#find the average of a list of numbers in a file

def main():

filename=input("Enter a file name please: ")

inFP= open(filename,"r")

count=0

summ=0

for line in inFP:

data= float(line)

summ+=data

count+=1

if count==0:

print("ERROR!!")

else:

print("The average is:", summ/count)

inFP.close()

main()

#make a multiplication table

#fuck this

def main():

print(" 1 2 3 4 5 6 7 8 9 10")

for i in range(1,11):

for j in range(1,i+1):

print(i, end="")

print("{0:>4}".format(i\*j), end=" ")

print()

main()

# Sean MacBride

# Program: smacbrideP4.py

# Description: A program to convert data in a file from base ten to base three, or base 3, based on earlier lines in the program

# Input: The name of the data file

# Output: All of the converted values, with a message for when the program is finished

#Given a number in base 10, prints the equivalent in base 3

def ter(orig):

#orig is the original input from the data file

#lenorig is the length of the original data file's integer values only

#origval is the integer value of the original data point

#newval is the list with the ternary values

#finalval is the final integer value of the ternary value, converted from the newval list

#prints out a statement of the final value in converted form

origval=eval(orig[0:len(orig)])

newval=[]

while origval//3!=0:

newval.append(origval%3)

origval=origval//3

newval.append(origval%3)

newval=newval[::-1] #Flipping the order of the list so it can be displayed correctly

finalval=0 #Set finalval as 0 for algorithim in for loop in next line

for i in range(0,len(newval)):

finalval+=newval[i]\*10\*\*(len(newval)-i-1)

orig=orig[0:len(orig)-1] #removes the line break that was in orig

print("The number ",orig," in base 10 is the same as ",finalval, " in base 3.",sep="")

print()

#Function that takes a ternary number and returns a base 10 number

def int(orig):

#orig is the original string value

#origlist is the list containing the original values

#newtot is the actual integer value in base 10

#prints out a statement of the final value in converted form

origlist=orig[0:len(orig)-1]

origlist=origlist[::-1]

newtot=0 # Set to 0 for use in the for loop in the next line

for i in range(0,len(orig)-1):

newtot+=eval(origlist[i])\*3\*\*i

orig=orig[0:len(orig)-1] #removes the line break that was in orig

print("The number ",orig, " in base 3 is the same as ",newtot, " in base 10.",sep="")

print()

#The findfile function directs the correct data to either int() or ter()

def findfile(filename):

#infile opens the file in read form

#commandline reads the line that directs

#valline is the value of the line before conversion

#this function directs the line data to whichever converter function is necessary

print()

infile=open(filename, "r")

commandline=infile.readline()

while eval(commandline)!=3:

if eval(commandline)==1:

valline=infile.readline()

int(valline)

elif eval(commandline)==2:

valline=infile.readline()

ter(valline)

commandline=infile.readline()

infile.close()

print("Finished!")

def main():

filename = input("Enter filename: ") # asking for file name

findfile(filename) # calls findfile function

#I have abided by the Wheaton honor code in this work

main()

#Gives the square root of a number rounded to 2 dec places

import math

def main():

num=eval(input("Enter your number here: "))

sqroot=math.sqrt(num)

sqroot= int((sqroot + 0.005) \* 100) / 100

print("The square root of",num,"is",sqroot)

main()

#Gives the average of 3 scores in a file

def main():

infile=open("scoresfeb19.dat","r")

print("Reading from file...")

score1=int(infile.readline())

score2=int(infile.readline())

score3=int(infile.readline())

print(score1,score2,score3)

average=(score1+score2+score3)/3.0

print("The average is",average)

infile.close()

main()

#Function that rolls a dice twice in main()

import random

def die():

a=random.randint(1,6)

return a

def main():

a=input("Press enter to roll the die!")

print(die())

print(die())

main()

