```
#!/usr/bin/python

import sys
import os
import numpy

iterate = 51
count = 0
stat = numpy.empty(iterate,dtype=float)

# Function to open the files that contain the matrices from the c code
# Checks to see if the multiplication results are the same from c and python
# Saves the time from the matrix multiplication in a numpy array
def openmatrices():
    global stat
    global count
    try:
        f1 = open(’/home/pi/ECE331/mmult/m1.txt’,’r’)
        f2 = open(’/home/pi/ECE331/mmult/m2.txt’,’r’)
        f3 = open(’/home/pi/ECE331/mmult/m3.txt’,’r’)
        f4 = open(’/home/pi/ECE331/mmult/time.txt’,’r’)
    except:
        print "\nCould not open the file\n"
        sys.exit()

    mat1 = numpy.loadtxt(f1,dtype=float)
    mat2 = numpy.loadtxt(f2,dtype=float)
    mat3 = numpy.loadtxt(f3,dtype=float)
    timed = numpy.loadtxt(f4,dtype=float)

    res = numpy.dot(mat1,mat2)

    if numpy.array_equal(res,mat3):
        count = count + 1

    stat[i] = timed

    f1.close()
    f2.close()
    f3.close()
    f4.close()

# Call the mmult executable and get the results
for i in range(iterate):
    os.system(’./mmult 1000 1000 1000 1000’)
    openmatrices()

# Give the summary of the matrix multiplication for 1000x1000 matrices
print "********** SUMMARY **********\n"
print "Matrix multiply with 1000x1000 matrices was successful {} out of {} times.".format(c
ount,iterate)
print "The average time multiplying 1000x1000 matrices together for {} runs was {} seconds
on a Raspberry Pi 4 4GB.".format(iterate,numpy.average(stat))
print "The minimum time was {} seconds and the maximum time was {} seconds and a standard d
eviation of {} seconds.\n\n".format(numpy.amin(stat),numpy.amax(stat),numpy.std(stat))

# Check for some errors
print "********** ERROR CHECKING **********\n"
print "Testing with not enough arguments: "
os.system(’./mmult 1 4 4’)
print "Testing with too many arguments: "
os.system(’./mmult 1 4 4 5 6’)
print "Testing with non number arguments: "
os.system(’./mmult wfe 4 4 5’)
print "Testing different variation of non number arguments: "
os.system(’./mmult 1 4 4 5ef’)
print "Testing with matrices that can’t multiply: "
os.system(’./mmult 1 3 4 5’)
```

```
print "Testing with dimension of 0: "
os.system('./mmult 0 4 4 5')
print "Testing with negative dimension: "
os.system('./mmult -1 4 4 5')
print "Error Checking completed"

# Check some test cases
count = 0
print "********** TEST CASES **********\n"
print "\nWorking Matrices Test case 1: "
os.system('./mmult 1 2 2 4')
openmatrices()
print "\nWorking Matrices Test case 2: "
os.system('./mmult 3 5 5 2')
openmatrices()
print "\nWorking Matrices Test case 3: "
os.system('./mmult 6 6 6 6')
openmatrices()
print "Testing Completed"
print "{} out of the 3 test cases passed.\n".format(count)
```