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
`message = 'CSC 108'` `print(message[4])`	`1`
`print(1 + 4 == 5 or 4/0 == 1)`	`True`

`values = [3.1, 2.7]`
`for value in values:`
`  value = value - 1`
`print(values)`
	`[3.1, 2.7]`

`print('1.5' + 2.0)`	`ERROR - can't concatenate str and float`

`for i in range(0, 5, 2):`
`  print(i)`
	`0`
	`2`
	`4`

`size = 25`
`if size >= 10:`
`  print('medium')`
`elif size >= 20:`
`  print('large')`
`else:`
`  print('small')`
	`medium`

`items = [0] + [2] + [4]`
`items = items.append(1)`
`items[1] = 3`
`print(items)`
	`ERROR - can't index into NoneType`

Question 2. [4 marks]

Complete this function according to its docstring description.

```
def extract_clothing_type(item):
    """ (str) -> str

    Return the type of clothing from item, which is formatted as
    <name>: <type> or <name>: <type>, <colour> depending on if the colour is
    known.

    >>> extract_clothing_type('Jacqueline Smith: jacket')
    'jacket'
    >>> extract_clothing_type('Paul: pants, purple')
    'pants'
    """



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

    Return a string containing all of the non-alphanumeric characters in
    message before the first uppercase character.

    >>> mystery('A!b!3')
    '!!'
    >>> mystery('hello, hi? 123')
    ', ? '
    """

    result = ''
    i = 0
    while not message[i].isdigit():
        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 has_element_of_size(items, size):
    """ (list of str, int) -> bool

    Return True iff items contains a string of exactly length size.

    >>> has_element_of_size(['hi', 'cat', 'hello'], 4)
    False
    >>> has_element_of_size(['hi', 'cat', 'hello'], 3)
    True
    """



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

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):
        result = result + value
        value = value + 1
    return result


value1 = 3
result1 = some_function(value1, 2)
value2 = 4
result2 = some_function(value2, 3)
```

Code	Output
`print(value1)`	3
`print(result1)`	7
`print(value2)`	4
`print(result2)`	15