

**ANL252**

**Python for Data Analytics**

# **Tutor-Marked Assignment**

**July 2021 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Tong Yew Tiong Alex** | **H0914367** |

**Tutorial Group: ­­­­­­­­­­­­ T 09**

**Instructor’s Name: Dr. Munish Kumar**

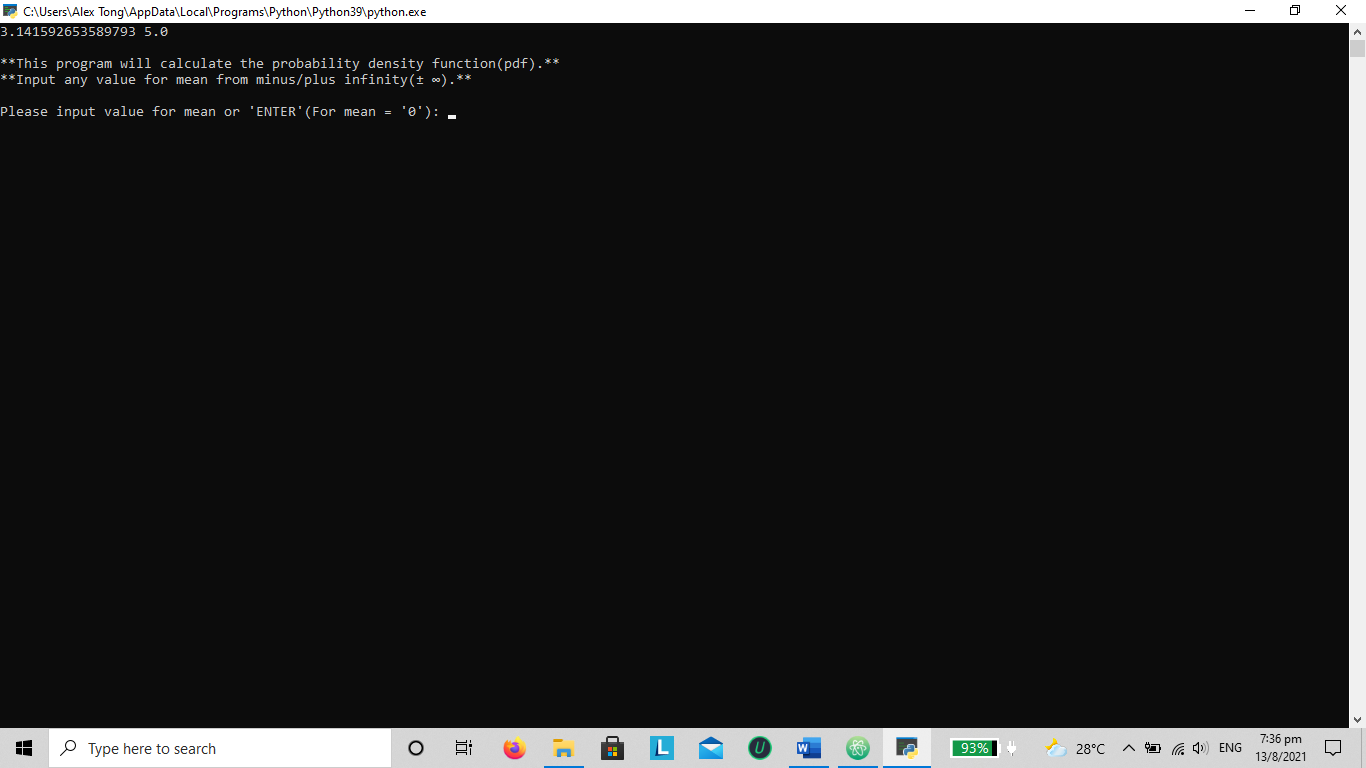
**Submission Date: 15/08/2021**

a)

# Question (a) Prepare the programming by importing the math package.

import math

print(math.pi, math.sqrt(25))



b)

# Question (b) To employ a program that ask user to enter the mean and variance of the distribution. Mean is any value ± ∞, and variance > 0.

# ‘Enter’ for mean = 0 and variance =1

valid\_input1 = False

while valid\_input1 == False:

input\_mean = input(f"\nPlease input value for mean or 'ENTER'(For mean = '0'.): ")

if input\_mean == "":

input\_mean = 0

try: # control mechanism for mean using try block

input\_mean = float(input\_mean)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input1 = True

print(f"\*\*The input value for mean is: {input\_mean}\*\*")

# 'Enter' for variance = 1 and control mechanism for variance.

print("\n\*\*The input value for variance need to be greater than '0'.\*\*")

input\_variance = input(f"Please input value for variance or 'ENTER'(For variance = '1'): ")

if input\_variance == "":

input\_variance = 1

input\_variance = float(input\_variance)

print(f"\*\*The input value for variance is: {input\_variance}.\*\*")

else:

valid\_input2 = False

while valid\_input2 == False:

try: # control for variance > 0 and variable as float

print("\n\*\*The input value for variance need to be greater than '0'.\*\*")

input\_variance = float(input(f"Please input value for variance.: "))

if input\_variance > 0:

print(f"\*\*The input value for variance is: {input\_variance}.\*\*")

