Work with your neighbor. (This will be graded for participation only.)

1. **Tuples.** Write a function `min_max(L)` that takes a list `L` of integers and returns a tuple of the smallest and largest even numbers in `L`. You may use the built-in min and max functions.

2. Given the dictionary `catalog` shown below:

```
>>> catalog
{ 'MIS': {'mis 101': 4, 'mis 102': 3, 'mis 202': 2},
  'CSC': {'csc 110': 4, 'csc 120': 4, 'csc 352': 3},
  'ECE': {'ece 111': 3, 'ece 222': 3, 'ece 333': 4}}
```

   Use the `items()` method in your solutions for a) and b) below:

   a) Print the keys and values of the `catalog`, separated by a ":"

   b) Print the keys and value of `catalog`, separated by a ":", in sorted order of the keys.

3. Given these assignments:

```
v = "ABCDEFGHIJ"
w = ( ('aa','ab','bc'), (12, 23, 34), [45, 56, 67, 78] )
x = { 'abc' : 12, 'cde' : 34, 'efg' : 56 }
y = [ ['pqr', 'stu','abc','def'], ['uvw','xyz','bcd','cde']  ]
```

(a) For each of the variables `v, w, x, y,` indicate what kind of a data object its value is (i.e., number, list, dictionary, etc.).

(b) Does this give an error?

```
w[2][3] = "hello"
```

(c) Does this give an error?

```
w[1][2] = "sunshine"
```

(d) What are the contents of `x` after the following statement:

```
x[ v[2:4].lower() ] = w[1][2]
```

4. (**We may not get to this problem!)** Given the class definition below:

```
class Student:
     def _ _init_ _(self, name, id):
          self._name = name
          self._id = id
```

```
def get_name(self):
    return self._name
```

a) Write a method `get_id()` that returns a `Student` object's id.

b) Create a `Student` object with name 'Harry' and id 342