Lab 4 - More on Strings and Loops

Q1: Print Patterns

(a) [***]

Define a function called `print_square_1()`. The function takes in a single parameter `n`, which you can assume is a positive odd number greater than or equal to 5. The function prints out an $n \times n$ square using asterisks that's divided into four small squares. See the examples below to understand the behaviour of the function:

- `print_square_1(5)` displays the following:

```
*****
* * *
*****
* * *
*****
```

- `print_square_1(9)` displays the following:

```
*********
*   *   *
*   *   *
*   *   *
*********
*   *   *
*   *   *
*   *   *
*********
```

(b) [***]

Define a function called `print_square_2()`. The function takes in a single parameter `n`, which you can assume is a positive odd number greater than or equal to 5. The function prints out a square using asterisks that's divided into four triangles. See the examples below to understand the behaviour of the function:

- `print_square_2(5)` displays the following:

```
*****
** **
* * *
** **
*****
```

- `print_square_2(9)` displays the following:

```
*********
**     **
* *   * *
*  * *  *
*   *   *
*  * *  *
* *   * *
**     **
*********
```

Q2: Longest Subsequence of Letters [***]

Define a function called `get_longest_subsequence()`. The function takes in a single parameter, which is a string. The function should return the longest subsequence from the input string that consists of only the letters from the English alphabet (both uppercase and lowercase letters included). If there are multiple subsequences of the same longest length, the function returns the first one. If the input string doesn't contain any letter, the function returns an empty string.

For example,

- `get_longest_subsequence('ab24[AaBbCDExy0longest$]')` should return `'AaBbCDExy'`.
- `get_longest_subsequence('a a a1234b|c|d ')` should return `'a'`.
- `get_longest_subsequence('12345 ')` should return `''`.