break

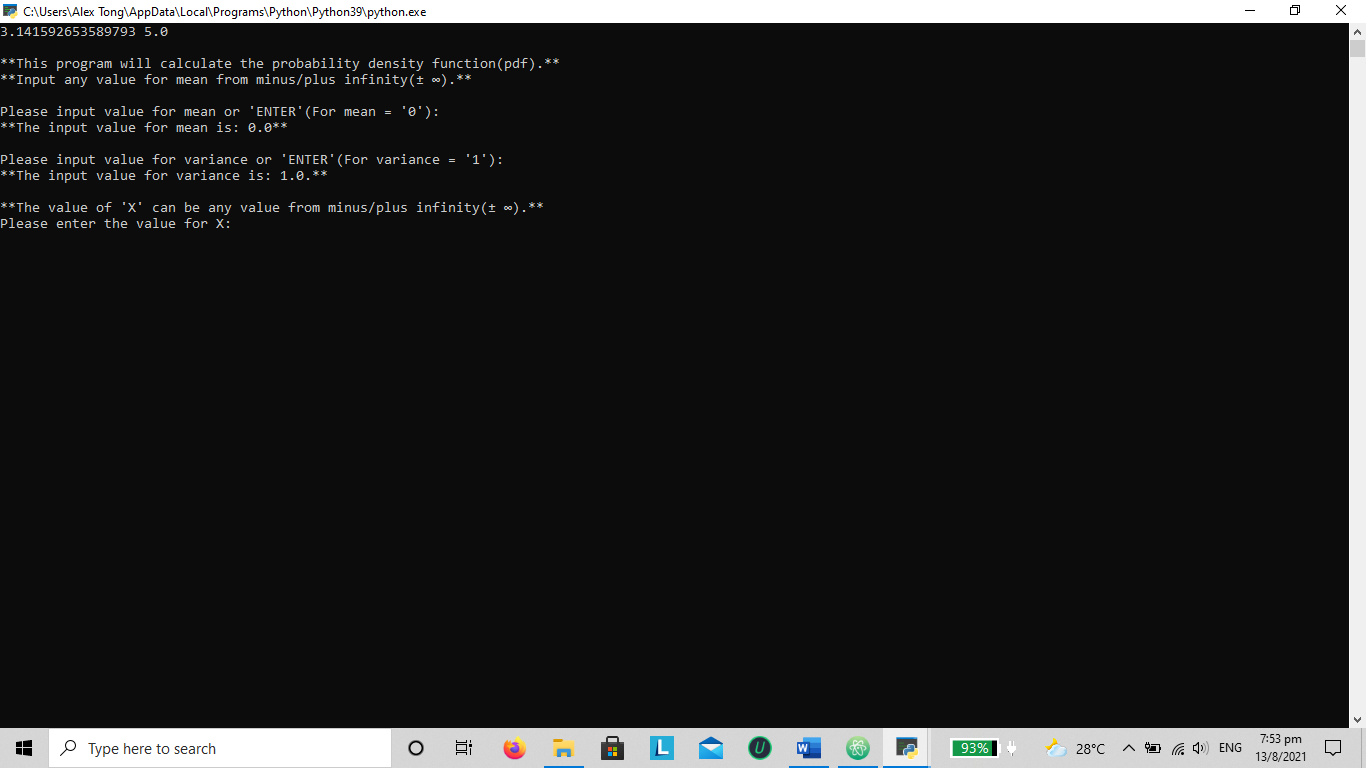
except ValueError:

print("Your input is not numeric. Please try again. ")

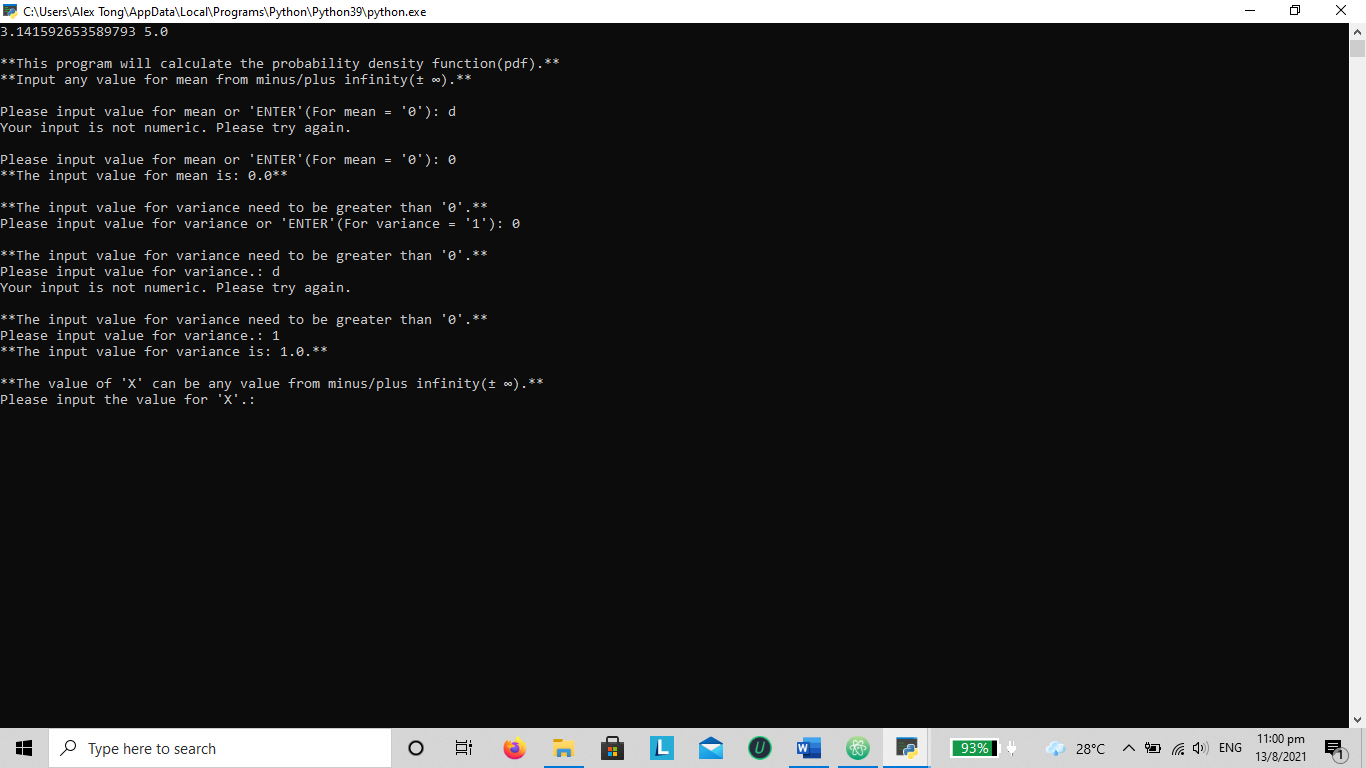
else:

valid\_input1 = True

When ‘ENTER’ is pressed



Output with test control when values d, 0 and 1 entered



c)

# Question (c) Design an input screen for user to enter the value 'X'.

# Input to be any value ± ∞ and employ control mechanism for the value.

valid\_input3 = False

while valid\_input3 == False:

print("\n\*\*The value of 'X' can be any value from minus/plus infinity(± ∞).\*\*")

input\_X = input(f"Please input the value for 'X'.: ")

try: # control for 'X' as a float variable

input\_X = float(input\_X)

except ValueError:

