Do not turn this page until you have received the signal to start.
(Please fill out the identification section above, write your name on the back of the test, and read the instructions below.)

Good Luck!

This midterm consists of 5 questions on 6 pages (including this one). When you receive the signal to start, please make sure that your copy is complete.

- Comments are not required except where indicated, although they may help us mark your answers.
- No error checking is required: assume all user input and all argument values are valid.
- If you use any space for rough work, indicate clearly what you want marked.
- You may use a pencil; however, work written in pencil will not be considered for remarking.

# 1: _____/ 3  
# 2: _____/ 2  
# 3: _____/ 4  
# 4: _____/ 4  
# 5: _____/ 5  
TOTAL: _____/18

Total Pages = 6
**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.)
    >>> contains_even_number([1, 0, 3, 5, 8, 7, 10])
    True
    >>> contains_even_number([5, 3, 7, -3])
    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
    >>> mystery([9])
    [9]
    >>> mystery([2, 5, 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 (s1, s2):
    ""
    Precondition: len(s1) == len(s2) and s1.isalpha() and s2.isalpha()
    ""
    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']
    """
```
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 corresponding 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'
    """
```
Short Python function/method descriptions:

__builtins__:
in(x) -> int
    Convert x to an integer, if possible. A floating point argument will be truncated towards zero.
len(x) -> int
    Return the length of list, tuple, or string x.
max(a, b, c, ...) -> object
    With two or more arguments, return the largest argument.
min(a, b, c, ...) -> object
    With two or more arguments, return the smallest argument.
print(value, ..., sep=' ', end='\n') --> NoneType
    Prints the values. Optional keyword arguments:
        sep: string inserted between values, default is a space.
        end: string appended after the last value, default is a newline.
range([start], stop, [step]) -> list-like-object of int
    Return the integers starting with start and ending with stop - 1 with step
    specifying the amount to increment (or decrement). If start is not specified,
    the sequence starts at 0. If step is not specified, the values are incremented by 1.
str(x) -> str
    Return an object converted to its string representation, if possible.
str:
x in s --> bool
    Produce True if and only if x is in s.
S.count(sub[, start[, end]]) -> int
    Return the number of non-overlapping occurrences of substring sub in string S[start:end].
    Optional arguments start and end are interpreted as in slice notation.
S.find(sub[,i]) -> int
    Return the lowest index in S (starting at S[i], if i is given) where the
    string sub is found or -1 if sub does not occur in S.
S.isalpha() -> bool
    Return True if and only if all characters in S are alphabetic
    and there is at least one character in S.
S.isdigit() -> bool
    Return True if and only if all characters in S are digits
    and there is at least one character in S.
S.islower() -> bool
    Return True if and only if all cased characters in S are lowercase
    and there is at least one cased character in S.
S.isupper() -> bool
    Return True if and only if all cased characters in S are uppercase
    and there is at least one cased character in S.
S.replace(old, new) -> str
    Return a copy of string S with all occurrences of the string old replaced with the string new.
list:
x in L --> bool
    Produce True if and only if x is in list L
L.append(object) -> NoneType
    Append object to end of list L.
L.extend(iterable) -> NoneType
    Extend list L by appending elements from the iterable. Strings and lists are
    iterables whose elements are characters and list items respectively.