Problem 1.

a) 1 point each

stu
xyz
mno
hij

b) 2 points each

True
e
45

c) 2 points each

b[0] = 44 legal
a[0][1] = 'ee' error
a[1][2] = 'ee' legal

\[c[0][1][1] = 'zz' \text{ error}\]

b[f[2:]] = 51 legal
c[1][2] = 'x' legal

Problem 2.

a) 2 points total
tuples
strings

b) 2 points

False

c) 2 points

\_init\_() creates an instance of a class and initialized all instance variables
or
\_init\_() initializes the attributes of a class when called; creates an object
Problem 3.

```python
def min_max(alist):
    # 6 points total
    odds = []
    for num in alist:
        if num % 2 == 1:
            odds.append(num)
    if len(odds) >= 1:
        return (min(odds), max(odds))
    else:
        return (None, None)
```

-2 if they set min and max to specific numbers

Problem 4.

6 points total
Give 2 points per answer. Expected answers are:

- bad use of variable names (examples: l and t)
- use of while loop rather than a for loop to iterate over a list
- use of booleans to control loop rather than use a for loop
- complicated logic in the booleans
  
  ```python
  n != True  -- This should start at True and be set to False
  ```
- no need for continue in the if statement (would continue anyway)
Problem 5.

# 20 total points

def process_input(fname):
    sfile = open(fname)
    students = {}
    for line in sfile:
        if line[0] == "#":
            continue
        student = line.strip().split(",")
        name = student[0]
        course_grades = []
        for c_pair in student[1:]:
            parts = c_pair.split(":")
            course_grades.append((parts[0], parts[1]))
        students[name] = course_grades
    return students

return students
Problem 6.

a) # 8 total points

def add_pairs(vals):
    assert len(vals) % 2 == 0
    pairs = []
    for i in range(0, len(vals), 2):
        pairs.append(vals[i] + vals[i+1])
    return pairs

b) # 8 total points

def all_odds(arglist):
    for num in arglist:
        if num % 2 == 0:
            return False
    return True

Problem 7.

a) 6 points

The length of s is the longest length of all strings in arglist from arglist[0] to arglist[i]

b) 4 points

The length of s is the longest of any string in arglist

Problem 8.

# 8 total points

def boolean_foo(u, v, w):
    return (u < w and w >= v) or (u > w)

    # return (u < w and w >= v) or (u >= w and u != w) ← this is the long form
Problem 9.  

# 8 points total

class Place:
    def __init__(self, name, latitude, longitude):
        self._name = name
        self._latitude = latitude
        self._longitude = longitude

    def __str__(self):
        return str(self._name) + " : " + str(self._latitude) + \
                "," + str(self._longitude)

    def get_latitude(self):
        return self._latitude

    def get_longitude(self):
        return self._longitude