Python Programming

Homework 1

Jordan Diaz

Solution #1:

# Jordan Diaz  
# This Program solves quadratic equations and graphs them  
import math  
import matplotlib.pyplot as plt  
  
  
def plot\_quadratic(var\_a, var\_b, var\_c, domain\_min, domain\_max, points, style):  
 *""" Function plots a quadratic function, needs a, b, c, min x, max x, number of points, and style"""* xs = []  
 ys = []  
 # prepare the domain for the function we graph  
 x = domain\_min  
 difference = (domain\_max - domain\_min) / points  
  
 while x <= domain\_max:  
 xs.append(x)  
  
 y = (var\_a \* x \*\* 2) + (var\_b \* x) + var\_c  
  
 ys.append(y)  
 x += difference  
  
 plt.plot(xs, ys, style)  
 plt.show()  
  
  
while True:  
 a = input("type the value of a: ")  
  
 if a == "":  
 break  
 a = float(a)  
  
 b = float(input("type the value of b: "))  
 c = float(input("type the value of c: "))  
  
 discriminant = (b \*\* 2) - (4 \* a \* c)  
  
 if discriminant < 0:  
 print("no real solutions")  
 plot\_quadratic(a, b, c, (-b / (2 \* a)), 4, 150, "rx-")  
 elif discriminant == 0:  
 x1 = x2 = (-b + math.sqrt(discriminant)) / (2 \* a)  
 print("one solution: ", x1)  
 plot\_quadratic(a, b, c, x1 - 2, x2 + 2, 150, "rx-")  
 elif discriminant > 0:  
 x1 = (-b - math.sqrt(discriminant)) / (2 \* a)  
 x2 = (-b + math.sqrt(discriminant)) / (2 \* a)  
 print("two solutions: {} and {}".format(x1, x2))  
 plot\_quadratic(a, b, c, x1 - 2, x2 + 2, 150, "rx-")  
  
 print()

Terminal Session for problem #1:

Calendar

Description automatically generated

Graph for a = 1, b = -1, c = -6

Shape

Description automatically generated

Solution #2:

# Jordan Diaz

# This program computes all possible Pythagorean triples (a, b, c)  
# This is similar to an assignment I did in discrete math  
# I learned that (n, n, n) = n^3 possibilities where each available position is another for loop  
# If there were quadruples instead of triples there would be 4 for loops  
  
def find\_pythagorean(n):  
 *"""This Function will go through every possibility of a, b, c and return a list of triples"""* triples\_list = []  
  
 for i in range(1, n + 1): # n^1  
 for j in range(1, n + 1): # n^2  
 for k in range(1, n + 1): # n^3  
 if ((i \*\* 2) + (j \*\* 2)) == k \*\* 2:  
 triple = (i, j, k)  
 triples\_list.insert(triples\_list.count(0), triple)  
 triples\_list.reverse()  
 return triples\_list  
  
  
for triples in find\_pythagorean(int(input("Please Enter a positive Integer: "))):  
 print(triples)

Terminal Session for problem #2

Text

Description automatically generated

Solution #3:

# Jordan Diaz  
# This Program finds duplicate substring and max duplicate substring  
# Cannot use str functions  
  
def find\_dup\_str(s, n):  
 *""" This Function determines returns the first occurring substring of length n (if any) """* if n >= len(s) or n < 1:  
 return ""  
  
 for i in range(0, len(s) - 1):  
 curr\_str = s[i: i + n]  
  
 for k in range(i + 1, len(s) - 1):  
 next\_str = s[k:k + n]  
  
 if curr\_str == next\_str:  
 return curr\_str  
 return ""  
  
  
def find\_max\_dup(s):

*""" This Function returns the largest duplicate substring"""*

curr\_str = ""  
 n = len(s) // 2  
  
 while curr\_str == "":  
 curr\_str = find\_dup\_str(s, n)  
 n -= 1  
  
 if n <= 0:  
 break  
  
 return curr\_str  
  
  
print(find\_dup\_str(input("Type a string to find first substring: "), int(input("Type the length of the substring: "))))  
print(find\_max\_dup(input("Type a string to find max substrings: ")))

Terminal Session for problem #3:

Text

Description automatically generated

Solution #4:

# Jordan Diaz  
# This program will plot any function evaluated from the terminal with a table

import math  
import matplotlib.pyplot as plotter  
  
  
def plot\_function(fun\_str, domain, points, style):  
 *""" Function plots a given function"""* xs = []  
 ys = []  
 # prepare the domain for the function we graph  
 x = domain[0]  
 difference = (domain[1] - domain[0]) / points  
  
 while x <= domain[1]:  
 xs.append(x)  
 x += difference  
  
 for x in xs:  
 y = eval(fun\_str)  
 ys.append(y)  
  
 plotter.plot(xs, ys, style)  
 plotter.xlabel("x - axis")  
 plotter.ylabel("y - axis")  
 plotter.title(fun\_str)  
 plotter.show()  
 print("{:>7s} {:>10s} ".format("x", "y"))  
 print("---------------------------")  
 for x in xs:  
 y = eval(fun\_str)  
 print("{:10.4f} {:10.4f} ".format(x, y))  
  
  
plot\_function(input("Enter a function with variable x: "),  
 (float(input("Enter x-min: ")), float(input("Enter x-max: "))),  
 int(input("Enter number of samples: ")),  
 "rx-")

A picture containing table

Description automatically generated

Terminal Session problem #4: (too long to show all)

Graph for Solution #4 Terminal input:

Chart, line chart

Description automatically generated