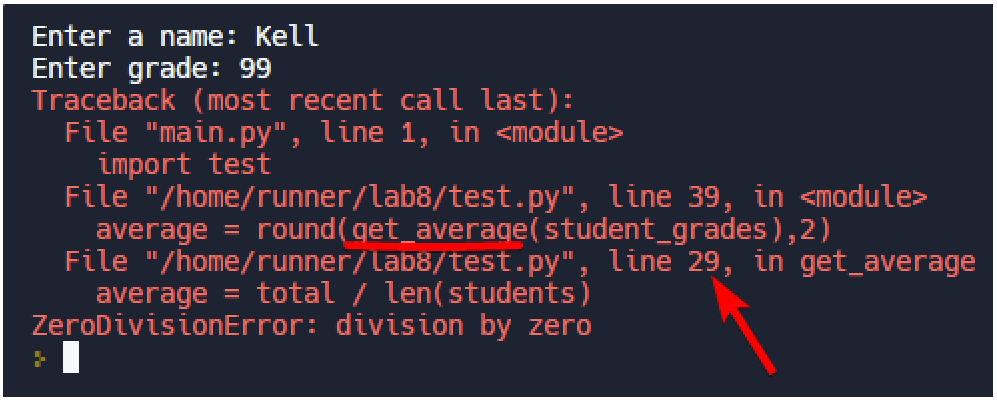
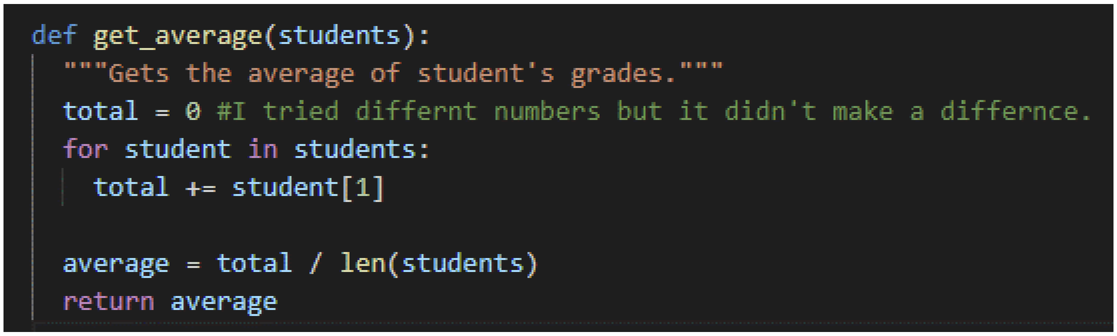
I’m just gonna use this doc to trace my steps so you can get an idea of how I debug stuff.



```
Enter a name: Kell
Enter grade: 99
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    import test
  File "/home/runner/lab8/test.py", line 39, in <module>
    average = round(get_average(student_grades),2)
  File "/home/runner/lab8/test.py", line 29, in get_average
    average = total / len(students)
ZeroDivisionError: division by zero
›
```

