a las implementaciones de los operadores posfijos. De esta forma el compilador hace la llamada a la función adecuada. A continuación se muestra la implementación del operador incremento.

```
// incremento prefijo
tDir operator++(tDir & dir) {
    dir = tDir((int(dir) + 1) % NUM_DIRECCIONES);
    return dir;
}

// incremento posfijo
tDir operator++(tDir & dir, int) {
    tDir aux = dir;
    dir = tDir((int(dir) + 1) % NUM_DIRECCIONES);
    return aux;
}
```

Para realizar la sobrecarga del operador decremento, se puede utilizar una instrucción condicional para controlar si se trata del último valor del tipo enumerado.

3. Implementa un módulo `tablero` con todas las funciones que se han implementado para manejar el tablero. En el fichero `tablero.h` coloca todas las declaraciones y los prototipos de las funciones vistos hasta este momento. En el fichero `tablero.cpp` coloca las implementaciones. No te olvides de poner el control de inclusiones múltiples al comienzo del archivo `.h`, y de incluir el archivo `.h` al comienzo del archivo `.cpp`
4. En el archivo en que se encuentra la función `main`,
 - (a) Incluye el archivo `tablero.h`
 - (b) Declara un tipo enumerado `tCarta` con los cuatro movimientos que puede realizar la tortuga. Declara un tipo `tVCartas` como un vector de cartas.

- (c) Implementa una función `realizarMovimiento` que dado un tablero, unas coordenadas y un vector de cartas, mueva la tortuga según las acciones indicadas en las cartas. Las acciones se realizan en el orden dado. Si en algún momento la tortuga llega a una joya, finaliza la ejecución de la secuencia, deja la tortuga en la posición de la joya, y devuelve el valor cierto. En otro caso, cuando finaliza la ejecución de la secuencia devuelve el valor falso. Para la implementación utiliza las funciones que implementamos en los problemas PR3.2 y PR3.3 de disparar la pistola y comprobar si el movimiento es posible.

Observa que si tienes declarados los literales del tipo enumerado `tDir` en el orden que se indicó en el problema PR3.2, la implementación de los giros se realiza con una sola instrucción.

5. Para subir al juez el problema debes subir los tres fichero (`tablero.h`, `tablero.cpp`, `principal.cpp`). Se seleccionan y después se elige el problema y el lenguaje.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con tres valores, el primero indica el número de jugadores que quieren jugar, el segundo la tortuga que va a desplazarse y el tercero el número de acciones que se van a realizar. En la línea siguiente se muestran las acciones a realizar. A continuación aparece la descripción de los tableros como en los problemas anteriores.

Salida

Para cada caso de prueba se dibuja el tablero después de realizar la secuencia de acciones indicada.

Entrada de ejemplo

```
4 3 5
AVANZAR AVANZAR AVANZAR GIROIZQUIERDA LASER
4
D@    @D
@@    @@
#@@@@#
@$$@
#@$$@#
@@@@
@@    @@
R     @U
---
2 2 3
AVANZAR GIRODERECHA AVANZAR
2
DC    CD
CC    CC
#@@@@#
@$ @
#@ $@#
@@@@
CC    CC
CC    CC
---
```

```

2 1 5
GIROIZQUIERDA AVANZAR AVANZAR GIRODERECHA LASER
2
DC      CD
CC      CC
@#@@@#@
  @$ @
  #@ $@#
  @@@@
CC      CC
CC      CC
---
0

```

Salida de ejemplo

```

vv**          **vv
****          ****
  ||*****||
    **0000**
  ||**0000**||
    **  ****
****          ****
    ^^      **^^

vv[]          []
[] []        []<<
  ||*****|| []
    **00  **
  ||** 00**||
    *****
[] []        [] []
[] []        [] []

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

```

Autor: Isabel Pita.