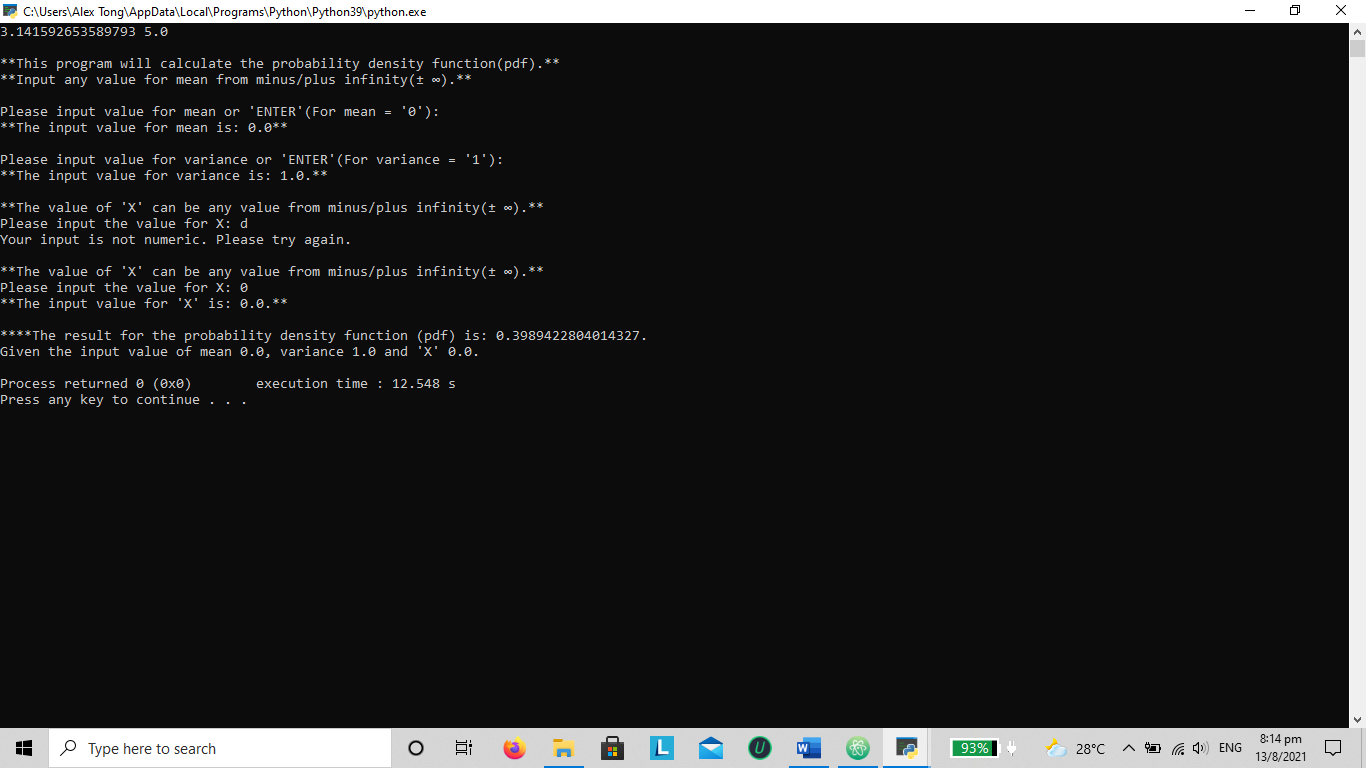
print("Your input is not numeric. Please try again.")

else:

valid\_input3 = True

print(f"\*\*The input value for 'X' is: {input\_X}.\*\*")

The output for ‘X’, with test on the control mechanism.



d)

# Question (d) To construct a user-defined function using the formula of the probability density function, based on part (b) and (c).

def power(a, b):

return a \* math.exp(b)

# Listing of the variables:

mean = (input\_mean)

variance = (input\_variance)

x = (input\_X)

div\_top = x - mean

div\_top\_sq = pow(div\_top, 2)

div\_bottom = 2 \* variance

part\_b = 1 / math.sqrt(2 \* math.pi \* variance)

part\_b\_c = -(div\_top\_sq / div\_bottom)

# Using the user defined function for fx(x)

fx\_x = power (part\_b, part\_b\_c)

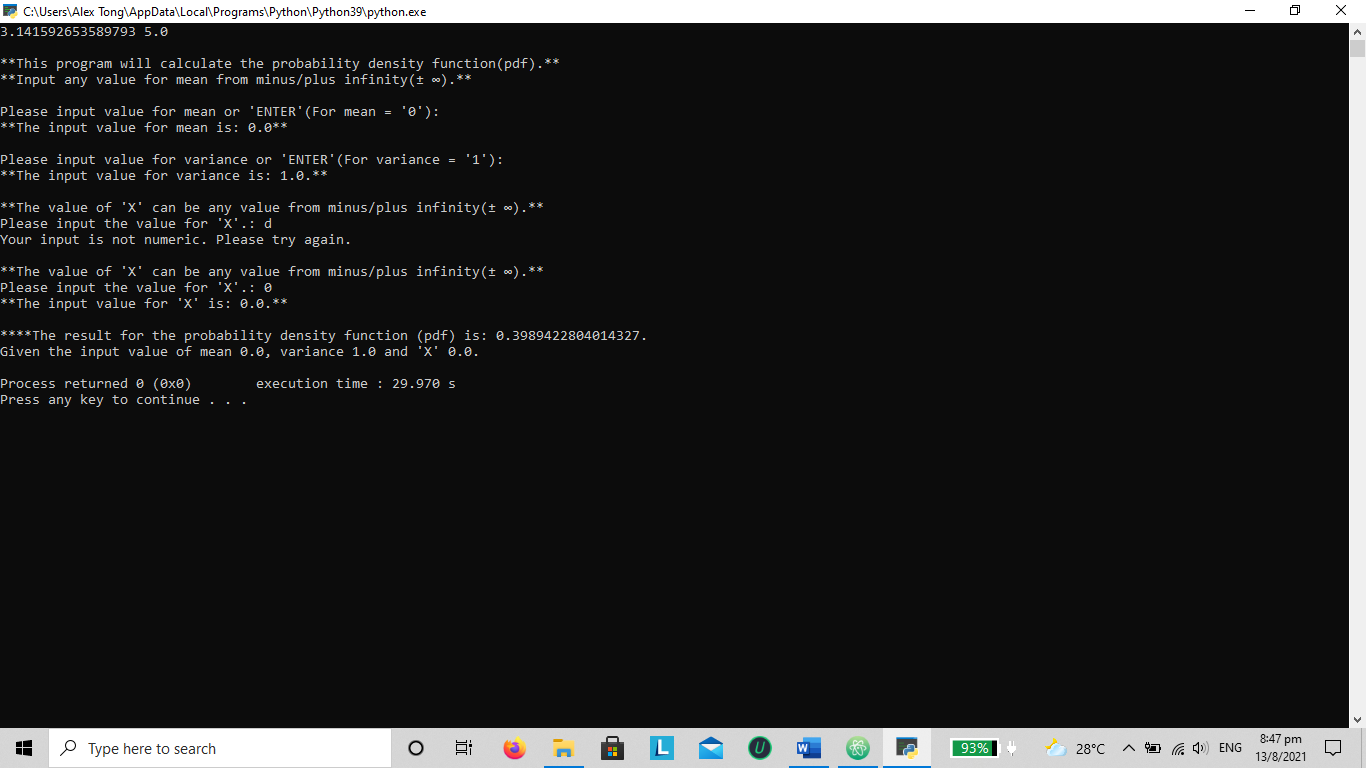
e)

