



PROYECTO DE APLICACIÓN NO. 02

“Laberinto”

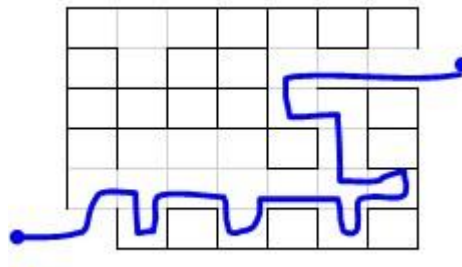
Objetivos Generales

- ✓ Practicar los conceptos teóricos sobre el desarrollo y codificación haciendo uso RISC V.
- ✓ Interrelacionar las diferentes aplicaciones prácticas y metodologías para el desarrollo de un producto específico.
- ✓ Manejar la estructura básica, instrucciones específicas, procedimientos y lógica propia del Lenguaje Ensamblador.
- ✓ Manejar las estructuras de arreglos bidimensionales aplicando el mapeo lexicográfico como método de obtención de direcciones.

Descripción General

Se tiene un laberinto lógico, formado por cuadrados que tienen uno o más de sus lados abiertos; el laberinto tiene una entrada definida y salida probable.

La descripción gráfica de las celdas del laberinto (representación gráfica simbólica) se presenta a continuación:



Se pueden tener diferentes laberintos, los cuales serán presentados como entrada, por medio de un archivo de texto (.txt), detallado posteriormente.

Para la resolución del laberinto, es decir encontrar la salida probable, se debe seguir la “regla de la mano derecha” que establece: al ingresar a un laberinto plano, si se coloca la mano derecha en la pared de la abertura de ingreso y



luego camina hacia adelante, sin alejar la mano de la pared, eventualmente se llegará a la salida del mismo, si dicha salida es inaccesible, se saldrá por la entrada.

Consideraciones Generales

Este programa deberá elaborarse en lenguaje RISC V.

La entrada al programa vendrá en un archivo plano de texto (.txt) y contendrá la descripción del laberinto:

- Dos enteros positivos, Filas y Columnas (**Filas ≤ 10 , Columnas ≤ 10**), que indican respectivamente el número de filas y columnas del laberinto.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42

- Un entero positivo que indica el número de celda donde se encuentra la entrada al laberinto, seguido de una letra que indica la pared en la que se encuentra el punto de ingreso específico (las paredes de cada celda se identificarán con las letras 'A', 'B', 'C', 'D', empezando por la pared vertical izquierda y se seguirá en el sentido de las agujas del reloj).
- Un entero positivo que indica el número de celda donde se encuentra la salida del laberinto, seguido de una letra que indica la pared en la que se encuentra el punto de egreso específico
- El resto de la entrada describe cada celda del laberinto, en cualquier orden, seguida de las paredes sobre las que tienen aberturas, también en cualquier orden.
- El final de la descripción del laberinto estará identificado por el número '0'.

La salida deberá generarse en un archivo plano de texto, cuyo nombre será "Salida_Laberinto.txt", almacenado de dónde se obtuvo el archivo de entrada, conteniendo:



- Búsqueda de la salida del laberinto, con el listado de la secuencia de celdas (con las paredes consultadas) empleada para intentar encontrar la salida.
- En la línea inmediata inferior de la secuencia de búsqueda debe contener la frase "Salida exitosa" o "Salida inaccesible", según corresponda como producto de la búsqueda.

Ejemplo de Entrada

07 06 01B 32C 02A 03B 05D 07D 09C 12B 12C 12D 13A 13C 14A 14C 19B 19C 20A 20D 22D 23 C 23D 24 A 24C
24D 25A 25D 27B 27D 29B 29C 29D 30A 30B 30C 30D 31A 31B 31C 0

Ejemplo de Salida

01A 01D 02D 02C 02B 01A

Salida inaccesible

Requerimientos Funcionales Específicos

- Entrada de datos, el archivo de entrada debe encontrarse en el mismo lugar donde se encuentra el .asm, y puede tener un nombre predefinido.
- Salida de datos, deberán crear un archivo en el mismo lugar donde se ejecuta el .asm conteniendo su solución.
- Programación defensiva respecto a la entrada, definiendo datos válidos y consistentes.

Nota: no es necesario que el usuario ingrese los paths de entrada y salida.

Entregables

- ✓ Código fuente del programa, en su archivo .asm.
- ✓ Documentación sobre el análisis y diseño de la solución, además un Manual de Usuario.