

Programming Exam 2

Time: 2018.11.27 10:10 – 12:00

(將程式檔上傳至 Moodle [學號_quiz2_p1.py](#)、[學號_quiz2_p2.py](#)、[學號_quiz2_p3.py](#)；
將程式碼複製到 [學號_quiz2.docx](#) [打上姓名系級學號、清楚標示題號]，並列印紙本給助教。)

Note: You are required to write comments (註解) for each part in your code.

In the following four exam problems, you will use string functions, e.g. `split()`, `if-else`, `while` loops and nested `while` loops, lists and nested lists, and conversion between strings and integers using `int()` and `str()`.

Problem 1

[學號_quiz2_p1.py](#)

Please follow the requirements of the following consecutive sub-problems to print the results.

- Allow the user to enter a square-number length integer x. (square-length is $2^2=4$, $3^2=9$, $4^2=16$)
- Same as (a), but you need to further examine whether the input is square-number length.
- Convert digits of x into integers, put them into a 1-dim list in order, and print the list.
- Compute the product of neighboring numbers in the 1-dim list, and put the resulting list.
- Convert digits of x into integers, put them into a 2-dim list (square matrix), and print the list.
- Add zeros surrounding the 2-dim list (i.e., zero padding), and put the new 2-dim list.
- Compute the summation of neighboring numbers in the new 2-dim list, and print the list.

Sample Input and Output

```
c:\Python37\workspace>python quiz2_p1.py
(a)+(b): Enter a positive integer with square-number length: 123456
Not square-number length. Try again!
(a)+(b): Enter a positive integer with square-number length: 123456789

(c) Put numbers into a 1-dim list, and print out.
[1, 2, 3, 4, 5, 6, 7, 8, 9]

(d) Compute the neighbor product in 1-dim list, and print out.
[2, 6, 24, 60, 120, 210, 336, 504, 72]

(e) Put numbers into a 2-dim list, and print out.
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

(f) Zero padding to the 2-dim list, and print out.
[0, 0, 0, 0, 0]
[0, 1, 2, 3, 0]
[0, 4, 5, 6, 0]
[0, 7, 8, 9, 0]
[0, 0, 0, 0, 0]

(g) Compute the neighbor summation in 2-dim list, and print out.
[7, 11, 11]
[17, 25, 23]
[19, 29, 23]
```

```
c:\Python37\workspace>python quiz2_p1.py
(a)+(b): Enter a positive integer with square-number length: 92839271
Not square-number length. Try again!
(a)+(b): Enter a positive integer with square-number length: 928392712

(c) Put numbers into a 1-dim list, and print out.
[9, 2, 8, 3, 9, 2, 7, 1, 2]

(d) Compute the neighbor product in 1-dim list, and print out.
[18, 144, 48, 216, 54, 126, 14, 14, 2]

(e) Put numbers into a 2-dim list, and print out.
[9, 2, 8]
[3, 9, 2]
[7, 1, 2]

(f) Zero padding to the 2-dim list, and print out.
[0, 0, 0, 0, 0]
[0, 9, 2, 8, 0]
[0, 3, 9, 2, 0]
[0, 7, 1, 2, 0]
[0, 0, 0, 0, 0]

(g) Compute the neighbor summation in 2-dim list, and print out.
[14, 28, 12]
[28, 17, 21]
[11, 19, 5]
```

```

C:\Python35\workspace\2018計算機概論>quiz2_p1.py
(a)+(b): Enter a positive integer with square-number length: 102938475647382
Not square-number length. Try again!
(a)+(b): Enter a positive integer with square-number length: 1029384756473820

(c) Put numbers into a 1-dim list, and print out.
[1, 0, 2, 9, 3, 8, 4, 7, 5, 6, 4, 7, 3, 8, 2, 0]

(d) Compute the neighbor product in 1-dim list, and print out.
[0, 0, 0, 54, 216, 96, 224, 140, 210, 120, 168, 84, 168, 48, 0, 0]

(e) Put numbers into a 2-dim list, and print out.
[1, 0, 2, 9]
[3, 8, 4, 7]
[5, 6, 4, 7]
[3, 8, 2, 0]

(f) Zero padding to the 2-dim list, and print out.
[0, 0, 0, 0, 0, 0]
[0, 1, 0, 2, 9, 0]
[0, 3, 8, 4, 7, 0]
[0, 5, 6, 4, 7, 0]
[0, 3, 8, 2, 0, 0]
[0, 0, 0, 0, 0, 0]

(g) Compute the neighbor summation in 2-dim list, and print out.
[4, 11, 15, 18]
[17, 21, 25, 27]
[17, 31, 23, 18]
[16, 19, 14, 9]

```

Problem 2

[學號 quiz2_p2.py](#)

Given a $m \times n$ matrix specified by the user, if an element is 0, set its entire row and column to 0. In this problem, you are also asked to meet the input requirement: enter multiple lines of integers separated by whitespaces until the character 'q' is provided. In addition, your program should output "Invalid matrix" if the lengths of some lines are different from other lines.

Sample Input and Output

<pre> c:\Python37\workspace>python quiz2_p2.py Enter the matrix by multiple lines: 1 1 1 1 0 1 1 1 1 q 1 0 1 0 0 0 1 0 1 c:\Python37\workspace>python quiz2_p2.py Enter the matrix by multiple lines: 1 2 3 4 5 0 1 0 3 6 1 1 1 1 q Invalid matrix. </pre>	<pre> c:\Python37\workspace>python quiz2_p2.py Enter the matrix by multiple lines: 1 0 3 5 2 7 2 1 1 2 3 4 0 1 6 0 3 2 1 5 2 8 3 3 2 3 1 5 2 9 1 2 5 1 8 1 q 0 0 0 0 0 0 0 0 1 0 3 4 0 0 0 0 0 0 0 0 2 0 3 3 0 0 1 0 2 9 0 0 5 0 8 1 </pre>
---	--

Bonus 1學號 quiz2_b1.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 $\langle 0, 0 \rangle$. The input matrix format is the same as Problem 2.

Sample Input and Output

```
c:\Python37\workspace>python quiz2_b2.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 quiz2_b2.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 quiz2_b2.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
```