### Question 1. [5 marks]

For each code fragment in the table below, select the answer that best describes the printed output, or the error that occurs when the code is run.

<table>
<thead>
<tr>
<th>Code Fragment</th>
<th>Answer choices</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>diff = float(2) - int(str(2))</code> &lt;br&gt;<code>print(diff != 0)</code></td>
<td>(A) True &lt;br&gt;(B) False &lt;br&gt;(C) an error occurs converting one of the arguments</td>
</tr>
<tr>
<td><code>phrase = ['All', 'the', 'chocolate']</code> &lt;br&gt;<code>phrase.insert(3, 'now') + ['please']</code> &lt;br&gt;<code>print(phrase)</code></td>
<td>(A) ['All', 'the', 'chocolate'] &lt;br&gt;(B) ['All', 'the', 'chocolate', 'now'] &lt;br&gt;(C) ['All', 'the', 'chocolate', 'please'] &lt;br&gt;(D) ['All', 'the', 'chocolate', 'now', 'please'] &lt;br&gt;(E) some other list is printed &lt;br&gt;(F) an error occurs because + is not defined for lists &lt;br&gt;(G) an error occurs because + is not defined for non-list and list</td>
</tr>
<tr>
<td><code>print('CAT!'.isupper() or 1 / 0 == 5)</code></td>
<td>(A) True &lt;br&gt;(B) False &lt;br&gt;(C) a ZeroDivisionError occurs &lt;br&gt;(D) another error occurs</td>
</tr>
<tr>
<td><code>L = ['for', 'this', 'was', 'on', 'seynt', 'Volantynys', 'day']</code> &lt;br&gt;<code>chaucer = ''</code> &lt;br&gt;<code>for i in range(len(L) // 3):</code> &lt;br&gt; &lt;br&gt; <code>chaucer = chaucer + L[2 * i][i]</code> &lt;br&gt;<code>print(chaucer)</code></td>
<td>(A) fa &lt;br&gt;(B) fi &lt;br&gt;(C) or &lt;br&gt;(D) fay &lt;br&gt;(E) ftw &lt;br&gt;(F) for &lt;br&gt;(G) ftwosVd &lt;br&gt;(H) some other characters are printed &lt;br&gt;(I) an error occurs inside the loop body</td>
</tr>
<tr>
<td><code>s = 'Je suis desja d'amour tanne'</code> &lt;br&gt;<code>print(s[s.find('j') : s.find('j') + 6])</code></td>
<td>(A) Je sui &lt;br&gt;(B) Je suis desja d'a &lt;br&gt;(C) jad'am &lt;br&gt;(D) ja d' &lt;br&gt;(E) ja d'a &lt;br&gt;(F) ja d'am &lt;br&gt;(G) some other characters are printed &lt;br&gt;(H) an error occurs during the assignment statement</td>
</tr>
</tbody>
</table>

Question 2. [4 marks]

Part (a) [2 marks]
You need to write a function that returns the number of characters that two strings have in common. Both strings are parameters to the function. Do Step 1 of the Function Design Recipe: write two example function calls and their expected results. As always, choose a good name for your function.
You do not need to write any other steps of the Function Design Recipe.

```python
>>> count_overlap('abc', 'abd')
2
>>> count_overlap('abc', 'def')
0
```}

Part (b) [2 marks]
You’ve decided you want to write a function that capitalizes the first number of characters in a string, where the string is made up only of alphabetic characters. Below are some example calls such as you might produce during Step 1 of the Function Design Recipe. Fill in the function header, including the type contract. As always, select good parameter names.

You do not need to write a description. You do not need to write the function body.

```python
def capitalize_first_letters(s: str, count: int) -> str

```
Question 3.  [4 marks]

Complete the following function according to its docstring.

```python
def bagel_order(bagel_type: str, cream_cheese: str, toasted: bool) -> str:
    """Return the bagel order with the given bagel_type, cream_cheese, and
    whether or not it is toasted.

    If toasted is False, the format is as follows:
    <bagel_type> bagel with <cream_cheese> cream cheese

    If toasted is True, the format is as follows:
    <bagel_type> bagel toasted with <cream_cheese> cream cheese

    If cream_cheese is '', then use 'regular'.
    If cream_cheese is 'no', then omit the last part of the string.

    There should exactly one space between each word in the order, and no
    extra leading or trailing spaces.
    """

    s = bagel_type + ' bagel'
    if toasted:
        s = s + ' toasted'
    if cream_cheese != 'no':
        if cream_cheese == '':
            cream_cheese = 'regular'
        s = s + ' with ' + cream_cheese + ' cream cheese'
    return s
```

Testing:

```python
>>> bagel_order('poppy seed', '', True)
'poppy seed bagel toasted with regular cream cheese'
>>> bagel_order('everything', 'light', False)
'everything bagel with light cream cheese'
>>> bagel_order('plain', 'no', True)
'plain bagel toasted'
```
Question 4.  [3 marks]

Fill in the box with the while loop condition required for the function to work as described in its docstring.

def find_digit(word: str) -> int:
    """Return the index of the first digit character in word, or the length of word
    if it does not contain any digit characters.
    >>> find_digit('!Ba4262')
    3
    >>> find_digit('123Hello')
    0
    >>> find_digit('cats!')
    5
    """
    i = 0
    while
        i = i + 1
    return i

i = 0
while i < len(word) and not word[i].isdigit():
    i = i + 1
return i

Question 5.  [5 marks]

Complete the function body below according to its docstring. Hint: consider using range on your answer.

def has_pair(s: str) -> bool:
    """Return True if and only if s has 2 consecutive characters
    (i.e., next to each other) that are the same, and False otherwise.
    >>> has_pair('programming!')
    True
    >>> has_pair('Llama')
    False
    """
    for i in range(len(s) - 1):
        if s[i] == s[i+1]:
            return True
    return False
Question 6.  [3 marks]

Complete the following function according to its docstring, without using the method str.count.

```python
def count_alphanumeric(s: str) -> float:
    """Return the percentage of characters in s that are alphanumeric (letters or digits).
    The percentage should be between 0.0 and 1.0.
    """
    num_alnum = 0
    for ch in s:
        if ch.isalnum():
            num_alnum = num_alnum + 1
    return num_alnum / len(s)
```