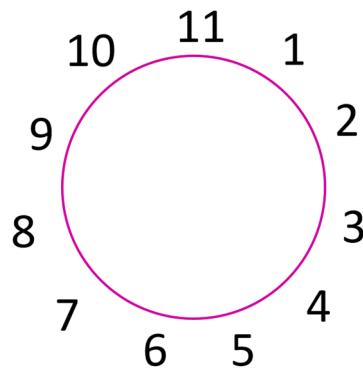


Homework 3

Moodle Submission Deadline: 2024/04/14 (Sunday) 23:59

Problem 1: Find the Last ID of Leaving a Round Table ([hw3_p1.py](#))

Assume there are n students who seat in the circle/round table. Each student is associated with an ID by order of the circle. An example with $n = 11$ is shown below.



Now the teacher asks these students to report the numbers 1, 2, and 3 repeatedly (1, 2, 3, 1, 2, 3, 1, 2, 3, ...) from student ID 1 in an ascending order. Students who report 3 is required to leave the table. Your task is to write a program that can allow the user to input the total number of students (n), and output the student ID who is the last one seating on the round table.

You will be able to practice the `while` loop and `list` data structure in this homework.

Sample Input/Output:

```
C:\Python35\workspace>python hw3_p1.py
Input the total number of students (n>0) : 11
The last ID is : 7

C:\Python35\workspace>python hw3_p1.py
Input the total number of students (n>0) : 200
The last ID is : 128

C:\Python35\workspace>python hw3_p1.py
Input the total number of students (n>0) : 3
The last ID is : 2
```

Problem 2: Evaluation of Polynomial Strings ([hw3_p2.py](#))

Write a program to evaluate the result of a polynomial string. Specifically, your program allows the user to input a string that represents a polynomial of variable X , and input the value of X . Then the result of evaluating the polynomial by putting the value of X will be printed out. Please refer to the following three examples. Note that if you totally use `print()`, you will get 0%. Besides, you cannot use the `eval()` function in this problem.

Example 1	Input polynomial: $X^5+3*X+2$ Input the value of X: 2 Evaluated Result: 40
Example 2	Input polynomial: $-X^3-12*X^2+100$ Input the value of X: 3 Evaluated Result: -35
Example 3	Input polynomial: $X^X+X*X+X$ Input the value of X: 5 Evaluated Result: 3155
Example 4	Input polynomial: $X^4+23*X^3+17*X^2+9453$ Input the value of X: -11 Evaluated Result: -4462

Sample Input and Output

```
C:\Python37\workspace>python hw3_p2.py
```

```
Input polynomial:  $X^5+3*X+2$ 
```

```
Input the value of X: 2
```

```
Evaluated Result: 40
```

```
C:\Python37\workspace>python hw3_p2.py
```

```
Input polynomial:  $-X^3-12*X^2+100$ 
```

```
Input the value of X: 3
```

```
Evaluated Result: -35
```

```
C:\Python37\workspace>python hw3_p2.py
```

```
Input polynomial:  $X^X+X*X+X$ 
```

```
Input the value of X: 5
```

```
Evaluated Result: 3155
```

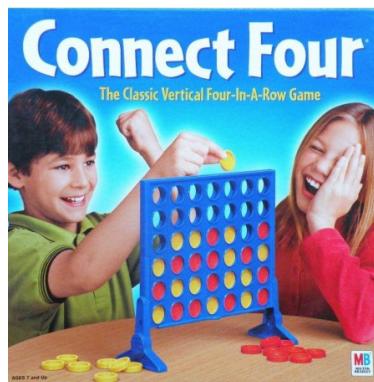
```
C:\Python37\workspace>python hw3_p2.py
```

```
Input polynomial:  $X^4+23*X^3+17*X^2+9453$ 
```

```
Input the value of X: -11
```

```
Evaluated Result: -4462
```

Problem 3: Game of Connect Four (hw3_p3.py)



https://lauamangud.ee/3979-large_default/connect-four-travel-edition.jpg

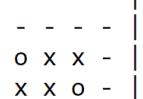
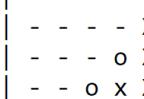
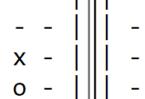
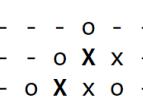
[[Wikipedia](#)] **Connect Four** is a two-player connection game in which the players first choose a color and then take turns dropping one colored disc from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the lowest available space within the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs. You can refer to the following animation of gameplay example provided by Wikipedia: https://en.wikipedia.org/wiki/Connect_Four#/media/File:Connect_Four.gif

You can also play the online Connect Four game: <https://www.mathsisfun.com/games/connect4.html>

Below we list the game rules of Connect Four:

- (1) There are two players with symbols “X” and “O”.
 - (2) The board game is a 6×7 two-dimensional grid where playing discs may be placed.
 - (3) The board game starts out empty.
 - (4) Player “X” plays first. Then play will alternate between the two players one round at a time.
 - (5) A playing disc is placed on the board by choosing a column into which to “drop” the disc. The disc starts in the highest row and then falls down to the lowest point available in that column.
 - (6) A column may become full and then no more playing discs may be dropped into that column.
 - (7) If all of the points on the board fill up without any player winning, then the game ends in a draw.

