Data Structure Homework 3

繳交期限： 2020/10/27 17:00 前

補交期限(7 折)： 2020/11/3 17:00 前

手寫題：

4. Using manual transformation, change the following postfix or prefix expressions to infix:

   a. A B * C – D +
   b. A B C + * D –
   c. + – * A B C D
   d. – * A + B C D

6. If the values of A, B, C, and D are 2, 3, 4, and 5, respectively, manually calculate the value of the following prefix expressions:

   a. + – * A B C D
   b. – * A + B C D

8. Determine the value of the following postfix expressions when the variables have the following values: A is 2, B is 3, C is 4, and D is 5.

   a. A B C D * – +
   b. D C * B A + –

程式題：

12. Write a program to implement Algorithm 3-9, "Parse Parentheses," matching braces rather than parentheses. In your implementation, push the line number into the stack rather than the opening brace. When an error occurs, print the line number for the unmatched opening brace or unmatched closing brace. Test your program by running the source code through itself (there should be no errors) and then test it with the following small program:

```
Test brace errors.
} line 2 closing brace is not paired
No braces.
   {opening brace is paired on same line}
No braces.
   {opening brace paired later
    No braces.
   } Closing brace paired two lines up.
{{{ Line 9. Three braces--only two paired.
  } First closing brace
 } Second closing brace.
End of program. One opening brace left.
```

ALGORITHM 3-9 Parse Parentheses

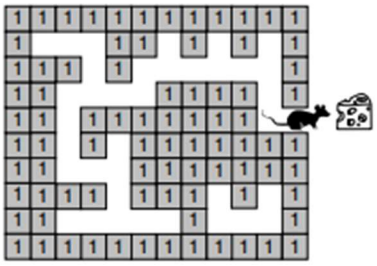
```
Algorithm parseParens
This algorithm reads a source program and parses it to make
sure all opening-closing parentheses are paired.
1  loop (more data)
   1   read (character)
   2   if (opening parenthesis)
       1   pushStack (stack, character)
   3   else
       1   if (closing parenthesis)
           1   if (emptyStack (stack))
               1   print (Error: Closing parenthesis not matched)
           2   else
               1   popStack(stack)
           3   end if
       2   end if
   4   end if
2  end loop
3  if (not emptyStack (stack))
   1   print (Error: Opening parenthesis not matched)
end parseParens
```

說明：

輸入一個檔案名稱判斷檔案內的大括弧是否配對，若是則印出”Valid”，否則印出所有未配對的大括弧在第幾行。可以用自己打的程式(.cpp or .c 檔)或者 12 題下面那個 small program 去測試自己的程式是否正確。Pseudocode 僅供參考可自行修改。

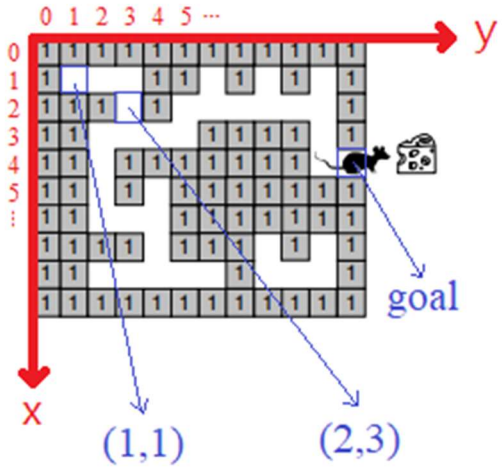
24. Write a program that simulates a mouse in a maze. The program must print the path taken by the mouse from the starting point to the final point, including all spots that have been visited and backtracked. Thus, if a spot is visited two times, it must be printed two times; if it is visited three times, it must be printed three times.

The maze is shown in Figure 3-25. The entrance spot, where the mouse starts its journey, is chosen by the user who runs the program. It can be changed each time.



