Introduction to Computing Systems

Assignment 4

Note: You MUST do the programming assignment by yourself. You are permitted to get help ONLY from the TAs and instructors.

Maze

Use recursion to get the path from the start cell to the exit cell of a given 6x8 maze.

The Maze File

The maze file will be given as following format:

1. The first line is the beginning address of the maze data. All maze files, i.e. the data stored in memory that describes the maze, use x4000 as their starting address. No exception.
2. The next two lines consist two non-negative integer **i, j** indicating the starting cell at **row i, column j.** (i<6,j<8)
3. And the following lines are the info for each cell of the maze. The order of storage is **row major**, i.e. for our 6*8 (6 rows and 8 columns) maze, the order of data is shown in Table 1:

Table 1

Line	Order
x4000	**starting row i**
x4001	**Starting column j**
x4002	Row 0, Column 0
X4003	Row 0, Column 1
x4004	Row 0, Column 2
x4005	Row 0, Column 3
x4006	Row 0, Column 4
x4007	Row 0, Column 5
x4008	Row 0, Column 6
x4009	Row 0, Column 7
…	…
x402e	Row 5, Column 4
x402f	Row 5, Column 5

x4030	Row 5, Column 6
x4031	Row 5, Column 7

The Cell Representation

Each line (16bit memory content) of the maze file saves the information of a corresponding cell of our maze. We set a protocol to represent each cell as Table 2:

Table 2

bit[15:5]	bit[4]	bit[3]	bit[2]	bit[1]	bit[0]
Not used	Destination	North	East	South	West
	1: Exit 0: Not exit	1: A pass in this direction 0: Inner wall or a border in this direction			

For example, the cell at row 1, column 7 in the sample maze is shown as Figure 1,
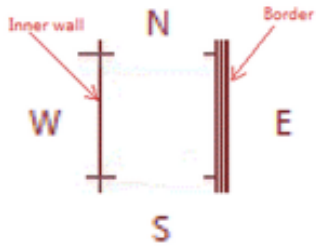

Figure 1

The initial binary representation of this cell in its memory location will be:

00000000000 0 1010

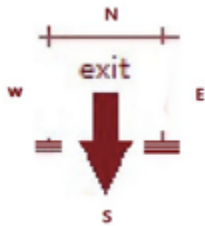The cell at row 5, column 1 which is an exit cell is shown in Figure 2,

Figure 2

The initial binary representation of this cell in its memory
location will be:

00000000000 1 0111

Note:

1. The starting cell for each maze would be at a **random** cell.
2. One can only get through a pass, yet cannot get through a wall
   or a border.

Output Format

In this program, you need to output the path on the console, each
2-bit decimal for a cell on the path, followed by a space symbol.
For each 2-bit decimal digit, the first bit is the row number, and
the second bit is the column number.
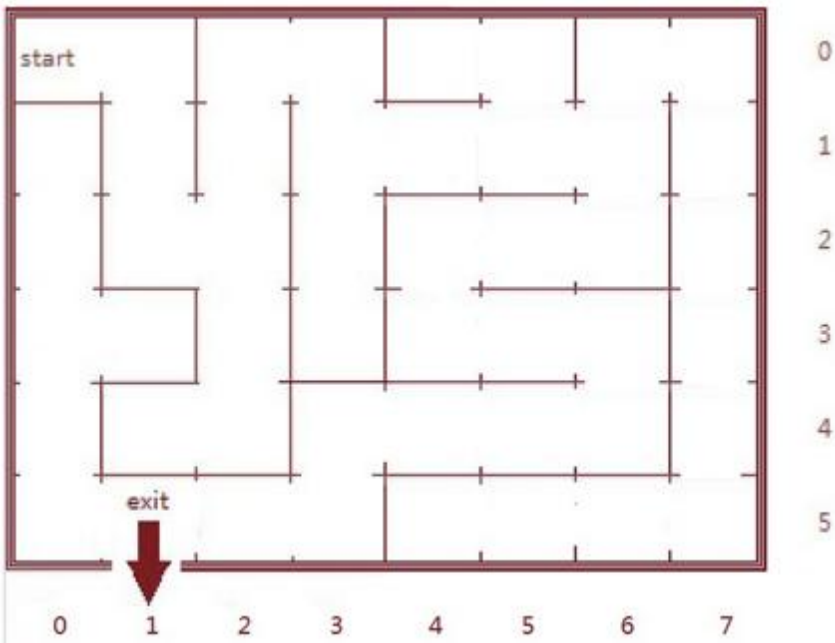
Sample Case

The Maze:

Figure 3 Maze

Maze file:

Table 3

0100000000000000
0000000000000000
0000000000000000
0000000000000100
0000000000000011

0000000000000110
0000000000000011
0000000000000100
0000000000000011
0000000000000110
0000000000000011
0000000000000010
0000000000001010
0000000000001010
0000000000001110
0000000000000101
0000000000001101
0000000000001011
0000000000001010
0000000000001010
0000000000001100
0000000000001011
0000000000001010
0000000000000110
0000000000000101
0000000000001001
0000000000001010
0000000000001110
0000000000000001
0000000000001010
0000000000001000
0000000000001100
0000000000000101
0000000000000011
0000000000001010
0000000000001010
0000000000000100
0000000000001001
0000000000000110
0000000000000101
0000000000000101
0000000000001001
0000000000001010
0000000000001100
0000000000010111
0000000000000101
0000000000001001
0000000000000100

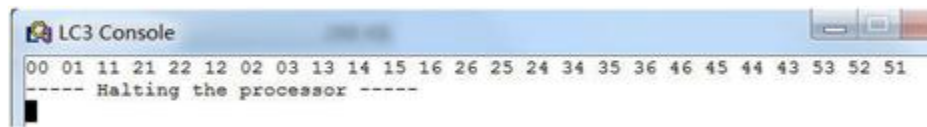| |
|---|
| 0000000000000101 |
| 0000000000000101 |
| 0000000000001001 |

Output:

Hints:

1) You need to initialize the stack pointer and memory location to save the result path. The stack of the procedure can be set at x6000 and the result path at x7000.

2) All test cases have **only one valid path,** so the output is unique.

3) We enumerate the row of the cells with a beginning of Row 0 instead of Row 1.

4) You should create more test cases by yourself and test your program.

Submit Your Program

The program to be submitted is the **.asm** file. You will submit the program file named as **maze.asm.** You SHOULD write a report for your program to briefly describe your idea for this assignment and how to use your program.

Submit your zip file to the website: http://10.214.208.4/intro2cs