# Question (e) Use formatted printing for (d).

# Input as follows: mean = 0, variance = 1, and X = 0.

print(f"\n\*\*\*\*The result for the probability density function (pdf) is: {fx\_x}. ")

print(f"Given the input value of mean {input\_mean}, variance {input\_variance}, and 'X' {input\_X}.")



f)

# Question (f) Using previous results to determine the cumulative distribution function with the given formula:

P(𝑋 ≤ 𝑘) ≈ 𝛼[𝑓𝑋(𝑎) + ⋯ + 𝑓𝑋(𝑘 − 2𝛼) + 𝑓𝑋(𝑘 − 𝛼) + 𝑓𝑋(𝑘)]

valid\_input4 = False

while valid\_input4 == False:

print("\n\*\*Alpha(α) is a range in steps.\*\*")

input\_alpha = input(f"Please input the value for alpha(α).: ")

try: # control for alpha

input\_alpha = float(input\_alpha)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input4 = True

print(f"\*\*The input value for alpha(α) is: {input\_alpha}.")

# input of value 'a' with control mechanism

valid\_input5 = False

while valid\_input5 == False:

try: # control for 'a' < 0 and variable as float

print("\n\*\*The input value for 'a' is from from minus infinity(- ∞).\*\*")

input\_a = float(input(f"Please input value for 'a'.: "))

if input\_a < 0:

print(f"\*\*The input value for variance is: {input\_a}.\*\*")

break

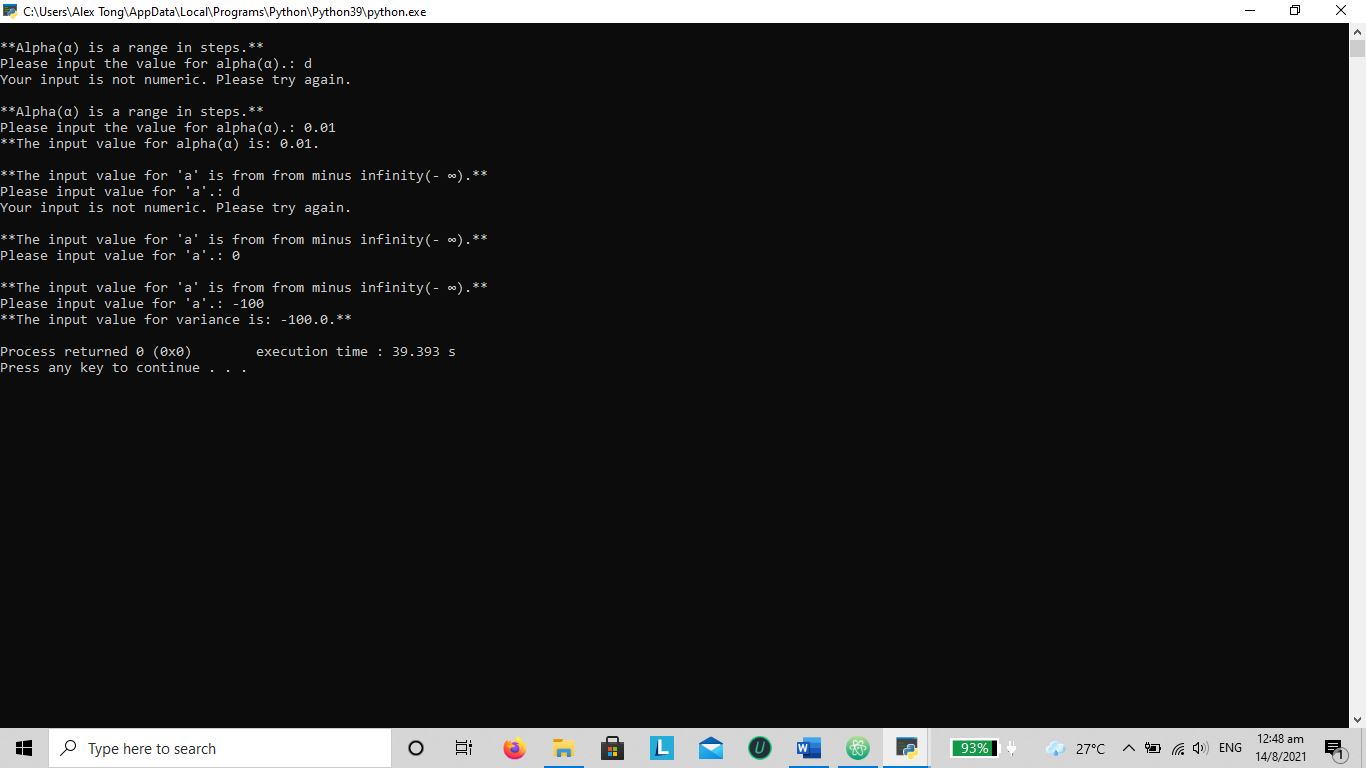
except ValueError:

print("Your input is not numeric. Please try again. ")

else:

valid\_input4 = True

Output for alpha and ‘a’ with various inputs.



# This part puts together the pieces to work-out the formula

# Formula: P(𝑋 ≤ 𝑘) ≈ 𝛼[𝑓𝑋(𝑎) + ⋯ + 𝑓𝑋(𝑘 − 2𝛼) + 𝑓𝑋(𝑘 − 𝛼) + 𝑓𝑋(𝑘)]

# I have not been able to complete from here, due to few reasons:

# I do know understand the formula, the necessarily technical skills at the moment and, while conducting the trial on excel, the outcomes were wrong

# The intention is to break the formula into smaller chunks to manage

# Rewriting the formula into: (a \* fx) \* sum(a + .. + (k - 2𝛼)) + (k - 𝛼)) + (k))

# Linking the variables from (c) to (d)

# Part 1 create a user def function

# def multipy (a, b):

# returns a \* b

# Listing of the variables:

k = input\_X # Given P(X <= k)

fx = fx\_x # fx from program returns fx(x) = 0.3989, 0.104 and 0.0584

a = input\_a

alpha = input\_alpha # will be the steps

steps = abs(a / alpha)

# part\_1 = (a \* fx)

# part\_2 = sum(a + .. + (k - 2𝛼)) + (k - 𝛼)) + (k))

# using a while loop

# i = 0

# while i <= steps

# i = i + 1

# to work the formula from sum[(k - 0) to (k - (steps \* alpha)]

# cdf = multipy(part\_1, part\_2)

# given k = 0, 1.64 and 1.96 with a mean of 0 and variance of 1

# the approximate answer for cdf(0) ≈ 0.5, cdf(1.64) ≈ 0.9495 and cdf(1.96) ≈ 0.9750

# cdf results from excel function norm.dis(X, mean, standard deviation, True)

# standard deviation is sqrt(variance), sqrt(1) = 1

g)