#Returns the fibonnaci sequence when prompted for a terms

def fib(a):

folder=0

fold=1

flist=[]

flist.append(fold)

for i in range(0,a-1):

f=fold+folder

flist.append(f)

folder=fold

fold=f

return flist

#Returns the sum of a fibonnaci sequence with a terms

def fibsum(a):

folder=0

fold=1

flist=[]

flist.append(fold)

for i in range(0,a-1):

f=fold+folder

flist.append(f)

folder=fold

fold=f

numsum=sum(flist)

return (numsum)

def main():

number=int(input("Enter in however many fibonacci numbers you want: "))

print(fib(number))

print("The sum of those numbers is",fibsum(number))

main()

#Manually making the power function

def power(x,y):

b=1

if x<0:

return print("Done")

else:

if y>0:

for i in range(0,y):

b=b\*x

elif y<0:

for i in range(0,-y):

b=b/x

else:

b=1

return b

def getvalues():

a=float(input("Enter your first value here "))

if a<0:

return ("Done","")

else:

b=int(input("Enter your second value here "))

return(a,b)

def main():

a,b=getvalues()

if type (a) is str:

print("Done")

else:

print(power(a,b))

main()

def sortem(a,b,c):

if a>=b>=c:

return (a,b,c)

elif a>=c>=b:

return (a,c,b)

elif b>=a>=c:

return (b,a,c)

elif b>=c>=a:

return(b,c,a)

elif c>=a>=b:

return (c,a,b)

elif c>=b>=a:

return (c,b,a)

def main():

x=eval(input("Enter your first number here "))

y=eval(input("Enter your second number here "))

z=eval(input("Enter your third number here "))

print(sortem(x,y,z))

main()

#Program that returns the Lastname, Firstinitial

firstname=str(input("What is your first name? "))

lastname=str(input("What is your last name? "))

fullname=str(input("What is your full name? "))

lastnamelen=len(lastname)

def main():

print(lastname+",","{0:1.1}".format(firstname)+".")

main()

#The auxiliary function that will return the larger number to main()

def maxx(a,b):

if a>b:

return a

elif b>a:

return b

elif b==a:

return a

#The main() function, which requests two values as an input, and with the maxx() function prints the larger value

def main():

one, two = eval(input("Enter two values: "))

print(maxx(one,two), "is the larger value")

main()

#The auxiliary function which returns the completed string with the longer answer prompt

def maxx(a,b,c):

if a>b:

c[0]=a

if b>a:

c[0]=b

#The

def main():

answer=[0]

one,two=eval(input("Enter two values: "))

maxx(one,two,answer)

print(maxx(answer[0], "is the larger of the two"))

main()

#A function that swaps the two values

def swap(a,b):

return b,a

def main():

one, two=eval(input("Two values! "))

one, two=swap(one,two)

print(one,two)

main()

#A program that models a tv remote

def getbutton():

newbutton=input()

while newbutton!="u" and newbutton!="d" and newbutton!="o":

newbutton=input()

return newbutton

def nextch(oldchannel,button):

if button=="u":

if oldchannel==13:

return 2

else:

return oldchannel+1

else:

if oldchannel==2:

return 13

else:

return oldchannel-1

def main():

channel=2

print(channel)

button=getbutton()

while button!="o":

channel=nextch(channel,button)

print (channel)

button=getbutton()

print("Goodbye!")

main()

#Program that prompts for a single character string value of a number, returns the int val +5

def num1(val):

val=ord(val)-48

return val

def num2(val):

if val=="0":

return 0

elif val=="1":

return 1

elif val=="2":

return 2

elif val=="3":

return 3

elif val=="4":

return 4

elif val=="5":

return 5

elif val=="6":

return 6

elif val=="7":

return 7

elif val=="8":

return 8

elif val=="9":

return 9

def main():

singval=input("Enter a single character: ")

numnew1=num1(singval)+5

numnew2= num2(singval)+5

print("First function +5 yields",numnew1)

print("Second function +5 yields", numnew2)

main()