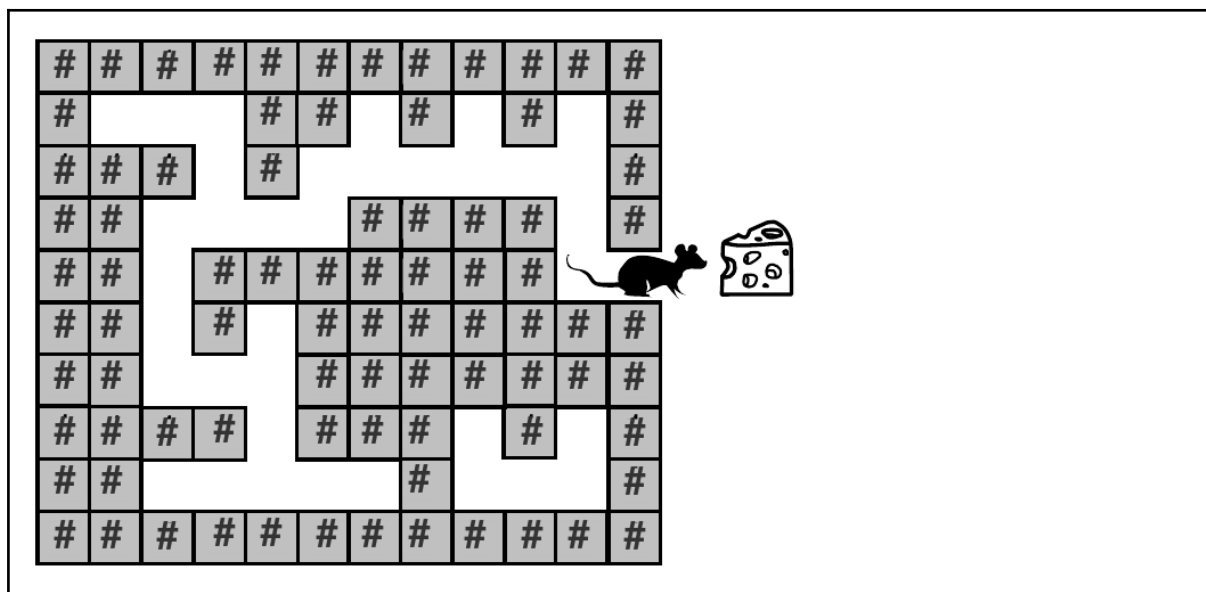


Exploración de un laberinto

Escribe un programa que simule un ratón en un laberinto. El programa debe imprimir el camino seguido por el ratón desde el punto de inicio hasta el punto final, incluyendo todos los lugares visitados y retrocedidos. Por lo tanto, si un lugar es visitado dos veces, debe ser impreso dos veces; si es visitado tres veces, debe ser impreso tres veces.

El laberinto se muestra en la Figura 1. El punto de entrada, donde el ratón comienza su recorrido, es elegido por el usuario que ejecuta el programa. Se puede cambiar en cada ejecución.



Se puede utilizar una matriz bidimensional como estructura de datos de apoyo para almacenar el laberinto. Cada elemento de la matriz puede ser negro o blanco. Un elemento negro es un cuadrado al que el ratón no puede ingresar o pared. Un elemento blanco es un cuadrado que puede ser utilizado por el ratón o el camino. En la matriz, un elemento negro se puede representar con un '#' y un elemento blanco con un espacio o '.'.

Cuando el ratón atraviesa el laberinto, visita los elementos uno por uno. En otras palabras, el ratón no considera el laberinto como una matriz de elementos; en cada momento de su recorrido, está solo en un elemento. Llamemos a este elemento el "currentSpot" (punto actual). Puede ser representado por una estructura con dos campos enteros. El primer



campo es la coordenada de fila y el segundo es la coordenada de columna del punto en el laberinto. Por ejemplo, la salida en la Figura 1 está en (5,12), es decir, fila 5, columna 12.

El programa comienza creando el laberinto que debe ser leído a partir de un archivo plano. Luego inicializa el punto de salida y solicita al usuario las coordenadas del punto de entrada. El programa debe ser robusto. Si el usuario ingresa coordenadas de un punto negro, el programa debe solicitar nuevas coordenadas hasta que se ingrese un punto blanco. El ratón comienza desde el punto de entrada y trata de llegar al punto de salida y su recompensa. Sin embargo, ten en cuenta que algunas posiciones de inicio no conducen a la salida.

A medida que el ratón avanza en su recorrido, imprime su camino con letra en mayúscula 'O'. Al entrar en un punto, el programa determina la clase de ese punto. La clase de un punto puede ser una de las siguientes:

- A. **Continuación:** un punto es una continuación si uno y solo uno de los vecinos (excluyendo el último punto) es un punto blanco. En otras palabras, el ratón solo tiene una opción.
- B. **Intersección:** un punto es una intersección si dos o más de los vecinos (excluyendo el último punto) son puntos blancos. En otras palabras, el ratón tiene dos o más opciones.
- C. **Callejón sin salida:** un punto es un callejón sin salida si ninguno de los vecinos (excluyendo el último punto) es un punto blanco. En otras palabras, el ratón no tiene punto para elegir. Debe retroceder.
- D. **Salida:** un punto es un punto de salida si el ratón puede salir del laberinto. Cuando el ratón encuentra una salida, es libre y recibe un trozo de queso como recompensa.

Para resolver este problema, necesitas dos pilas. La primera pila, la pila de visitados, contiene el camino que el ratón está siguiendo. Cuando el ratón llega a un punto, primero verifica si es una salida. Si no lo es, su ubicación se coloca en la pila. Esta pila se usa si el ratón llega a un callejón sin salida y debe retroceder. Cuando el ratón retrocede al último punto de decisión, también imprime el camino de retroceso.

Cuando el ratón entra en una intersección, las alternativas se colocan en una segunda pila. Este punto de decisión también está marcado por un token de decisión especial que se coloca en la pila de visitados. El token de decisión tiene coordenadas de (-1, -1). Para seleccionar un camino, una alternativa se saca de la pila de alternativas y el ratón continúa en su camino.

Durante el retroceso, si el ratón llega a un token de decisión, se descarta el token y se selecciona la siguiente alternativa de la pila de alternativas. En este punto, se imprime un



guión (-) junto a la ubicación para mostrar que se está seleccionando el siguiente camino alternativo.

Si el ratón llega a un callejón sin salida y ambas pilas están vacías, el ratón está atrapado en una parte del laberinto sin salida. En este caso, imprime un mensaje de estar atrapado y termina la búsqueda de una salida.

Después de cada intento, independientemente del resultado, se le debe dar al usuario la oportunidad de detenerse o continuar.

Archivo laberinto

```
#####  
#   #   ##  ##          #  
      #   #####  
# #   ##  ####  ##  ##  
# #   # #  ##   ###  #  
#           ##  ##  # #  
##### # #           # #  
##### ###  # #   ##  #  
#           # #   # #  #  
##### #   # #   #   # #  
#####
```

Solución

```
#####  
#-00#000##  ##          #  
000000#000  #####  
#-#-00  ##0  ####  ##  ##  
#-#-00#  #0##  00  ##  #  
#--00   00#00##  #  #  
#####-#  #-00000##  #  #  
#####-###--#-#00##  #  
#-----#-#  0#  #  #  
#####-#--#-#-#  #  #  
#####
```