**(a)**

**Code:**

#importing math package

import math

**(b)**

**Code:**

#Determine Mean, Variance, X when user does not enter any input

miu = 0

var = 1

#If Else statements: User can only input numeric values

mean = input("Please enter the mean: ")

if mean == "":

print (f"Mean entered has been auto defaulted to {miu}")

else:

mean = float(mean)

if isinstance (mean, float) == False:

print(f"Error! You have entered a non-numeric value of the mean.")

else:

miu = float(mean)

print (f"Mean: {miu}")

#Determine Variance entered by User

#If Else statements, User can only enter positive infinity numbers, non-zero, and anything non-numeric.

#While loop until user enters input to meet requirement

variance = input("Please enter the variance: ")

if variance == "":

print (f"Variance is auto defaulted to : {var}")

else:

while float(variance) <= 0:

variance = input("Please input another value: ")

else:

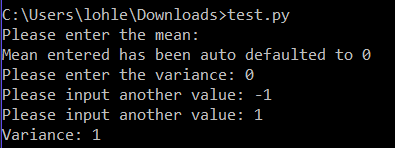
var = float(variance)

print(f"Variance: {var} ")

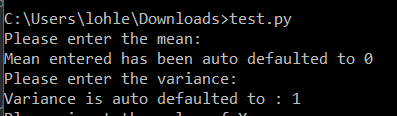
**(b)**

**Results:**

When the user enters a variance value smaller or equal to 0:

****

When the user press ENTER without providing values for the variance and mean:



**(c)**

**Code:**

#User enters value of X

#Value can only be integers, python will reject if its not an integer

while True:

try:

X = float(input("Please input the value of X: "))

except ValueError:

print(f"Error!: You have enter a non-numeric value of X.")

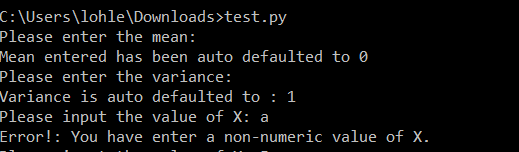
continue

else:

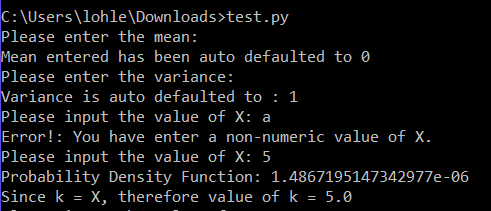
break

**Results:**

X-value can only be a integer. Python will reject it if it is not a number.



Python accepts X-value as it is an integer:



**(d)**

**Code:**

#PDF Function

def pdf(miu, var, X):

c = 1/(math.sqrt(math.tau \* var))

d = (-(X-miu)\*\*2)/(2\*var)

return (c \* (math.exp(d)))

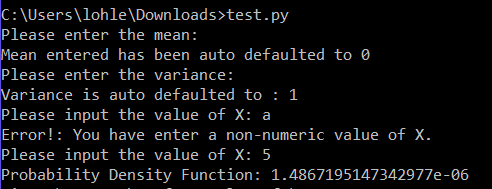
**(e)**

**Code:**

#Formatted printing

print(f"Probability Density Function: {pdf(miu, var, X)}")

**Results:**

****

**(f)**

**Code:**

k = X

print(f"Since user input value of X = value of k, therefore k = {X}")

#Value of a can only be negative infinity.

while True:

try:

a = float(input("Please input the value of a: "))

except ValueError:

print(f"Error!: You have enter a non-numeric value of a.")

continue

else:

break

if a == "":

a = float(input("Please input another value of a: "))

else:

a = float(a)

print(f"Value of a: {a} ")

#Value of alpha is a decimal number. E.g. 0.01, 0.1 etc

while True:

try:

alpha = float(input("Please input the value of alpha: "))

except ValueError:

print(f"Error!: You have enter a non-numeric value of alpha.")

continue

else:

break

if alpha == "":

alpha = float(input("Please input another value of Alpha: "))

else:

alpha = float(alpha)

print(f"Alpha value: {alpha} ")

cdf = 0

#fx(a) + fx(pattern) + fx(k)

#as long as fx(k) is smaller than fx(a), python will keep calculating fx (pattern) until it reaches a point where fx(a) is smaller than fx(k).

dict = dict()

while a <=k:

cdf = pdf(miu, var, X)

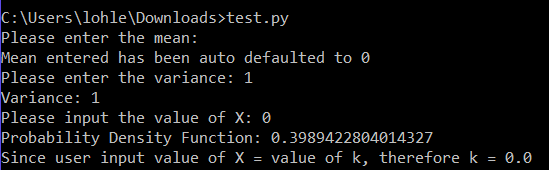
cdf = alpha\*cdf

dict[round(a, 4)] = round(cdf, 4)

a+=alpha

**Results:**

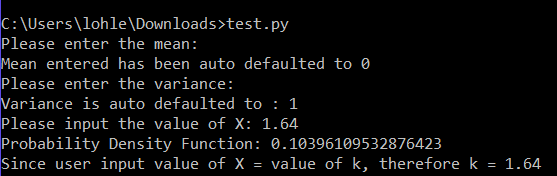
**When k = 0, mean = 0, variance = 1:**

****

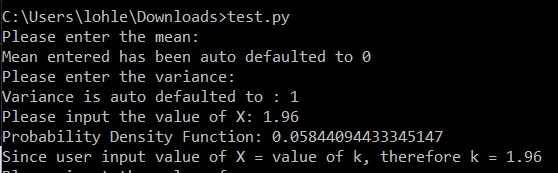
**(f)**

**Results:**

When k = 1.64, mean = 0, variance = 1:

****

When k = 1.96, mean = 0, variance = 1:

****

**(g)**

1. While loop & try except – value of X (user input); Ensures that only numeric value of X can only be entered. Python will reject anything that is non-numeric.

2. Function created: pdf; to calculate the function 𝑓𝑋 (x) formula

3. Since value of k = Value of X input by user, therefore k = X.

4. Value of a (random variable X); Number that is close to negative infinity.

Try except & if else statement; Ensures that only numeric value of a can only be entered. Python will reject anything that is non-numeric, including blanks etc. Value of a = Value of random variable X

5. Value of a ; This is the step range. Try except & if else statement. It ensures that only numeric value of alpha can only be entered. Python will reject anything that is non-numeric, including blanks etc.

6. Dictionary & array - cdf; continuously calculates (function pdf multiplied to alpha, and stops once a = k) and stores all probabilities (with the corresponding x) of those x’s between X and k.

As long as random variable X is smaller or equal to k. It prints all probabilities (with the corresponding x) of those x’s between X and k. Note: this “X” refers to random variable X.

**(h)**

**Code:**

#dictionary

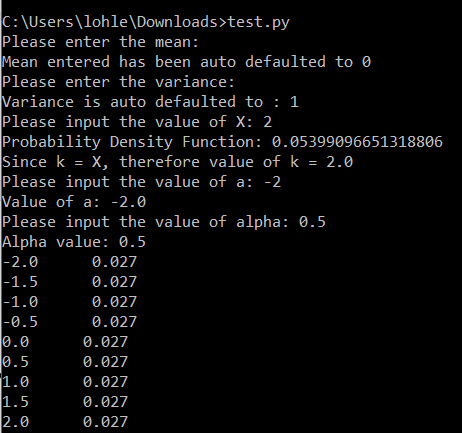
for key in dict:

print(key, end = " ")

print(dict[key])

**Results:**

When mean = 0, variance = 1, print x = 2, when alpha = 0.5, a = -2:

****

**Imported TMA01 coding file:**

****