Here are the four possible end game examples.

			
1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
horizontal win	vertical win	diagonal win	draw

Besides the rules of Connect Four, your program is also required to do the following items.

- (1) Examine whether the column is full. If full, show "This column is full. Try another column."
 - (2) Examine whether the user input number is in the correct range (0-6),
If not in the correct range, show "Out of range, try again [0-6]."
 - (3) Examine whether the user input is valid (only numbers are valid).
If the input is not valid, show "Invalid input, try again [0-6]."
 - (4) If someone wins the game, show the lowercase symbol ("o" or "x") for all connected 4 discs.
If more than 4 connected discs or more than two connected lines, lowercase all of them.

Sample Input/Output:

Start of the game.

C:\Python35\workspace\2018計算機概論\hw3>hw3_p3.py

Player X >>

Allow each player to enter the column and show the updated table.

+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	X	O	+	+	+
0	1	2	3	4	5	6		

Player X >> 3

+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+
			X				
+	+	+	+	+	+	+	+
			X		O		
+	+	+	+	+	+	+	+
0	1	2	3	4	5	6	

Player 0 >>

+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
				X		X		
+	+	+	+	O	O	X	O	
+	+	+	+	+	+	+	+	+
0	1	2	3	4	5	6		

Player 0 >> 5

Player X >>

If invalid inputs or out of range, ask the user to input again.

+	+	+	+	+	+	+	-	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
							0	
+	+	+	+	+	+	+	+	+
				X			X	
+	+	+	+	+	+	+	+	+
			X	0		0		X
+	+	+	+	+	+	+	+	+
0	1	2	3	4	5	6		

```
Player X >> ahsknc  
Invalid input, try again [0-6].  
Player X >>
```

						X	
					O		
			X		X		
		X	O	O	X	O	
0	1	2	3	4	5	6	

```
Player 0 >> 68
Out of range, try again [0-6].
Player 0 >>
```

If the column is full, ask the user to input again.

Player X >> 2

+	+	+	+	+	-	+	+	+
		X						
+	+	+	+	+	+	+	+	+
		0	0	0	1			
+	+	+	+	+	+	+	+	+
		0	X	X	1			
+	+	+	+	+	+	+	+	+
		X	0	X	X	X		
+	+	+	+	+	+	+	+	+
0	X	0	0	0	1	X	0	
+	+	+	+	+	+	+	+	+
0	1	2	3	4	5	+	+	6

Player 0 >> 2
This column is full. Try another column.
Player 0 >>

If 4 discs are connected, need to lowercase them.
(Here are two examples: diagonal and vertically connected)

Player X >> 4

Winner: X

C:\Python35\workspace\2018計算機概論\hw3>

Player 0 > 3

+	+	+	+	+	+	+	+			
+	-	-	-	-	-	-	-			
+	-	-	-	-	-	-	-			
				0						
+	-	-	-	-	-	-	-			
	1	0	1	X	1	0	1	0		
+	-	-	-	-	-	-	-			
	1	X	1	X	1	0	1	X		
+	-	-	-	-	-	-	-			
	1	0	1	X	1	0	1	X	X	
+	-	-	-	-	-	-	-			

+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+
				o			
+	+	+	+	X	o	o	+
		X		X		X	
+	+	+	+	+	+	+	+
		o		X		X	
+	+	+	+	+	+	+	+
0	1	2	3	4	5	6	

Winner: O

C:\Python35\workspace\2018計算機概論\hw3>

If more than 4 discs get connected, lowercase all of its discs.

<pre>+-----+ +-----+ o +-----+ 0 X o X +-----+ X X X o 0 +-----+ X 0 X 0 o +-----+ X X X 0 0 X o +-----+ 0 1 2 3 4 5 6</pre> <p>Winner: 0</p>	<pre>+-----+ +-----+ +-----+ 0 +-----+ 0 x x x x x +-----+ X 0 0 X 0 0 0 +-----+ 0 1 2 3 4 5 6</pre> <p>Winner: X</p>
---	---

If multiple 4-disc lines, lowercase all of them. (Here are two examples)

<pre>+-----+ +-----+ +-----+ 0 +-----+ 0 o o o o o X +-----+ X X o X X 0 X +-----+ X o X X 0 X 0 +-----+ o X X X 0 0 0 +-----+ 0 1 2 3 4 5 6</pre> <p>Winner: 0</p>	<pre>+-----+ +-----+ +-----+ 0 +-----+ 0 x x x x x +-----+ X 0 0 X 0 0 0 +-----+ 0 1 2 3 4 5 6</pre> <p>Winner: X</p>
---	---

An example of draw (平手).

