**Week 7 In-Class Exercises (Extra)**Part I: Strings and Loops

**Q1: Encode Strings [ \*\*\* ]**

Write a function called encode\_message() that takes in a string as its only parameter. Call this parameter text. You can assume that text doesn’t contain any space. The function ***returns*** a string that encodes text in the following way: for each segment of consecutive characters that are the same in text, the segment will be encoded as the character followed by the number of times it appears in the segment and a space. For example, 'aaa' is encoded as 'a3 ', 't' is encoded as 't1 '. The entire string text is encoded by encoding all its segments.

For example,

* encode\_message('aaabbcccccde') returns 'a3 b2 c5 d1 e1'.
* encode\_message('112333&&$9999999999') returns '12 21 33 &2 $1 910'.
* encode\_message('') returns ''.

**Q2: Print Patterns**

1. [ \*\*\* ] Define a function called print\_diamond(). The function takes in an integer n as its parameter. You can assume that n is at least 2. The function prints a diamond shape where each side of the diamond has exactly n '\*'.

For example,

* print\_diamond(4) prints

\*  
 \* \*  
 \* \*  
\* \*  
 \* \*  
 \* \*  
 \*

* print\_diamond(2) prints

\*  
\* \*  
 \*

1. [ \*\*\* ] Define a function called print\_diamond\_using\_str(). The function takes in a string text as its parameter. The function prints a diamond shape using characters from text sequentially, starting from the top of the diamond.

You can assume that

* text doesn’t contain any space,
* text has at least 4 characters,
* the length of text is a multiple of 4.

For example,

* print\_diamond('12345678') prints

1  
 2 8  
 3 7  
 4 6  
 5

* print\_diamond('opqrstuvwxyz') prints

o  
 p z  
 q y  
r x  
 s w  
 t v  
 u

1. [ \*\*\* ] Define a function called print\_squares(). The function takes in an integer n as its parameter. You can assume that n is at least 3 and (n-3) is always a multiple of 4. The function prints out the following kind of pattern, which always consists of squares.

For example,

* print\_squares(3) prints

\*\*\*  
\* \*  
\*\*\*

* print\_squares(7) prints

\*\*\*\*\*\*\*  
\* \*  
\* \*\*\* \*  
\* \* \* \*  
\* \*\*\* \*  
\* \*  
\*\*\*\*\*\*\*

* print\_squares(11) prints

\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* \*\*\*\*\*\*\* \*  
\* \* \* \*  
\* \* \*\*\* \* \*  
\* \* \* \* \* \*  
\* \* \*\*\* \* \*  
\* \* \* \*  
\* \*\*\*\*\*\*\* \*  
\* \*  
\*\*\*\*\*\*\*\*\*\*\*