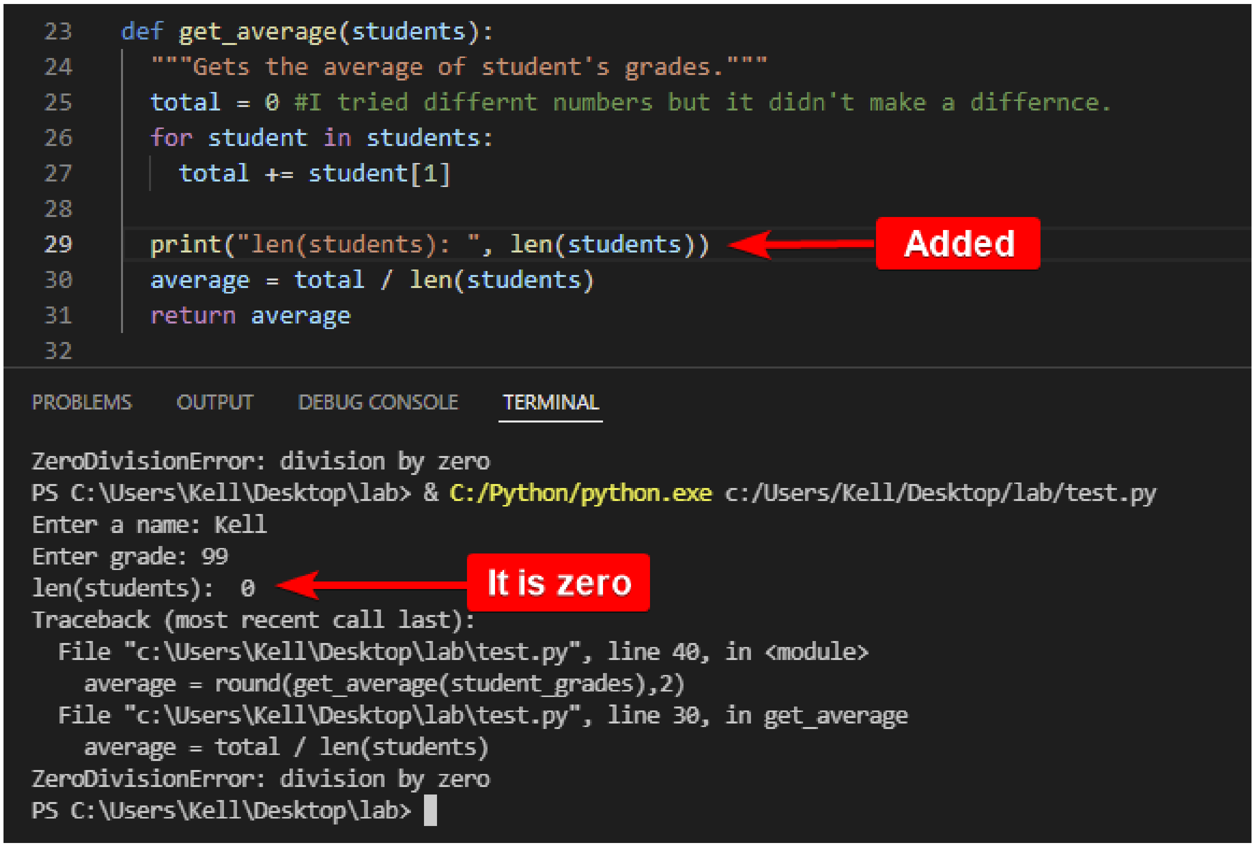
So, from the error message, I know the program crashes at line 29 of test.py. We see 3 errors here because the console is just giving us details as to where the problem is – really, there is only one line of code causing the crash. It also says that we are trying to divide by zero (which is not possible in math or computing), so we know what to look out for.

So, I’m going to go to that line of the code and see if I notice anything weird about it.



```
def get_average(students):
  """Gets the average of student's grades."""
  total = 0 #I tried differnt numbers but it didn't make a differnce.
  for student in students:
    total += student[1]

  average = total / len(students)
  return average
```

Since the error is saying that we are trying to divide by 0, I’m going to guess that the inputted array of “students” is actually empty. So **len(students)** must be returning 0. I’ll test it by printing out the value for **len(students)** just before line 29.



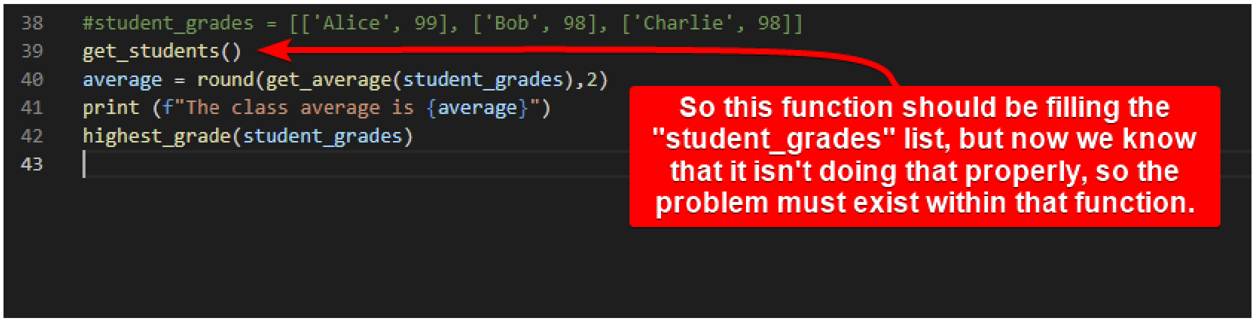
```
23  def get_average(students):
24      """Gets the average of student's grades."""
25      total = 0 #I tried differnt numbers but it didn't make a differnce.
26      for student in students:
27          total += student[1]
28
29      print("len(students): ", len(students))
30      average = total / len(students)
31      return average
32
```

