**Every Boiler Engineering Code – Entry Level Programming**

**Week 3 – Programming Exercises**

1. **(15 points Sum and average of numbers)** Write a Python program with a loop that asks the user to enter a series of positive numbers. The user should enter a negative number to signal the end of the series. After all the positive numbers have been entered, the program should display their sum and average (**format the precision of average as 2.**)

**(use 9, 13, 5, 8, 19, 78 to test your program) 9 13 5 8 19 78**

**CODE:**

## PROBLEM #1

# NAME: Tomoki Koike

# DUE: 3/19/2019

# DESCRIPTION: This program will accept a positive value input from the user

# unitl the user terminates with a negative input. The program will compute the

# sum and the average of the user inputs

# STAND: Class of 2020

##

# Initialize sum and counter

sum1 = 0

i = 0

# Accpeting the user input (with input validation)

while True:

try:

enter\_num = float(input('Please enter a positive integer to continue, or enter a negative integer to terminate -> '))

except ValueError:

print('Sorry, cannot understand. Please try again.')

continue

if enter\_num >= 0:

# Computing the sum

sum1 += enter\_num

i += 1

continue

else:

break

# Computing the Average

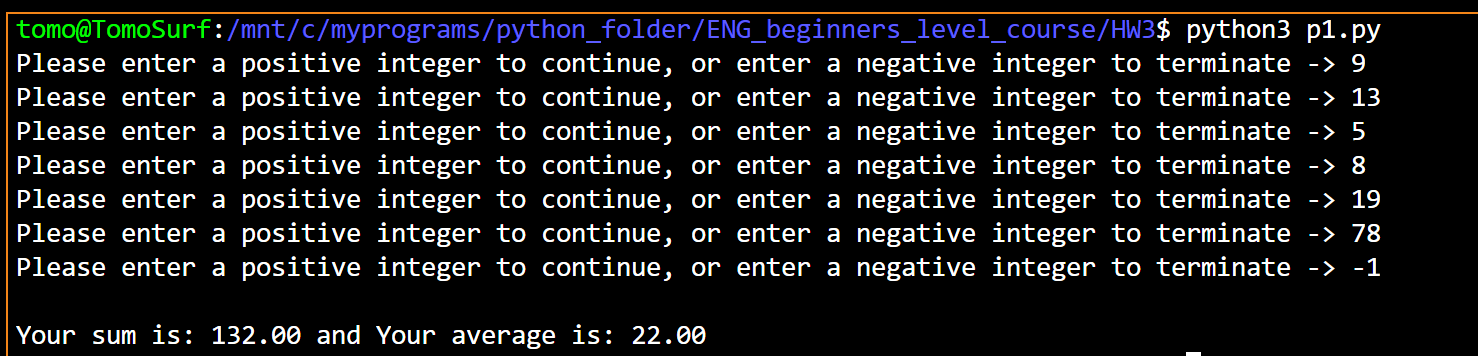
avg = sum1 / i

# Final output

print()

print('Your sum is: {0:.2f} and Your average is: {1:.2f}'.format(sum1, avg))

**SAMPLE EXECUTION:**



1. **(15 points)** Write a program that uses nested loops to collect data and calculate the average rainfall over a period of years. The program should first ask for the number of years. The outer loop will iterate once for each year. The inner loop will iterate twelve times, once for each month. Each iteration of the inner loop will ask the user for the inches of rainfall for that month. After all iterations, the program should display the number of months, the total inches of rainfall, and the average rainfall per month for the entire period.

**(use number of years as 2, use any number as the inches of rainfall for each month to test your code.)**

**CODE:**

## PROBLEM #2

# NAME: Tomoki Koike

# DUE: 3/19/2019

# DESCRIPTION: This program will accept a user input of number of years to enter data for

# percipitation for the month. And the program will then calculate the total number of

# months, total inches of percipitation, and the average percipitation per month.

# STAND: Class of 2020

##

# import module

import numpy as np

# list of month names

months = [

'January',

'February',

'March',

'April',

'May',

'June',

'July',

'August',

'September',

'October',

'November',

'December'

]

# Function for accpeting the user input (with input validation)

def input\_valid(prompt, input\_type):

while True:

try:

if input\_type == 'int':

enter = int(input(prompt))

else:

enter = float(input(prompt))

except ValueError:

print('Sorry, could not understand. Please try again.')

continue

if enter < 0:

print('Please enter a proper value.')

continue

else:

break

return enter

# Preallocate the vector to hold the input percipitation values

data = []

# Accept user input of number of years

num\_year = input\_valid('Please enter the number of years you will enter data for -> ', 'int')

# The nested loop

for i in range(num\_year):

print()

print('Year', i+1, ':')

print('-----------------------------------')

for n in range(12):

rainfall = input\_valid('Enter the perciption (in) for {} -> '.format(months[n]), 'float')

data.append(rainfall)

