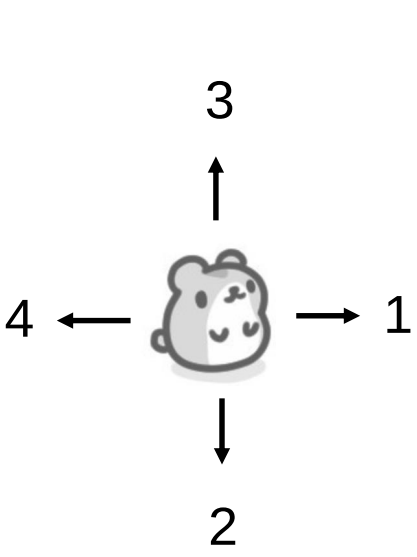
HW1 The rat in a maze

Objective:

1. Practice and use the stack function
2. Use array and structure type to solve the problem.

Description:

1. Load “maze.txt” to build a maze for the rat.
2. The maze was constructed by a 17 x 17 matrix as shown in **Fig. 1**.
3. Zeros represent the open path and ones are barriers.
4. Four directions (right -> down -> up -> left) are given and the movements of the rat must follow the order.
5. The rat should start its journey from and end at open path points.
6. Output the journey of the rat.



```
11111111111111111
10000000000101001
10100111111001101
10101100001010101
10111010100000001
10000011011111001
11111000010001001
10110010000100101
10110100101110111
10000111000100001
10110011000100101
10110010000011101
10111001111110101
10110010000010001
10010111110101111
10010000000000001
11111111111111111
```

Fig.1

Code execution:

You can use Dev-C++ to run your code.
Sample output:

1. Input the “maze.txt” and output the maze loaded from the file, as shown in **Fig. 2**.

```
enter filename : maze.txt
1111111111111111
1000000000101001
1010011111001101
1010110000101010
1011101010000001
1000011011111001
1111100001000100
1011001000010010
1011010010111011
1000011100010000
1011001100010010
1011001000001110
1011100111111010
1011001000001000
1001011111010111
1001000000000001
1111111111111111
```

Fig. 2

2. Then continuously enter a start and an exit position for a journey, as shown in **Fig. 3**.

(1) “enter start position : ” for input start coordinates.
(2) “enter exit position : ” for input exit coordinates.
*All of the input numbers for start and exit are
s.t. 1 <= x , y <= 15 ,
(x , y) = 0 (All of the entered positions should be open path)

Success	Failed
enter start position : 1 1 enter exit position : 15 15 0:1,1 1:1,2 2:1,3 3:1,4 4:1,5 5:1,6 6:1,7 7:1,8 8:1,9 9:1,10 10:1,9	enter start position : 13 15 enter exit position : 1 12 0:13,15 1:12,15 2:11,15 3:10,15 4:9,15 5:9,14 6:9,13 7:10,13 8:10,12 9:9,12 10:10,12
.....	.....
127:13,4 128:14,4 129:15,4 130:15,5 131:15,6 132:15,7 133:15,8 134:15,9 135:15,10 136:15,11 137:15,12 138:15,13 139:15,14 140:15,15 successfully escaped!!	203:8,13 204:9,13 205:9,14 206:9,15 207:10,15 208:11,15 209:12,15 210:13,15 211:13,14 212:13,13 213:12,13 214:13,13 215:13,14 216:13,15 Failed to escape.

Fig. 3

Output the path of the rat visited step by step, and check the rat can reach the exit or not, as shown in **Fig. 3**.

3. The code should be able to keep entering the position pairs until a coordinate position (-1,-1) is entered, as shown in **Fig. 4**.

```
enter start position : -1 -1
end the code.
```

Fig. 4

Deadline: (To be confirmed)

~~Regular: at 11:59 p.m., Monday Oct. 25, 2021.~~
Regular(updated): at 11:59 p.m., Friday Oct. 29, 2021.
Note: Handing late will lead to a perceptibly lower score.

Hand in:

Zip your rat.cpp, readme.pdf to StudentID_hw1.zip and upload the zip file to Moodle.

If you have any questions about the homework, you can e-mail to the TAs:
Eric: p76101039@gs.ncku.edu.tw
Kevin: p76101479@gs.ncku.edu.tw

Grade policies:

1.(10%)-readme file:

Briefly explain your code, such as which program language you used(C/C++) and the execution environment you used, e.g. Dev-C++, Codeblocks, Xcode, terminal, or there is any problem you met (at least 50 words) in a pdf file. (Can be in Chinese or in English)

2.(10%)-comments of your code:

Explain your code sections in your .cpp file in more details, about one comment for every 3~5 lines. (Must in English only)
*If there is any accident to your assignment while executing your code (e.g. compilation error or any exception), and you didn't write comments or readme file, arguments about scores will not be accepted.

3.(10%)-load file:

Successfully load file and show the loaded maze (see Fig. 2)

4.(10%)-loop and end of program:

Successfully keep entering the position pairs to print the journey of the rat and break the loop until the coordinate position (-1, -1) is entered (see Fig. 3 and Fig. 4)

5.(60%)-correctness:

Successfully get correct results of the maze.txt provided and an additional testing maze (other maze.txt), includes
(30%) successfully show the results of both columns in Fig. 3
(30%) successfully show the correct results of test cases from another maze.txt which is not given.

*Discussion is encouraged, but DO NOT share your code to your classmate. If plagiarism is found, both students will get 0 score in this assignment.