Question 1.  [3 marks]
Write the body of the following function according to its docstring description.

```python
def contains_even_number(L):
    """ (list of int) -> bool

    Return True iff at least one item in L is even. (Zero is considered even.)
    """

    for item in L:
        if item % 2 == 0:
            return True
    return False
```

Question 2.  [2 marks]
Complete the example function calls by adding arguments that result in the return values shown. (For each example call, there are several correct answers, and providing any one of them will earn full marks.)

```python
def mystery(L):
    """ (list of list of int) -> list of int

    >>> # One inner list with 9 at index 1.
    >>> mystery([[1, 9]])
    [9]
    >>> # Three inner lists with 2, 5, and 8 at index 1 of each list.
    >>> mystery([[1, 2], [4, 5], [7, 8]])
    [2, 5, 8]
    """

    result = []
    for item in L:
        result.append(item[1])
    return result
```
Question 3.  [4 marks]

Read the function header and body and then complete the docstring. Give a meaningful function name, the type contract, the description, and two examples that return different values.

```python
def is_reverse(s1, s2):
    """ (str, str) -> bool

    Precondition: len(s1) == len(s2) and s1.isalpha() and s2.isalpha()

    Return True iff s1 is the same as s2 reversed.
    
    >>> is_reverse('hello', 'olleh')
    True
    >>> is_reverse('abcd', 'acbd')
    False
    """

    i = 0
    flag = True

    for ch in s1:
        ch2 = s2[len(s2) - 1 - i]
        if ch != ch2:
            flag = False
        i = i + 1

    return flag
```
Question 4.  [4 marks]

Write the body of the following function according to its docstring description.

```python
def pass_fail(grades):
    """ (list of number) -> NoneType

    Replace each grade in grades with 'Pass', 'Fail', or 'Invalid'.
    Grades between 0 and 100 (inclusive) are considered valid. Valid grades
    below 50 are failing grades and all other valid grades are passing grades.
    
    >>> grades = [60, 40, 45, -5, 80, 100, 110]
    >>> pass_fail(grades)
    >>> grades
    ['Pass', 'Fail', 'Fail', 'Invalid', 'Pass', 'Pass', 'Invalid']
    ""

    for i in range(len(grades)):
        if grades[i] >= 50 and grades[i] <= 100:
            grades[i] = 'Pass'
        elif grades[i] >= 0 and grades[i] < 50:
            grades[i] = 'Fail'
        else:
            grades[i] = 'Invalid'
```
Question 5. [5 marks]

Write the body of the following function according to its docstring description.

```python
def letter_to_greek(letters, greek_words):
    """ (str, list of str) -> str

    Preconditions: len(letters) >= 1 and len(greek_words) >= 1
    greek_words contains exactly one word that starts with each letter in letters.
    greek_words may also contain words that do not start with a letter in letters.

    Return a new string of Greek words separated by commas in which each word
    starts with the correponding letter in letters.

    >>> letter_to_greek('s', ['beta', 'sigma', 'alpha'])
    'sigma'
    >>> letter_to_greek('bang', ['delta', 'alpha', 'nu', 'beta', 'gamma', 'sigma'])
    'beta,alpha,nu,gamma'
    """

greek = ''
for ch in letters:
    for word in greek_words:
        if word.startswith(ch):  # OR: if word[0] == ch:
            greek += word + ',',  # OR: greek = greek + word + ','
return greek[:-1]
```