

You have been practicing calling/testing functions from the main program. This lab gets you familiarized with calling functions from within a function. The functions being called/used with in another function can be referred to as *helper* functions.

1. Write a python function that accepts a number `numericGrade` which represents a student's grade, and returns this student's academic standing according to the following criteria:

Numerical Grade	Academic standing
80 - 100%	Excellent
70 - 79%	Good
60 - 69%	Adequate
50 - 59%	Marginal
0 - 49%	Inadequate

Function specification	Usage example:
<code>getStanding(number)</code> input: a number (integer) output: academic standing (String).	<code>getStanding(67)</code> returns "Adequate"

2. Consider that you have two lists `studentNames` and `studentGPAs`. They are parallel lists because the values in the same list position correspond to each other, i.e. student with name stored in `studentNames[2]` has a grade point average stored in `studentGPAs[2]`.

```
studentNames = ['John', 'Joe', 'Lisa', 'Jennifer', 'Mike']
studentGPA = [89, 73, 67, 43, 56]
```

Write a function that (1) accepts a student's name, (2) checks the student's grade from the parallel list, then (3) reports the student's academic standing. (*HINT*: call the `getStanding` function defined in step 1)

Function specification	Usage example:
<code>getStandingByName(name)</code> input: a name (String) output: academic standing (String).	<code>getStandingByName(Joe)</code> returns "Good"

3. Write a functions `getMax` and `getMaxStudent` that returns the maximum GPA or the student the student with the max GPA respectively from the parallel lists.

Function specification	Usage example:
<code>getMax(studentGPAs)</code> input: list output: integer.	<code>getMax(studentGPAs)</code> returns 89
<code>getMaxStudent(StudentNames, studentGPAs)</code> input: list output: string.	<code>getMaxStudent(StudentNames, studentGPAs)</code> returns "John"

3. Write a functions `getMin` and `getMinStudent` that returns the maximum GPA or the student the student with the max GPA respectively from the parallel lists.

Function specification	Usage example:
<code>getMin(studentGPAs)</code> input: list output: integer.	<code>getMin(studentGPAs)</code> returns 43
<code>getMinStudent(StudentNames, studentGPAs)</code> input: list output: string.	<code>getMin(StudentNames, studentGPAs)</code> returns "Jennifer"

3. Write a function `getAvg` that returns the average GPA from the parallel lists.

Function specification	Usage example:
<code>getAvg(student_GPA)</code> input: list output: real number.	<code>getAvg(student_GPA)</code> returns 65.6 which is the average

4. Print a report of all students' academic performance in the parallel lists. Specially print the number of students enrolled (the number of students in the list), and the max/min/average GPA. In addition, for each student, print their name and academic standing.

The main function may look something like this: (strings after # signs are comments which are ignored by python. On the lines following each comment, write the actual code that implements the requirements specified in comments.)

```
# initialize lists
student_names = [...]
student_GPA = [...]
# get the number of students in student_names

# get max GPA and student by calling getMax and getMaxStudent, and print it.

# get min GPA and student by calling getMin and getMinStudent, and print it.

# get average GPA by calling getAvg, and print it.
# print the details: for each student, print his/her
# academic standing by calling getStandingByName
```

Sample output:

The total number of enrolled students is 5.
The max GPA is 89 for John.
The min GPA is 43 for Jennifer.
The average GPA is 65.6.

The following is the detailed report:
John's academic standing is: Excellent
Joe's academic standing is: Good
Lisa's academic standing is: Adequate
Jennifer's academic standing is: Inadequate
Mike's academic standing is: Marginal

Thanks to Yiqiao for creating this week's lab 😊