<pre>Player 0 >> 6 +-----+ 0 0 X 0 0 X 0 +-----+ X X 0 X X 0 X +-----+ 0 0 X 0 0 X 0 +-----+ X X 0 X X 0 X +-----+ 0 0 X 0 0 X 0 +-----+ X X 0 X X 0 X +-----+ 0 1 2 3 4 5 6</pre> <p>Draw</p>	<pre>C:\Python35\workspace\2018計算機概論\hw3></pre>
--	--

Here are more samples of inputs and outputs. Please visit the following two videos.

https://www.dropbox.com/s/8wz73ln9kim55qq/hw3_p3_ex1.mp4?dl=0

https://www.dropbox.com/s/1lka7t7y8m694v9/hw3_p3_ex2.mp4?dl=0

Problem 4: Replacing Number in a Matrix ([hw3_p4.py](#))

Given a matrix representing an image entered by a user, in which each $\langle i, j \rangle$ index pair indicates the pixel location in the image and is associated with a color value z . The user is also allowed to input a pair of targeted location $\langle x, y \rangle$ and a target color value k . Your task is to replace the color z of the given pixel $\langle x, y \rangle$ and all of its adjacent (excluding diagonally adjacent) same colored z pixels with the given target color k . Note that the index of the most left-top pixel is $\langle 0,0 \rangle$.

Sample Input and Output

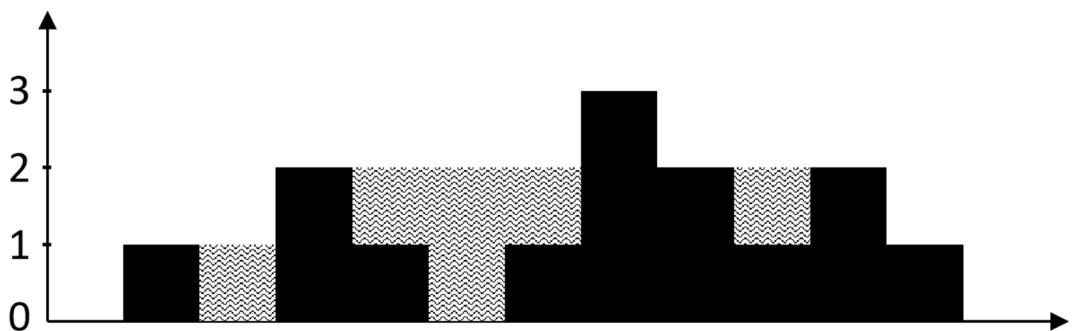
```
C:\Python37\workspace>python hw3_p4.py
Enter index x, y, k (separated by whitespace): 0 0 5
Enter the matrix by multiple lines:
0 0 0
0 1 1
2 2 0
q
5 5 5
5 1 1
2 2 0

C:\Python37\workspace>python hw3_p4.py
Enter index x, y, k (separated by whitespace): 3 1 7
Enter the matrix by multiple lines:
2 1 2 2
2 0 2 0
1 0 2 0
1 2 2 2
q
2 1 7 7
2 0 7 0
1 0 7 0
1 7 7 7

C:\Python37\workspace>python hw3_p4.py
Enter index x, y, k (separated by whitespace): 4 4 3
Enter the matrix by multiple lines:
0 0 1 1 1 2 2 2
1 0 1 1 1 1 0 0
1 0 0 1 1 0 1 1
2 2 2 2 2 0 1 0
1 1 1 2 2 0 2 2
1 2 1 2 2 2 2 0
2 1 1 1 1 2 1 1
2 1 1 1 1 2 2 1
q
0 0 1 1 1 2 2 2
1 0 1 1 1 1 0 0
1 0 0 1 1 0 1 1
3 3 3 3 3 0 1 0
1 1 1 3 3 0 3 3
1 2 1 3 3 3 3 0
2 1 1 1 1 3 1 1
2 1 1 1 1 3 3 1
```

Problem 5: Water Filling ([hw3_p5.py](#))

Given a sequence of n non-negative integers specified by the user, which stands for the land height above sea level, where the width of each bar is 1. Your task is to find how many units of water can be filled at most based on the input integer sequence. Take the following figure as an example for the integer sequence [0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1], which are represented by black blocks. Water can be filled in are shown by shaded blocks. In this example, at most 6 units of water can be trapped.



Sample Input and Output

```
c:\Python37\workspace>python hw3_p5.py
Input sequence of seats: 0 1 0 2 1 0 1 3 2 1 2 1
Water: 6

c:\Python37\workspace>python hw3_p5.py
Input sequence of seats: 5 0 3 4 5 6 0 1 2 0 1 0 4
Water: 28

c:\Python37\workspace>python hw3_p5.py
Input sequence of seats: 0 0 1 2 3 2 1 0 0
Water: 0
```

Note

This is a homework for each **individual**. 必須於程式檔內註解註明系及姓名學號。

How to Submit Your Homework?

Submission in NCKU Moodle

Before submitting your homework, please zip the files in a zip file, and name the file as “學號_hw3.zip”. For example, if your 學號 is H12345678, then your file name is:

“H12345678_hw1.zip” or “H12345678_hw1.rar”

When you zip your files, please follow the instructions provided by TA's slides to submit your file using NCKU Moodle platform <http://moodle.ncku.edu.tw> .

Have Questions about This Homework?

Please feel free to visit TAs, and ask/discuss any questions in their office hours. We will be more than happy to help you.