# Compute sum of percipitation, and average percipitation per month

# Total number of months

tot\_months = num\_year \* 12

# Sum

sum\_ppt = np.sum(data)

# Average

avg\_ppt = sum\_ppt / tot\_months

# Final outputs

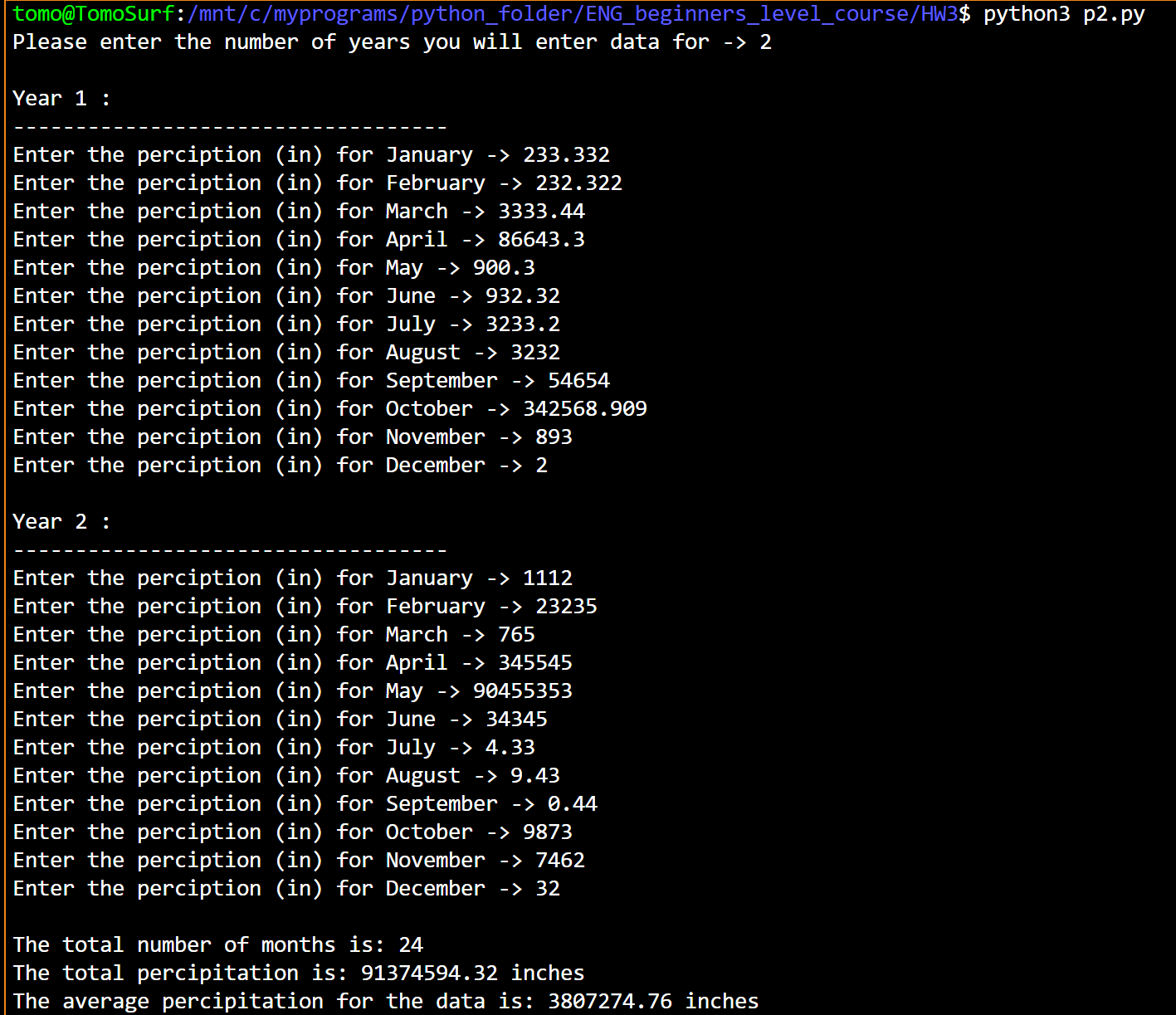
print()

print('The total number of months is:', tot\_months)

print('The total percipitation is: {0:.2f}'.format(sum\_ppt), 'inches')

print('The average percipitation for the data is: {0:.2f}'.format(avg\_ppt), 'inches')

**SAMPLE EXECUTION:**



1. **(15 points)** Write a program that predicts the approximate size of a population of organisms. The application should allow the user to enter the starting number of organisms, the average daily population increase (as a percentage), and the number of days the organisms will be left to multiply. For example, assume the user enters the following value:

Starting number of organisms 2

Average daily increase: 30%

Number of days to multiply: 10

The program should display the following table of data:

Day Approximate Population

1 2

2 2.6

3 3.38

4 4.394

5 5.7122

6 7.42586

7 9.653619

8 12.5497

9 16.31462

10 21.209

**Use the following data to test your code:**

**Starting number of Organisms: 4**

**Average daily increase: 20%**

**Number of days to multiply: 10**

**CODE:**

# NAME: Tomoki Koike

# DUE: 3/19/2019

# DESCRIPTION: This program is designed to compute the population increase

# of an organism based on the user inputs of initial number of organisms,

# percent increase per day, and the number of days to multiply.

# STAND: Class of 2020

##

# Function for accpeting the user input (with input validation)

def input\_valid(prompt, input\_type):

while True:

try:

if input\_type == 'int':

enter = int(input(prompt))

else:

enter = float(input(prompt))

except ValueError:

print('Sorry, could not understand. Please try again.')

continue

if enter < 0:

print('Please enter a proper value.')

continue

else:

break

return enter

# Accepting user inputs

# Initial number of organisms (current population)

curr\_pop = input\_valid('Enter the initial population of the organism -> ', 'float')

# The percent increase per day of the organism

avg\_increase = input\_valid('Enter the average daily population increase (%) -> ', 'float')

# The number of days to multiply

num\_days = input\_valid('Enter the number of days the organism multiplies -> ', 'int')

print()

print('{0:25}{1}'.format('Day Approximate','Population') )

# The loop to compute the population increase

for x in range(num\_days):

# Caldulating the increased population

if x == 0:

pass

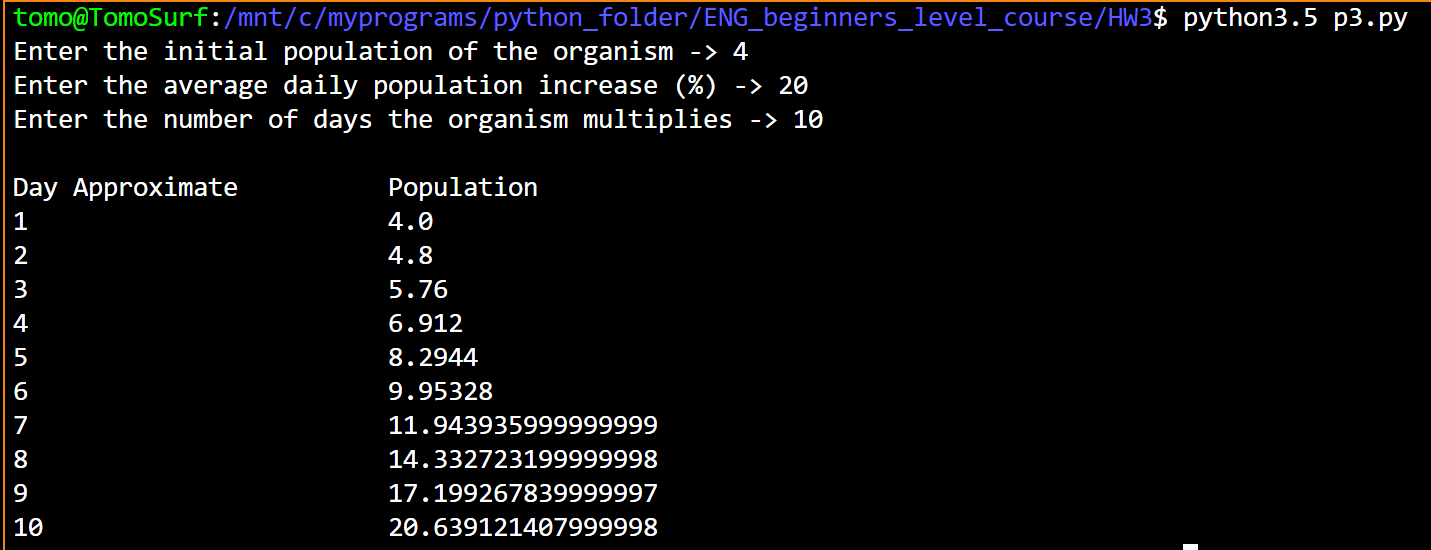
else:

curr\_pop \*= (1 + avg\_increase / 100)

# Printing Results

print('{0:25}'.format(str(x+1)), curr\_pop)

**SAMPLE EXECUTION:**



1. **(Bonus question 10 points)** Write a Python program that uses nested loops to draw this pattern:

**\*\*\*\*\*\*\***

**\*\*\*\*\*\***

**\*\*\*\*\***

**\*\*\*\***

**\*\*\***

**\*\***

**\***

**CODE:**

## PROBLEM #4

# NAME: Tomoki Koike

# DUE: 3/19/2019

# DESCRIPTION: This program will print out a up-side-down

# triangle using a loop

# STAND: Class of 2020

##

base = 7

for r in range(base,0,-1):

for c in range(r):

print('\*', end="")

print()

**EXECUTION:**