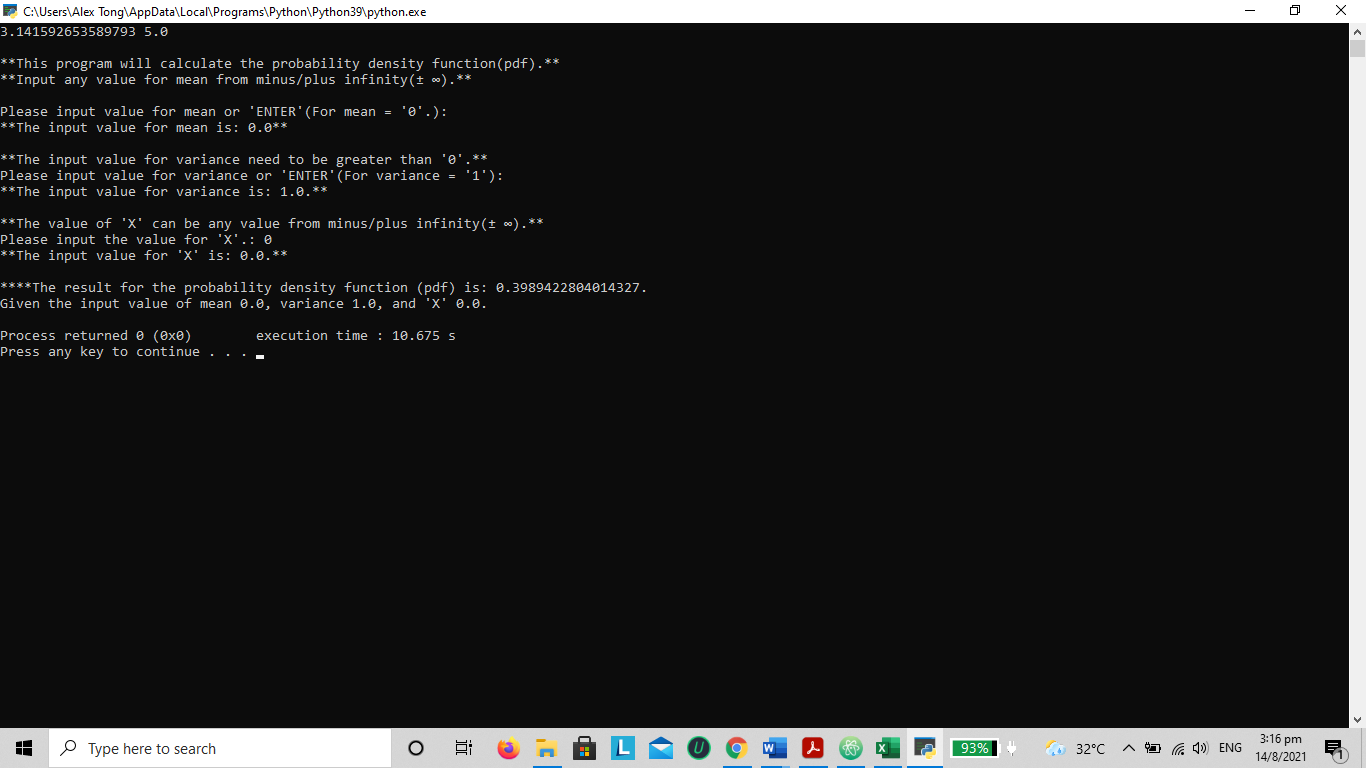
The program is written with the intention is estimate the cumulative distribution function (cdf) given the following formula:

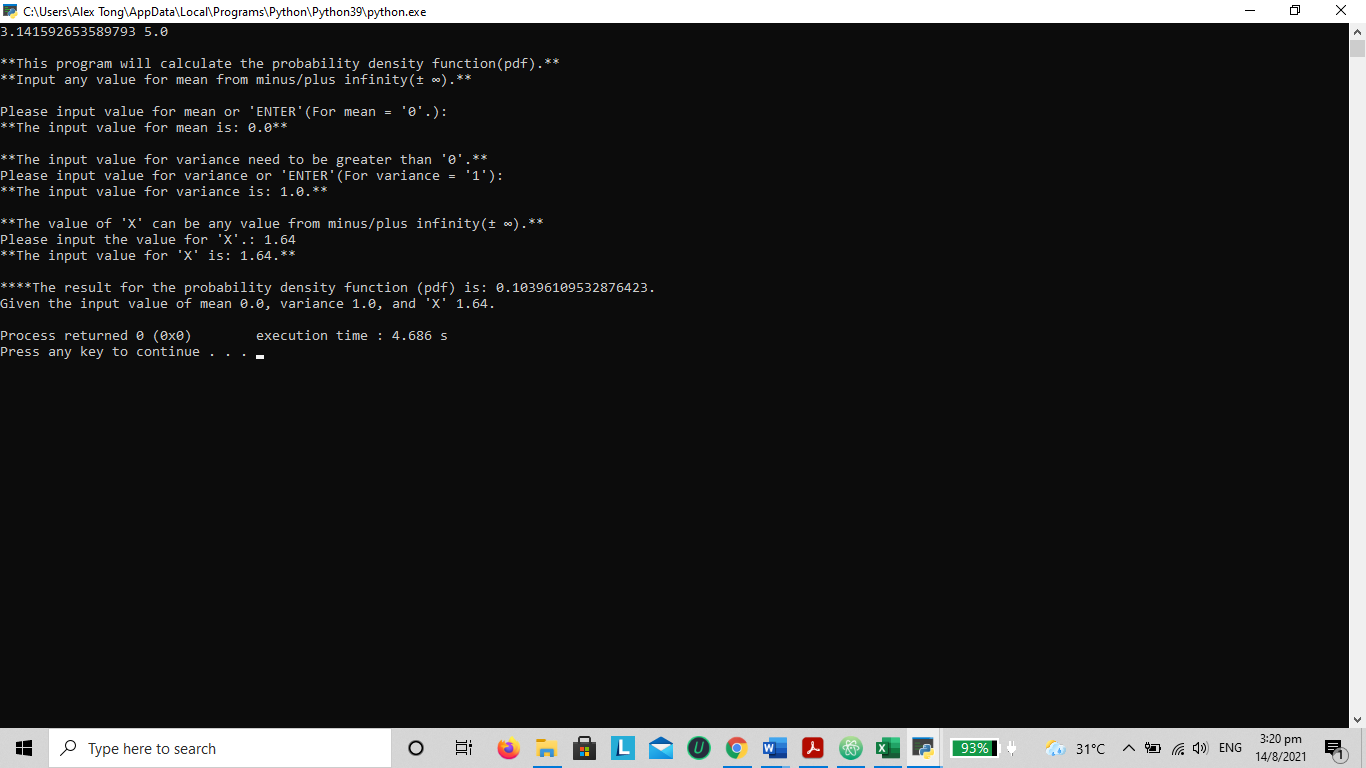
P(𝑋 ≤ 𝑘) ≈ 𝛼[𝑓𝑋(𝑎) + ⋯ + 𝑓𝑋(𝑘 − 2𝛼) + 𝑓𝑋(𝑘 − 𝛼) + 𝑓𝑋(𝑘)]

