#--------------------------------------------------------

# Developer---Abel Gonzalez

# Course------CS1213-01

# Project-----Project #9

# Due---------December 5,2018

#

# This program computes semester averages and letter

# grades for this course. The program will input grade

# data from a file. The file will contain data for an

# entire class of students.

#--------------------------------------------------------

#--------------------------------------------------------

# gradeletter(numgrade); Takes a numerical value as an

# argument, numgrade, and assigns a letter grade for the

# value dependent on where it lies on the spectrum of

# values.

#--------------------------------------------------------

def gradeletter(numgrade):

if numgrade >= 90:

letter = 'A'

elif numgrade >= 80:

letter = 'B'

elif numgrade >= 70:

letter = 'C'

elif numgrade >= 60:

letter = 'D'

else:

letter = 'F'

return letter

#--------------------------------------------------------

# Bool(quiz\_line); Takes a line from the lines in the

# text document and returns either Boolean value of true

# or false based on how many objects are in the list.

#--------------------------------------------------------

def Bool(quiz\_line):

if len(quiz\_line) >= 2:

return True

else:

return False

#--------------------------------------------------------

# quiz\_avg(quiz\_line); Computes the average for the

# quiz lines in the text document

#--------------------------------------------------------

def quiz\_avg(quiz\_line):

total = 0

grades = quiz\_line.split()

length = len(grades)

drop = 100

for q in grades:

partial = int(q.replace('.',''))

total += partial

if partial < drop:

drop = partial

if Bool(grades):

total = total - partial

length -= 1

return (total/length)

#--------------------------------------------------------

# project\_avg(project\_line); Computes the average for

# the project lines in the text document.

#--------------------------------------------------------

def project\_avg(project\_line):

total = 0

grades = project\_line.split()

length = len(grades)

for q in grades:

total += float(q)

return (total/length)\*10

#--------------------------------------------------------

# test\_avg(test\_line); Computes the average for the test

# line using the weights of each item to be used in

# semester\_avg function.

#--------------------------------------------------------

def test\_avg(test\_line):

grades = test\_line.split()

first = float(grades[0])

second = float(grades[1])

final = float(grades[2])

length = len(grades)

total = (first\*0.2)+(second\*0.2)+(final\*0.25)

return total/0.65

#--------------------------------------------------------

# semester\_avg(quizes,projects,tests); Computes the

# semester average given the averages for the quizzes,

# projects, and tests. Uses the weights of the items in

# the calculations.

#--------------------------------------------------------

def semester\_avg(quizes,projects,tests):

quizweight = quizes \* 0.15

projectweight = projects \* 0.2

testweight = tests \* 0.65

return quizweight + projectweight + testweight

#--------------------------------------------------------

# main(); opens two seperate files(one for reading and

# one for writing) and computes and prints the semester

# average and letter grade for students in a class onto

# the second file.

#--------------------------------------------------------

def main():

filename = input('Enter input file name:')

outfile = input('Enter output file name:')

inputFile = open(filename,'r')

outputFile = open(outfile,'w')

lines = inputFile.readlines()

index = 0

while index <= len(lines)-1:

names = lines[index]

quiz\_scores = lines[index+1]

project\_scores = lines[index+2]

test\_scores = lines[index+3]

quiz = quiz\_avg(quiz\_scores)

project = project\_avg(project\_scores)

test = test\_avg(test\_scores)

total\_avg = semester\_avg(quiz,project,test)

alphagrade = gradeletter(total\_avg)

outputFile.write('%-20s %6.1f %s \n'%(names.strip('\n'),total\_avg,alphagrade),)

index += 4

inputFile.close()

outputFile.close()

return

#--------------------------------------------------------

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MAIN PROGRAM\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#--------------------------------------------------------

main()