Added

```
ZeroDivisionError: division by zero
PS C:\Users\Kell\Desktop\lab> & C:/Python/python.exe c:/Users/Kell/Desktop/lab/test.py
Enter a name: Kell
Enter grade: 99
len(students):  0
Traceback (most recent call last):
  File "c:\Users\Kell\Desktop\lab\test.py", line 40, in <module>
    average = round(get_average(student_grades),2)
  File "c:\Users\Kell\Desktop\lab\test.py", line 30, in get_average
    average = total / len(students)
ZeroDivisionError: division by zero
PS C:\Users\Kell\Desktop\lab>
```

It is zero

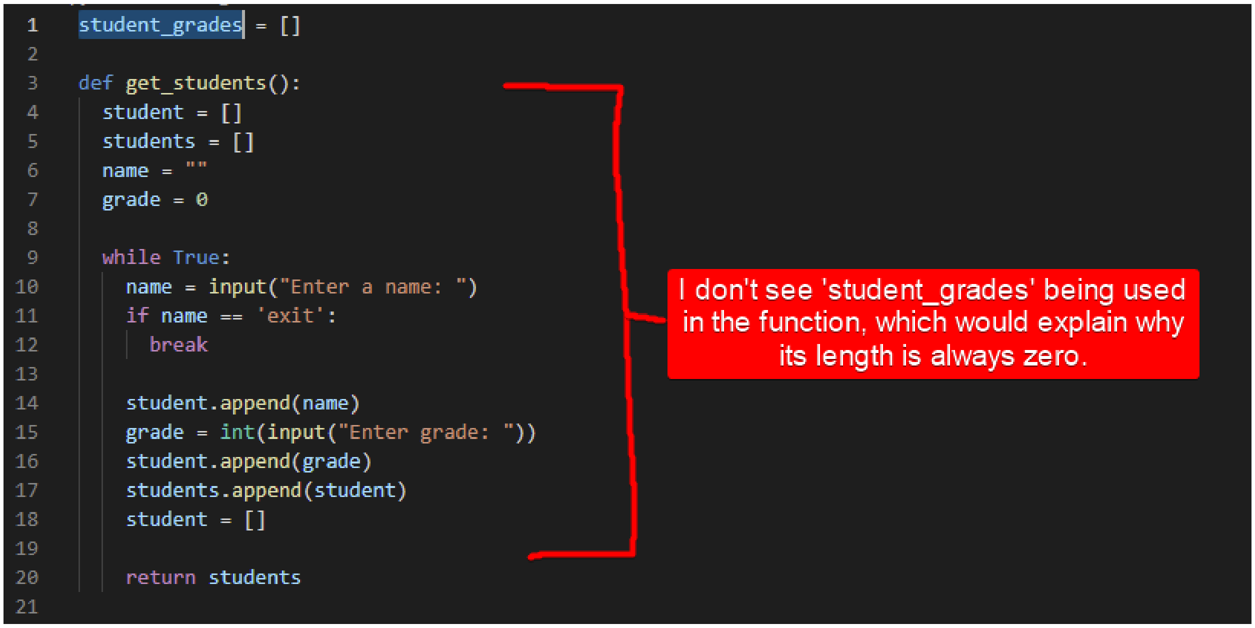
So, the list of students is never filled. So now I'll go look at the code that should be filling this list.



```
38  #student_grades = [['Alice', 99], ['Bob', 98], ['Charlie', 98]]
39  get_students()
40  average = round(get_average(student_grades),2)
41  print (f"The class average is {average}")
42  highest_grade(student_grades)
43
```

So this function should be filling the "student_grades" list, but now we know that it isn't doing that properly, so the problem must exist within that function.