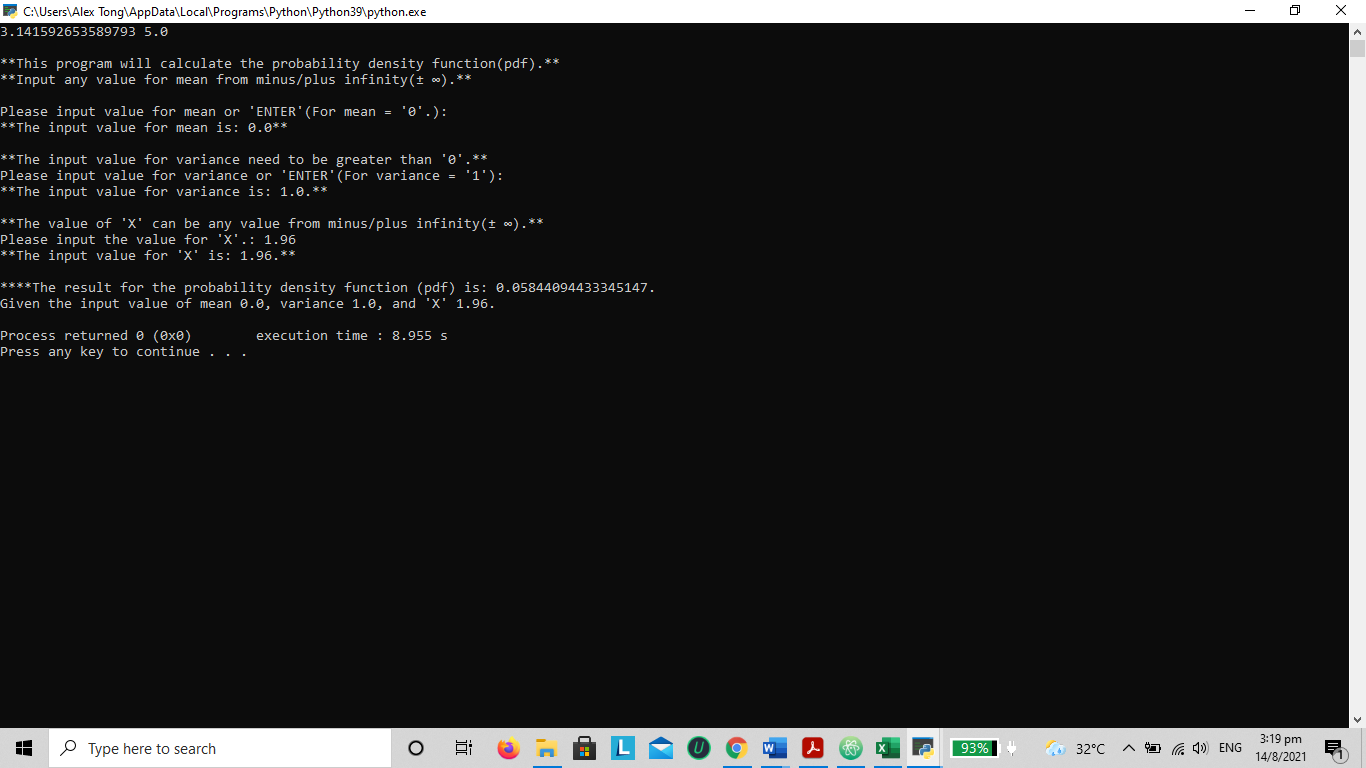
Using outputs from question (b) to (d) with the following inputs:

Mean = 0, variance = 1, and X where P(X <= k) = 0, 1.64 and 1.96

The outputs are as follows:







From the formula, alpha is multiplied by the sum of [𝑓𝑋(𝑎) + ⋯ + 𝑓𝑋(𝑘 − 2𝛼) + 𝑓𝑋(𝑘 − 𝛼) + 𝑓𝑋(𝑘)]. Create a define function by rewriting the original formula to:

(alpha \* fX) \* sum[(a) + … + (k − 2𝛼) + (𝑘 − 𝛼) + (𝑘)].

def multipy (a, b):

returns a \* b

The returned results should be approximately when k = 0, cdf ≈ 0.5, k = 1.64, cdf ≈ 0.9499 and k = 1.96, cdf ≈ 0.9750 given mean of 0 and variance of 1.

h)

# Question (h) requires the storage of value 'X' and its corresponding probability

# where x are the keys and probability as values of the dictionary

# the distribution mean is 0 and variance is 1

import math

finallist = {}

# The design for this part of the program will require user to enter 3 inputs

# and will run through a Loop using the 3 inputs to create the outcomes.

# Part 1 of the question, inputs will be X\_1 = 5, X\_2 = -5 and steps = 0.1

# Part 2 of the question, inputs will be X\_1 = 2, X\_2 = 2 and steps = 0.5

valid\_input\_1 = False

while valid\_input\_1 == False:

input\_1 = input(f"Please input the value for 'X\_1' (X\_1 is a +ve value).: ")

try: # control for 'X\_1' as a float variable

input\_1 = float(input\_1)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input\_1 = True

valid\_input\_2 = False

while valid\_input\_2 == False:

input\_2 = input(f"Please input the value for 'X\_2' (X is a -ve value).: ")

try: # control for 'X\_2' as a float variable

input\_2 = float(input\_2)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input\_2 = True

valid\_input\_3 = False

while valid\_input\_3 == False:

input\_3 = input(f"Please input the step between X\_1 and X\_2.: ")

try: # control for 'steps' as a float variable

input\_3 = float(input\_3)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input\_3 = True

# Printing of the 3 input variables: X\_1, X\_2 and steps

print(f"\nThe input values are as follows: X\_1 is {input\_1}, X\_2 is {input\_2}, and step is {input\_3}.")

