Fundamentos de Programación y Lab.
PRÁCTICA No. 3
RESOLUCIÓN DE UN LABERINTO

Mtro. Jorge Rodríguez

Este documento presenta la especificación de requisitos de un programa que, a través de un algoritmo recursivo, debe encontrar el camino de salida óptimo (menor cantidad de pasos) de un laberinto que cuenta con uno o más caminos que llevan de la entrada a la salida.

I. DESCRIPCIÓN DEL SISTEMA

Dado un laberinto, se requiere un programa que permita encontrar el mejor camino que lleve del punto de entrada al punto de salida. Se entiende por el "mejor camino" a aquel que recorre el menor número de pasos. El laberinto siempre deberá de contar con una entrada, con una salida y con al menos un camino que permita llegar desde la entrada hasta la salida.
El programa será ejecutado desde la línea de comandos y, en función del argumento que se indique, podrá presentar la solución de dos formas posibles:

a) $ laberinto nombre_archivo
b) $ laberinto nombre_archivo -pasos

a) Si no se indica argumento, al momento de ejecutar el programa se despliega en pantalla el laberinto, se presiona la tecla <enter> y se despliega la solución óptima, indicando cuántos pasos hay entre la entrada y la salida e indicando cuántos caminos de salida se encontraron.

b) si se indica el argumento –pasos, al ejecutar el programa se despliega en pantalla el laberinto y, al presionar la tecla <enter>, comienza a desplegarse el proceso de resolución para finalizar con el despliegue del laberinto con el camino óptimo marcado, el número de pasos entre la entrada y la salida e indicando cuantos caminos de salida se encontraron.

II. ALCANCES Y LIMITACIONES

Al momento de ejecutar el programa se debe validar que el argumento que se pasa, si es el caso, sea correcto, así como que no se reciba más de dos argumentos. Se deberá desplegar el mensaje de error correspondiente, así como la forma adecuada de usar el programa; por ejemplo, si el usuario ejecuta:
$laberinto –p
el programa debe contestar:

Error, opción incorrecta.
Uso: $ laberinto nombre_archivo
$ laberinto nombre_archivo -pasos

No se considera necesario contemplar algún esquema adicional de validación. Se asume que el programador siempre usará un laberinto cuyo perímetro sea una pared, sin importar si norma una figura regular o irregular, así como que siempre habrá una entrada, una salida y al menos un camino que permita ir de la entrada a la salida.

El tamaño máximo del laberinto es de 30 columnas por 30 renglones.

Ejemplo:
Laberinto sin resolver:

```
**********
**         *
** **** *** *
** **** *** *
** **** *** *
**E       S*
**********
```

Laberinto resuelto:

```
**********
**         *
** **** *** *
** **** *** *
** **** *** *
**E ……. S*
**********
```

Camino óptimo: 10 pasos.
Se encontraron 3 caminos de salida.

III. RESTRICCIONES DE PROGRAMACIÓN

Para que esta práctica pueda obtener una calificación aprobatoria es indispensable que se cumplan los siguientes requisitos de programación.

1) El laberinto debe estar modelado por medio de una matriz de caracteres (puede considerarse también como un arreglo o lista de cadenas).
2) Las paredes se representarán con el carácter '*'.
3) La entrada se representará con el carácter 'E'.
4) La salida se representará con el carácter ´S´.
5) Las casillas del camino se marcarán con el carácter '.'.
6) El algoritmo que se emplee para recorrer el laberinto debe ser 100% recursivo (función recursiva).

7) Debe existir una función que analice el laberinto y que encuentre la posición o coordenadas correspondientes al punto de entrada, de tal manera que en el programa no se tenga que indicar de manera estática o a través de constantes o etiquetas dichas coordenadas.

IV. RESTRICCIONES PARA EL PROGRAMADOR

Se deben respetar las siguientes indicaciones:

1) Para el desarrollo del programa se debe emplear exclusivamente lo visto en clase, o alguna otra de las funciones estándar de C que sirvan para dar mejor presentación al programa, tal como la función system("clear").
2) El código del programa debe estar debidamente comentado y alineado.
3) Debe usar identificadores significativos.
4) Debe usar funciones correctamente.
5) La presentación de los datos y la limpieza del despliegue de los resultados en pantalla será un factor a considerar para la calificación de la práctica.

V. DOCUMENTACIÓN

Se deberá entregar un trabajo escrito con la definición del problema, el diseño de la solución (diagramas de bloques y/o pseudocódigo, en tantos niveles como sea necesario), debiendo considerar los siguientes puntos:

1. La documentación deberá presentarse usando el formato de artículos del Institute of Electrical and Electronics Engineers (IEEE), que es el mismo formato que observa este documento.
2. Cada error ortográfico será penalizado con 0.5 puntos menos en la calificación de la documentación.

Nota: Exceptuando el código fuente, todo el texto escrito en el documento, incluyendo el pseudocódigo y las palabras escritas con letras mayúsculas (por ejemplo, en títulos), deberá acentuarse correctamente.