說明：

輸入起點座標，模擬老鼠從起點走到終點的移動路徑，老鼠每次只能往上下左右其中一個方向移動一步，座標定義如下圖：



輸入的起始點為牆壁或超出範圍請印出”Invalid input”

```
Please input entrance spot : -1 0
Invalid input
```

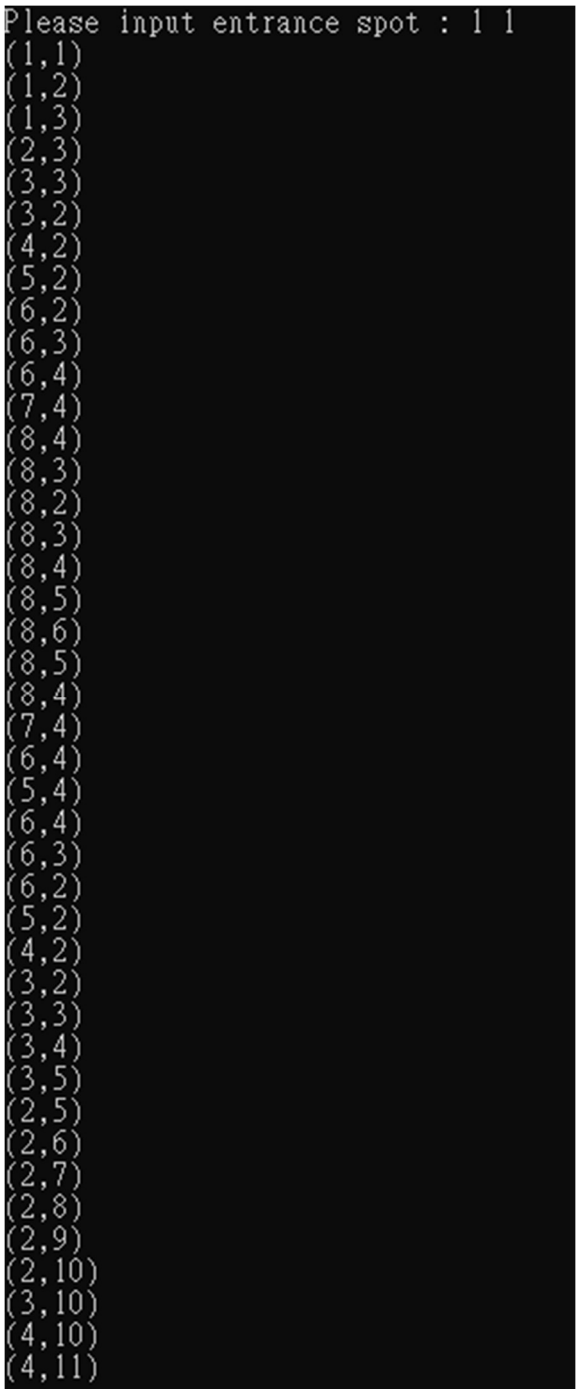
```
Please input entrance spot : 10 0
Invalid input
```

```
Please input entrance spot : 0 12
Invalid input
```

```
Please input entrance spot : 0 -1
Invalid input
```

```
Please input entrance spot : 0 0
Invalid input
```

答案正確的格式如下：



若無法到達終點請在走過所有可能路徑後再次回到起點時印出” Unable to reach”

問題與注意事項：

1. 地圖和終點固定，可寫死在程式裡或讀入” HW3_24_inputFile.txt”轉換成你所需要的格式。(讀取檔案時，被讀取的檔案請與執行檔放在同一個目錄下，請勿使用絕對位址)
2. 可以使用任何資料結構儲存你的地圖和計算你的答案，答案正確即可。
3. 輸入輸出請依照範例的格式，輸入為兩個數字，數字間以空白隔開，不用考慮輸入錯誤(非數字)或數字溢位等問題。
4. 輸出的答案可以是任何從起點開始到終點結束的合理路徑(每一步都在範圍內且非牆壁，任連續的兩步位置皆相鄰)。