Question 1.  

Part (a)  [1 mark] Write the output of the code below in the box below it.

```python
x = 3
y = x - 1
x = 4
print(x, y)
```

Solution:  4 2

Part (b)  [1 mark] Write the output of the code below in the box below it.

```python
s = '0123456789'
i = 0
while i < len(s) and int(s[i]) < 2:
    print(s[i])
i = i + 1
```

Solution:
0
1

Part (c)  [1 mark] Fill in the box with Python code that will make the program behaviour match the comments. You may not make any other changes to the code or add code outside the box.

```python
def check_carry_on(weight, liquid_free):
    ''' (float, bool) -> bool
    Return True if and only if the bag’s weight is less than 22
    and the bag does not contain liquid as indicated by liquid_free.'''
    return weight < 22 and liquid_free
```

Solution:  return weight < 22 and liquid_free

Part (d)  [1 mark] Fill in the box with Python code that will make the program behaviour match the comments. You may not make any other changes to the code or add code outside the box.

```python
s1 = "computer"
s2 = "midterm"

# Using only s1, s2, concatenation, and indexing and/or slicing, print the string 'code'.
print(s1[0:2] + s2[2] + s2[4])
```

There are many other correct answers.
Question 2.  [6 marks]

Part (a)  [4 marks]

For the function below, complete the function header (using a meaningful function name) and write a
docstring (including the type contract, description and two examples).

```python
def duplicate_digits(s):
    ''' (str) -> str
    Return a copy of s where each digit now occurs twice.
    >>> duplicate_digits('ab79c88')
    'ab7799c8888'
    >>> duplicate_digits('747')
    '774477'
    '''
    result = ''
    for ch in s:
        if ch.isdigit():
            result = result + ch + ch
        else:
            result = result + ch
    return result
```

Part (b)  [2 marks]

Write a Python program that prompts the user to enter a string, calls the function from Part (a) passing
the string that the user entered as an argument, and prints the value returned by the function. You may
not change the function.

SOLUTION: The code will be different depending on what the student has named the function.

```python
user_string = input('please enter a string')
result = duplicate_digits(user_string)
print(result)
```
Question 3.  [10 marks]

Part (a)  [5 marks] Complete the function according to its docstring.

VOWELS = 'aeiou'
HIDDEN = '~'

def points_for_hidden_consonants(puzzle, view, points_by_position):
    ''' (str, str, str) -> int

    Return the points for the hidden consonants for this view of puzzle.
    Points are awarded based on the position of the hidden consonants.
    points_by_position is a string of digits that define the number of points
    awarded for a hidden consonant in the corresponding position of the view.
    Hidden vowels do not earn points.
    
    >>> points_for_hidden_consonants('apple', '^^^^e', '31211')
    4
    >>> points_for_hidden_consonants('cat', '^at', '311')
    3
    '''
    score = 0
    for i in range(len(view)):
        if view[i] == HIDDEN and puzzle[i] not in VOWELS:
            score += int(points_by_position[i])
    return score
**Part (b)**  [5 marks] A piano teacher says that students in grade 5 and up should practice 10 minutes per song daily, but students in grades 4 and below only need to practice 5 minutes per song. But the teacher also has a daily limit on the total time spent practicing and says to stop when the limit is reached, even if the student has not finished practicing each individual song for the recommended amount of time. Her daily limit is as follows:

- students in grades 1-2: 15 minute limit
- students in grades 3-4: 30 minute limit
- students in grade 5 and up: 60 minute limit

Complete the following function according to the description above and the docstring below.

```python
def practice_needed(num_songs, grade_level):
    ''' (int, int) -> int
    Return the number of minutes of practice needed per day according to this teacher for num_songs at the level grade_level.
    >>> practice_needed(5, 4) # 5 songs at grade 4 level
    25
    >>> practice_needed(4, 2) # 4 songs at grade 2 level
    15
    '''

    if grade_level > 4:
        minutes_per_song = 10
    else:
        minutes_per_song = 5

    total_time = num_songs * minutes_per_song

    if grade_level < 3:
        cap = 15
    elif grade_level < 5:
        cap = 30
    else:
        cap = 60
    return min(cap, total_time)
```

```python
>>> practice_needed(5, 4) # 5 songs at grade 4 level
25
>>> practice_needed(4, 2) # 4 songs at grade 2 level
15
```