Question 1. [4 marks]

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

```python
x = 4
y = x + 5
x = 7
print(x, y)
```

Solution: 7 9

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

```python
s = 'coMPuTer'
i = 0
while i < len(s) and not s[i].isupper():
    print(s[i])
i = i + 1
```

Solution:
c
   o

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 can_enter(height, with_adult):
    ''' (float, bool) -> bool
    Return True if and only if the child’s height is greater than 0.92 or
    the child is with an adult as indicated by with_adult.'''
    return height > 0.92 or with_adult
```

Solution: return height > 0.92 or with_adult

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 = "midterm"
s2 = "science"

# Using only s1, s2, concatenation, and indexing and/or slicing, print the string 'test'.
```

Solution: print(s1[3:5] + s2[0] + s1[3]) There are other possibilities as well.
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 replace_non_digits(s):
    ''' (str) -> str
    Return a copy of s where all non-digit characters have been replaced by hyphens.
    >>> replace_non_digits('416.946.8488')
    '416-946-8488'
    >>> replace_non_digits('ABC123')
    '---123'
    '''
    result = ''
    for ch in s:
        if ch.isdigit():
            result = result + ch
        else:
            result = 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 = replace_non_digits(user_string)
print(result)
```
**Question 3.** [10 marks]

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

```
HIDDEN = '\''

def merge_views(view1, view2):
    ''' (str, str) -> str

    view1 and view2 are views of the same puzzle. Some of the alphabetic
    characters may be revealed and some may be hidden (that is, replaced
    by HIDDEN). Return a view of the puzzle where the revealed characters are
    those that are revealed in at least one of view1 and view2.
    
    >>> merge_views('^^^l^', 'a^^^e')
    'a^^le'
    >>> merge_views('^ppl^', '^pp^e')
    '^pple'
    ''
    
    merged_view = ''
    for i in range(len(view1)):
        if view1[i] != HIDDEN:
            merged_view = merged_view + view1[i]
        else:
            merged_view = merged_view + view2[i]
    return merged_view
```

Part (b) [5 MARKS] The estimated cooking time for potatoes is usually 5 minutes per potato, but if the potatoes are extra large, it is 7 minutes per potato. When cooking more than 2 potatoes of any size, the cooking time calculated is too long, so 3 minutes is deducted from the estimated cooking time. (All potatoes being cooked together are the same size.)

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

```python
def cooking_time(num_potatoes, extra_large):
    ''' (int, bool) -> int
    Return the estimated time in minutes to cook num_potatoes potatoes depending on whether or not they are extra_large.
    >>> cooking_time(5, True)  # 5 extra large potatoes
    32
    >>> cooking_time(6, False)  # 6 regular potatoes
    27
    >>> cooking_time(2, False)  # 2 regular potatoes
    10
    '''
    minutes_per_potato = 7 if extra_large else 5
    minutes = minutes_per_potato * num_potatoes
    return minutes - 3 if num_potatoes > 2 else minutes
```