# using while loop to run X for part 1 from -5 in steps of 0.01 to 5

# and part 2 when inputs are X\_1 = 2, X\_2 = -1 and step = 0.5, the

# while loop will run from -2 in steps of 0.5 to 2

i = input\_2

while i <= input\_1:

input\_X = input\_2

i = i + input\_3

def power(a, b):

return a \* math.exp(b)

# Listing of the variables:

mean = 0 # the question sets mean at 0 and variance at 1

variance = 1

x = input\_X

div\_top = x - mean

div\_top\_sq = pow(div\_top, 2)

div\_bottom = 2 \* variance

part\_b = 1 / math.sqrt(2 \* math.pi \* variance)

part\_b\_c = -(div\_top\_sq / div\_bottom)

# Using the user defined function for fx(x)

fx\_x = power (part\_b, part\_b\_c)

pdf = round(fx\_x, 4)

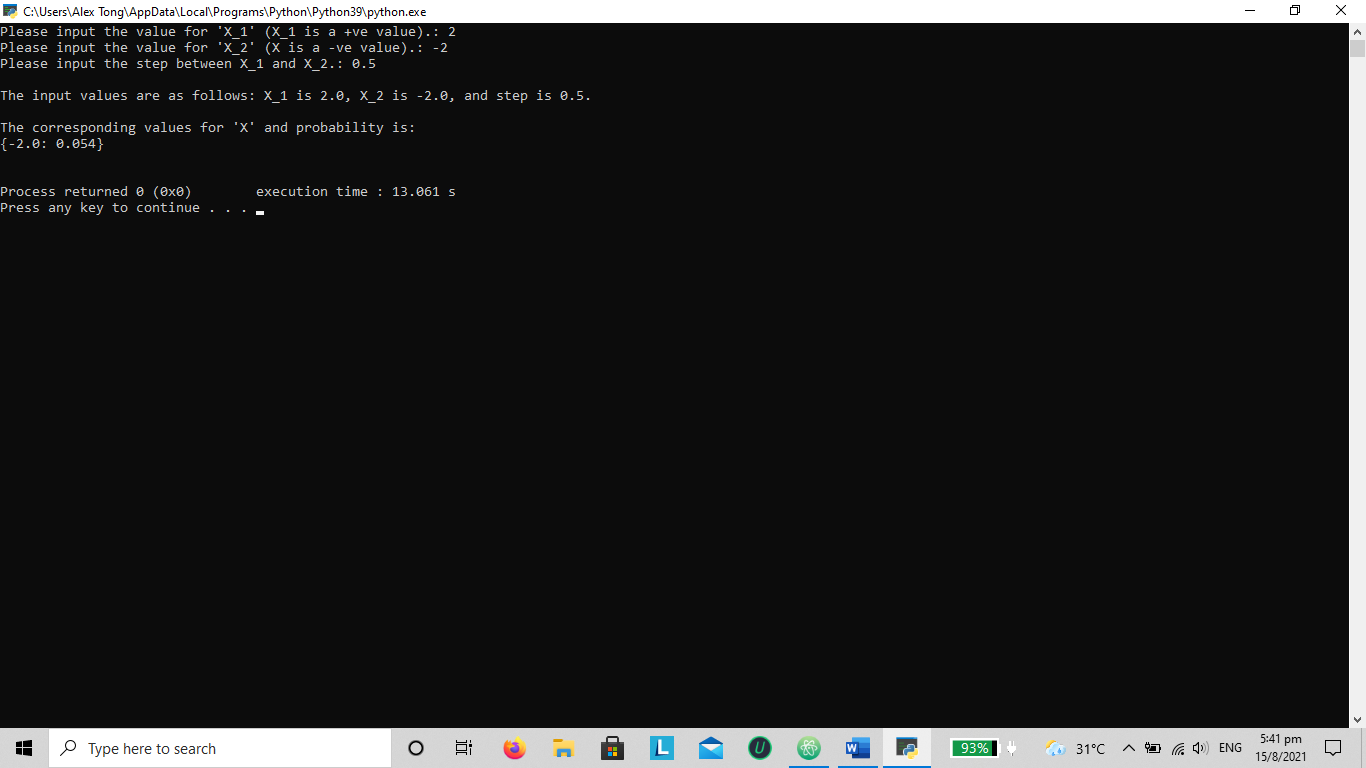
# storing the corresponding values for 'X' and probability

templist = {x: pdf}

finallist = {\*\*finallist, \*\*templist} #merge 2 dict

print(f"\nThe corresponding values for 'X' and probability is:\n{finallist}\n")

Output of X from -2 to 2 and steps at 0.5 with corresponding probability



Appendix:

# TMA project for ANL252: Python for Data Analytics

# Submitted by Tong Yew Tiong Alex PI No: H0914367

# Lecturer: Dr Munish Kumar

# submission date: 15 August 2021

# Question (a) Prepare the programming by importing the math package.

import math

print(math.pi, math.sqrt(25))

# Question (b) To employ a program that ask user to enter the mean and variance

# of the distribution. Mean is any value ± ∞, and variance > 0.

print("\n\*\*This program will calculate the probability density function(pdf).\*\*")

print("\*\*Input any value for mean from minus/plus infinity(± ∞).\*\*")

# 'Enter' for mean = 0 and control mechanism for mean.

valid\_input1 = False

while valid\_input1 == False:

input\_mean = input(f"\nPlease input value for mean or 'ENTER'(For mean = '0'.): ")

if input\_mean == "":

input\_mean = 0

try: # control mechanism for mean using try block

input\_mean = float(input\_mean)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input1 = True

print(f"\*\*The input value for mean is: {input\_mean}\*\*")

# 'Enter' for variance = 1 and control mechanism for variance.

print("\n\*\*The input value for variance need to be greater than '0'.\*\*")

input\_variance = input(f"Please input value for variance or 'ENTER'(For variance = '1'): ")

if input\_variance == "":

input\_variance = 1

input\_variance = float(input\_variance)

print(f"\*\*The input value for variance is: {input\_variance}.\*\*")

else:

valid\_input2 = False

while valid\_input2 == False:

try: # control for variance > 0 and variable as float

print("\n\*\*The input value for variance need to be greater than '0'.\*\*")

input\_variance = float(input(f"Please input value for variance.: "))

if input\_variance > 0:

print(f"\*\*The input value for variance is: {input\_variance}.\*\*")

break

except ValueError:

print("Your input is not numeric. Please try again. ")

else:

valid\_input1 = True

# Question (c) Design an input screen for user to enter the value 'X'.

# Input to be any value ± ∞ and employ control mechanism for the value.

valid\_input3 = False

while valid\_input3 == False:

print("\n\*\*The value of 'X' can be any value from minus/plus infinity(± ∞).\*\*")

input\_X = input(f"Please input the value for 'X'.: ")

try: # control for 'X' as a float variable

input\_X = float(input\_X)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input3 = True

print(f"\*\*The input value for 'X' is: {input\_X}.\*\*")

