Given the following Python code:

```python
def mars_explore_main(num_trips):
    '''()
    - None
    main function for mars_explore:
    set up print and graphical output
    then call mars_explore repeatedly
    data is printed; no value is returned
    >>> mars_explore_main(100)
    ...
    # label for print output
    print("xpos", "\t",
          "ypos", "\t",
          "water", "\t",
          "temp")

    # set up graphical output
    reset(); speed(0)
    title("Mars Rover")
    display_color = "blue"
    color(display_color)
    dot(20, 'blue') # mark the rover start position

    # explore Mars
    n = 0
    while n < num_trips:
        mars_explore()
        n += 1

    return #None

>>> mars_explore_main(100)
```

(a) How many times will function mars_explore be called?
100

(b) What value is returned?
None
(1) (12 pts.) Given the following Python function:

```python
def transcribe(dna):
    '''(str) -> str

    function describes transcription
    of input dna to messenger rna,
    which is returned

    >>> transcribe('ACGT TGCA')
    'UGCAACGU'
    >>> transcribe('GATTACA')
    'CUAAUGU'
    >>> transcribe('cs5')   # lowercase doesn't count
    ...

    ctr = 0
    rna = ''

    for ch in dna:
        while ctr < len(dna):
            if ch == 'A':
                rna += 'U'
            elif ch == 'C':
                rna += 'G'
            elif ch == 'G':
                rna += 'C'
            elif ch == 'T':
                rna += 'A'
                ctr += 1

    return rna
```

Identify the following:

(a) function header:
`def transcribe(dna):`

(b) parameter(s):
`dna`

(c) type contract:
`(str) -> str`

(d) Mark up the code given above to make the changes needed to change the while loop into a for loop. (see changes in code)
(2) (6 pts.) Given the following Python function:

def mystery(astring):
    """(str) -> Boolean
    ""
    ctr = 0
    for c in astring:
        if c.isdigit():
            ctr += 1
    return (ctr >= 3)

What is the result when the following code is executed?

(a) >>> mystery('CIS122')
True
(b) >>> mystery('122CIS')
True
(c) >>> mystery('UODUCKS')
False

(3) (8 pts.) Given the following Python code:

def wonder(word):
    """(str) -> ??
    ""
    check1 = 'Q' in word
    check2 = 'q' in word
    return check1 or check2

(a) What type of value does the function return? Boolean

(b) What is the result of executing the following code:

    >>> wonder('The quick brown fox')
    True

    >>> wonder('THE QUICK BROWN FOX')
    True

    >>> wonder('try this')
    False
(4) (10 pts.) Given the following Python function:

```python
def payday(hourly_wage, hours):
    '''(number, number) -> number

    Calculate and return paycheck amount based on hourly_wage and hours_worked. Overtime is time-and-a-half for hours over 40 but less than or equal to 60, and double-time for hours over 60.
    >>> payday(10, 35)
    350
    >>> payday(10, 45)
    475.0
    >>> payday(10, 61)
    720.0
    >>> payday(25, 0)
    0
    ...
    if hours <= 40:
        paycheck = hourly_wage * hours
    elif hours <= 60:
        regular_pay = hourly_wage * 40
        overtime = hours - 40
        overtime_pay = (1.5 * hourly_wage) * overtime
        paycheck = regular_pay + overtime_pay
    else: # more than 60 hours
        regular_pay = hourly_wage * 40
        overtime_pay = (1.5 * hourly_wage) * 20
        overtime = hours - 60
        more_overtime_pay = (2 * hourly_wage) * overtime
        paycheck = regular_pay + overtime_pay + \n            more_overtime_pay
        paycheck = round(paycheck, 2) # dollar format
    return paycheck
```

(a) Show the change(s) needed if double pay were for all hours over 50. (You do not have to actually make the changes, just circle what needs to be changed.)
(8 changes – marked in bold in code)

(b) Suggest an improvement to the original function that would have required fewer changes to the original code to implement the change:
# before the if statement
doubletime = 60 #[or 50]

# then use doubletime var name rather than literal 60 throughout the code
(5) (6 pts.) Given the following Python code:

```python
from turtle import *
def poly(num_sides, scolor):
    '''(int, str) -> None
    ...'''
    fillcolor(scolor)
    begin_fill()
    side = 100
    angle = 360 / num_sides
    turnctr = 0

    while turnctr < num_sides:
        fd(side)
        lt(angle)

    end_fill()

    return #None
```

(a) What is the result when `>>> poly(6, 'blue')` is executed?

infinite loop

(b) The code as written is missing a key line. Add the missing line:

```
turnctr += 1 #last line of the while loop block
```

(c) Now what is the result when `>>> poly(6, 'blue')` is executed?
(6) (4 pts.) What is the result when the following Python code is executed?

def any_upper(astring):
    ...
    for ch in astring:
        if ch.isupper():
            return True
        else:
            return False

(a) >>> any_upper('CIS122')
True

(b) >>> any_upper('122CIS')
False

(Challenge) If val is a Boolean variable, then the following statement:

val = (val == False)

has what effect?

It changes the value of val, i.e., if val was originally True, it will be False after the assignment statement is executed and if val was originally False, it will have the value True after the assignment statement is executed.