Question 1.  [4 marks]

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)
print(y)
```

Solution:

```
4
2
```

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

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

Solution:

```
r
e
```

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 within_capacity(num_passengers, within_limit):
    ''' (int, bool) -> bool
    Return True if and only if an airplane has fewer than 10 passengers
    and cargo weight is within the limit as indicated by within_limit.'''
    return num_passengers < 10 and within_limit
```

Solution: `return num_passengers < 10 and within_limit`

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 = "science"

# Using only s1, s2, concatenation, and indexing and/or slicing, print the string 'test'.
print(s1[-3:-1] + s2[0] + s1[5])
```
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 invert_case(s):
    ''' (str) -> str
    Return a copy of s where all uppercase characters are converted to lowercase
    and all lowercase characters are converted to uppercase.
    >>> invert_case('abCd9ef')
    'ABcD9EF'
    >>> invert_case('CDE fgh')
    'cde FGH'
    '''
    result = ''
    for ch in s:
        if ch.isupper():
            result = result + ch.lower()
        else:
            result = result + ch.upper()
```

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 = invert_case(user_string)
print(result)
```
Question 3.  [10 marks]

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

```python
def select_characters(s1, s2, selection):
    ''' (str, str, str) --> str

    Return a new string where each character is a character from either s1 or s2
    selected based on selection. selection is made up of 1s and 2s and
    indicates whether the character at the corresponding position of the new
    string should be from that position in s1 or that position in s2.
    s1, s2 and selection are the same lengths.

    >>> select_characters('coat', 'hard', '1221')
    'cart'
    >>> select_characters('pizza', 'hints', '11222')
    'pints'
    '''

    result = ''
    for i in range(len(selection)):
        if selection[i] == '1':
            result = result + s1[i]
        else:
            result = result + s2[i]
    return result
```

```python
>>> select_characters('coat', 'hard', '1221')
'cart'
>>> select_characters('pizza', 'hints', '11222')
'pints'
```
Part (b) [5 marks] A library charges overdue fees for a borrowed book using the following fee schedule:

- less than 4 days late: 1 dollar per day
- 4 to 6 days late: 2 dollars per day (for all days, including the first 3 days)
- more than 6 days late: 3 dollars per day (for all days, including the first 6 days)

Borrowers of books are in one of these age groups: CHILD, ADULT or SENIOR. A CHILD gets charged only half of the fees and a SENIOR gets charged only one quarter of the fees. An ADULT pays the full fee.

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

```
CHILD = 'child'
ADULT = 'adult'
SENIOR = 'senior'

def overdue_fees(days_late, age_group):
    ''' (int, str) -> number
    Return the fees for a book that is days_late days late for a borrower
    in the age group age_group.
    >>> overdue_fees(2, SENIOR) # 2 days late, SENIOR borrower
    0.5
    >>> overdue_fees(5, ADULT) # 5 days late, ADULT borrower
    10
    '''
    if days_late < 4:
        cost_per_day = 1
    elif days_late <= 6:
        cost_per_day = 2
    else:
        cost_per_day = 3

    fees = days_late * cost_per_day

    if age_group == CHILD:
        fees = fees / 2
    elif age_group == SENIOR:
        fees = fees / 4

    return fees
```