Question 1. [7 marks]

Beside each code fragment in the table below, write the printed output when the code fragment is executed. If the code would cause an error, instead write ERROR and give a brief explanation of why an error occurs.

Code	Output or Cause of Error
`for i in range(1, 6, 2):` `  print(i)`	1 3 5
`message = 'CS is fun'` `print(message[4])`	s
`print(0.5 + '2')`	ERROR - can't use + with float and str

`items = [1] + [2] + [3]`
`items = items.append(4)`
`items[1] = 5`
`print(items)`
	ERROR - can't index into NoneType

`print(2 + 2 == 4 or 4/0 == 1)`	True

`values = ['a', 'b']`
`for value in values:`
`  value = value.upper()`
`print(values)`
	['a', 'b']

`x = 5`
`if x > 2:`
`  print('mid')`
`elif x > 4:`
`  print('high')`
`else:`
`  print('low')`
	mid

Question 2. [4 marks]

Complete this function according to its docstring description.

```
def extract_first_name(listing):
    """ (str) -> str

    Return the first name from listing. listing will be formatted as either
    <last name>, <first name>  or <last name>, <first name>: <phone number>
    depending on if the person has a phone number.

    >>> extract_first_name('de Lara, Eyal')
    'Eyal'
    >>> extract_first_name('Smith, Jacqueline: (416) 123-4567')
    'Jacqueline'
    >>> extract_first_name('Campbell, Jennifer: 4169991234')
    'Jennifer'
    """



    if ':' in listing:
        return listing[listing.find(',') + 2:listing.find(':')]

    else:
        return listing[listing.find(',') + 2:]
```

Question 3. [5 marks]

In the function below, complete (i) a precondition the function requires, (ii) the function description, and (iii) the example function calls by adding arguments that result in the return values shown. Write your preconditions and description in the large box, and the example calls in the smaller boxes. For the example calls, there may be several correct answers, and providing any one of them will earn full marks.

```
def mystery(message):
    """ (str) -> str

    Precondition: len(message) > 0 and message contains at least one uppercase character

    Return a string of all the non-alphanumberic characters in message before
    the first uppercase letter.

    >>> mystery("hello, Cat")
    ', '
    >>> mystery('what? yes? NO!')
    '? ? '
    """

    result = ''
    i = 0
    while not message[i].isupper():
        if not message[i].isalnum():
            result = result + message[i]
        i = i + 1
    return result
```

Question 4. [4 marks]

Complete this function's body according to its docstring description.

```
def is_valid_collection(collection, size):
    """ (list of str, int) -> bool

    Return True iff every item in collection has exactly the length size.

    >>> is_valid_collection(['cat', 'dog', 'fox'], 3)
    True
    >>> is_valid_collection(['cat', 'dog', 'mouse'], 3)
    False
    """



    for item in collection:
        if len(item) != size:
            return False
    return True
```

Question 5. [4 marks]

Consider the following function definition and additional Python statements. Beside each Python statement in the table below, write what is printed when the statement is executed. Assume all of the code above the table is run first, and then each print statement is executed in the order listed.

```
def some_function(value, num_times):
    """ (int, int) -> int
    """

    result = 0
    for i in range(num_times):
        value = value + 1
        result = result + value
    return result

value1 = 5
result1 = some_function(value1, 2)
value2 = 2
result2 = some_function(value2, 3)
```

Code	Output
`print(value1)`	5
`print(result1)`	13
`print(value2)`	2
`print(result2)`	12