# Question (d) To construct a user-defined function using the formula of the

# probability density function, based on part (b) and (c).

def power(a, b):

return a \* math.exp(b)

# Listing of the variables:

mean = (input\_mean)

variance = (input\_variance)

x = (input\_X)

div\_top = x - mean

div\_top\_sq = pow(div\_top, 2)

div\_bottom = 2 \* variance

part\_b = 1 / math.sqrt(2 \* math.pi \* variance)

part\_b\_c = -(div\_top\_sq / div\_bottom)

# Using the user defined function for fx(x)

fx\_x = power (part\_b, part\_b\_c)

# Question (e) Use formatted printing for (d).

# Input as follows: mean = 0, variance = 1, and X = 0.

print(f"\n\*\*\*\*The result for the probability density function (pdf) is: {fx\_x}. ")

print(f"Given the input value of mean {input\_mean}, variance {input\_variance}, and 'X' {input\_X}.")

# Question (f) Using previous results to determine the cumulative distribution function

# with the given formula :P(𝑋 ≤ 𝑘) ≈ 𝛼[𝑓𝑋(𝑎) + ⋯ + 𝑓𝑋(𝑘 − 2𝛼) + 𝑓𝑋(𝑘 − 𝛼) + 𝑓𝑋(𝑘)]

valid\_input4 = False

while valid\_input4 == False:

print("\n\*\*Alpha(α) is a range in steps.\*\*")

input\_alpha = input(f"Please input the value for alpha(α).: ")

try: # control for alpha

input\_alpha = float(input\_alpha)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input4 = True

print(f"\*\*The input value for alpha(α) is: {input\_alpha}.")

# input of value 'a' with control mechanism

valid\_input5 = False

while valid\_input5 == False:

try: # control for 'a' < 0 and variable as float

print("\n\*\*The input value for 'a' is from from minus infinity(- ∞).\*\*")

input\_a = float(input(f"Please input value for 'a'.: "))

if input\_a < 0:

print(f"\*\*The input value for variance is: {input\_a}.\*\*")

break

except ValueError:

print("Your input is not numeric. Please try again. ")

else:

valid\_input4 = True

# This part puts together the pieces to workout the formula

# Formula: P(𝑋 ≤ 𝑘) ≈ 𝛼[𝑓𝑋(𝑎) + ⋯ + 𝑓𝑋(𝑘 − 2𝛼) + 𝑓𝑋(𝑘 − 𝛼) + 𝑓𝑋(𝑘)]

# I have not been able to complete from here, due to few reasons:

# I do know understand the formula, the neccessarily technical skills at the moment

# and while conducting the trial on excel, the outcomes were wrong

# The intention is to break the formula into smaller chunks to manage

# Rewriting the formula: (a \* fx) \* sum(a + .. + (k - 2𝛼)) + (k - 𝛼)) + (k))

# Linking the variables from (c) to (d)

# Part 1 create a user def function

# def multipy (a, b):

# returns a \* b

# Listing of the variables:

k = input\_X # Given P(X <= k)

fx = fx\_x # fx from program returns fx(x) = 0.3989, 0.104 and 0.0584

a = input\_a

alpha = input\_alpha # will be the steps

steps = abs(a / alpha)

# part\_1 = (a \* fx)

# part\_2 = sum(a + .. + (k - 2𝛼)) + (k - 𝛼)) + (k))

# using a while loop

# i = 0

# while i <= steps

# i = i + 1

# to work the formula from sum[(k - 0) to (k - (steps \* alpha)]

# cdf = multipy(part\_1, part\_2)

# given k = 0, 1.64 and 1.96 with a mean of 0 and variance of 1

# the approximate answer for cdf(0) ≈ 0.5, cdf(1.64) ≈ 0.9495 and cdf(1.96) ≈ 0.9750

# cdf results from excel function norm.dis(X, mean, standard deviation, True)

# standard deviation is sqrt(variance), sqrt(1) = 1

# Question (h) reuires the storage of value 'X' and its corresponding probability

# where x are the keys and probability as values of the dictionary

# the distribution mean is 0 and variance is 1

import math

finallist = {}

# Question (h) is done on a separate file due to part (f)

# The design for this part of the program will require user to enter 3 inputs

# and will run through a Loop using the 3 inputs to create the outcomes.

# Part 1 of the question, inputs will be X\_1 = 5, X\_2 = -5 and steps = 0.1

# Part 2 of the question, inputs will be X\_1 = 2, X\_2 = 2 and steps = 0.5

valid\_input\_1 = False

while valid\_input\_1 == False:

input\_1 = input(f"Please input the value for 'X\_1' (X\_1 is a +ve value).: ")

try: # control for 'X\_1' as a float variable

input\_1 = float(input\_1)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input\_1 = True

valid\_input\_2 = False

while valid\_input\_2 == False:

input\_2 = input(f"Please input the value for 'X\_2' (X is a -ve value).: ")

try: # control for 'X\_2' as a float variable

input\_2 = float(input\_2)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input\_2 = True

valid\_input\_3 = False

while valid\_input\_3 == False:

input\_3 = input(f"Please input the step between X\_1 and X\_2.: ")

try: # control for 'steps' as a float variable

input\_3 = float(input\_3)

except ValueError:

print("Your input is not numeric. Please try again.")

else:

valid\_input\_3 = True

# Printing of the 3 input variables: X\_1, X\_2 and steps

print(f"\nThe input values are as follows: X\_1 is {input\_1}, X\_2 is {input\_2}, and step is {input\_3}.")

# using while loop to run X for part 1 from -5 in steps of 0.01 to 5

# and part 2 when inputs are X\_1 = 2, X\_2 = -1 and step = 0.5, the

# while loop will run from -2 in steps of 0.5 to 2

i = input\_2

while i <= input\_1:

input\_X = input\_2

i = i + input\_3

def power(a, b):

return a \* math.exp(b)

# Listing of the variables:

mean = 0 # the question sets mean at 0 and variance at 1

variance = 1

x = input\_X

div\_top = x - mean

div\_top\_sq = pow(div\_top, 2)

div\_bottom = 2 \* variance

part\_b = 1 / math.sqrt(2 \* math.pi \* variance)

part\_b\_c = -(div\_top\_sq / div\_bottom)

# Using the user defined function for fx(x)

fx\_x = power (part\_b, part\_b\_c)

pdf = round(fx\_x, 4)

# storing the corresponding values for 'X' and probability

templist = {x: pdf}

finallist = {\*\*finallist, \*\*templist} #merge 2 dict

print(f"\nThe corresponding values for 'X' and probability is:\n{finallist}\n")