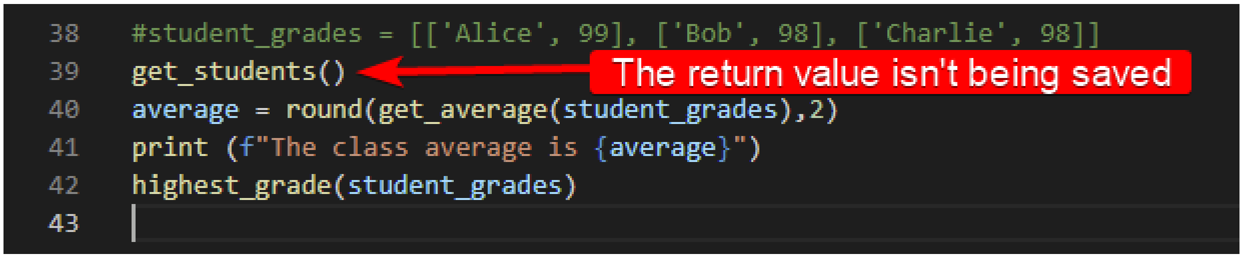
So now I'll try to see why **get_students()** isn't populating the **student_grades** global variable.



```python
student_grades = []

def get_students():
    student = []
    students = []
    name = ""
    grade = 0

    while True:
        name = input("Enter a name: ")
        if name == 'exit':
            break

        student.append(name)
        grade = int(input("Enter grade: "))
        student.append(grade)
        students.append(student)
        student = []

        return students

```

I don't see 'student_grades' being used in the function, which would explain why its length is always zero.

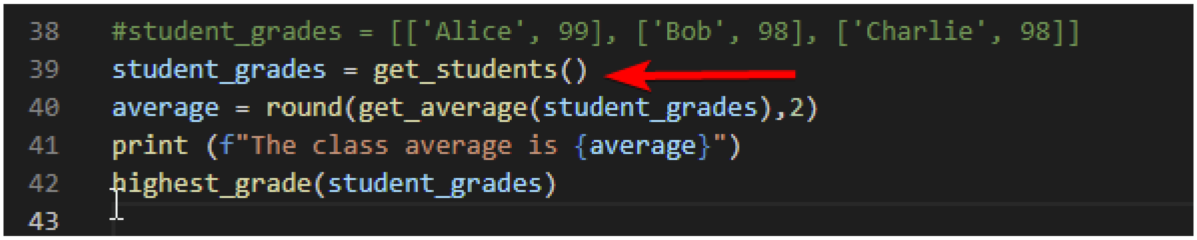
So, **student_grades** isn't being used anywhere in this function. That's okay though because this function is returning a list of students. My question now though is whether that list is being captured in a variable. Just calling the function isn't enough, you need to assign its return value to a variable in order to save the results.



```python
#student_grades = [['Alice', 99], ['Bob', 98], ['Charlie', 98]]
get_students()
average = round(get_average(student_grades),2)
print (f"The class average is {average}")
highest_grade(student_grades)

```

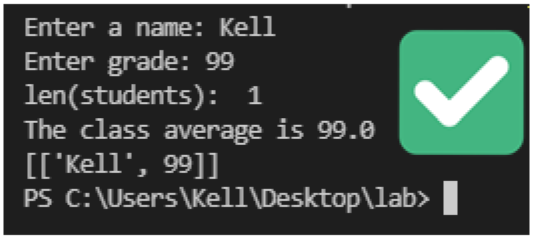
The return value isn't being saved

The fix should be this:



```python
#student_grades = [['Alice', 99], ['Bob', 98], ['Charlie', 98]]
student_grades = get_students()
average = round(get_average(student_grades),2)
print (f"The class average is {average}")
highest_grade(student_grades)

```

Anddddd voila



```
Enter a name: Kell
Enter grade: 99
len(students):  1
The class average is 99.0
[['Kell', 99]]
PS C:\Users\Kell\Desktop\lab>
```