**Name: Session:**

**Programming I**

**Lab Exercise 11.9.2023**

**When you have completed these programs, submit your source code.**

1. **Using a Point Class**

Write a function distance(p1, p2) that returns the distance between two points. I have provided a Point class for you to use:

from point import Point

alternatively, if you do not wish to use my Point class your distance function should be of the form:

def distance(x1, y1, x2, y2)

In order to calculate the distance you should use the Pythagorean Theorem



1. **Using 3D Point Class**

Write a function distance3D(p1, p2) that returns the distance between two points in 3D space. I have provided a Point3d class for you to use:

from point import Point

alternatively, if you do not wish to use my Point class your distance function should be of the form:

def distance3d(x1, y1, z1, x2, y2, z2)

In order to calculate the distance you should use the Pythagorean Theorem



1. **Collision Detection of Balls**

Many games have complex physics engines, and one major function of these engines is to figure out if two objects are colliding. Weirdly-shaped objects are often approximated as balls. In this problem, we will figure out if two balls are colliding.

We will think in 2D to simplify things, though 3D isn’t different conceptually. For calculating collision, we only care about a ball’s position in space and its size. We can store position with its center x-y coordinates, and we can use its radius for size. So a ball is a tuple of (x, y, r). To figure out if two balls are colliding, we need to compute the distance between their centers, and then see if this distance is less than the sum of their radii. If so, they are colliding.

Write a function collide(t1, t2) that takes two balls (tuples) as parameters and computes if they are colliding. Then call the function with two sets of balls. The first set is (0, 0, 1) and (3, 3, 1); these should not be colliding. The second set is (5, 5, 2) and (2, 8, 3); these should be colliding.

Notes on the use of Tuples:

ball1 = (1, 2, 3) refers to a ball with its center at point (1, 2) and a radius of 3. You can access the elements of the Tuple as such:

center = Point(ball1[0], ball1[2])

radius = ball1[2]

1. **Collision Detection in 3D**

Modify the function in problem 3 so that it works in 3D.

1. **Create a List of Prime Numbers**

A positive whole number n > 2 is prime if no number between 2 and  (inclusive) evenly divides n. You have already written a function that accepts a value n as input and determines if the value is prime. Use the previous function to write a function that returns a list of every prime number less than or equal to n.

1. **Modify the above program to factor an integer**
2. **Calculating Pi Using Monte Carlo Methods**

In this activity we will be using random number generation to find the value of Pi. Image a square box with a side of one unit with a quarter circle inside of it that has a radius of 1. The area of the box is 1.0 and the area of the quarter circle is Pi/4. If we generate random (x, y) coordinates from (0.0, 0.0) to (1.0, 1.0) the probability that the point is inside the circle is the ratio of these areas (Pi/4).

1. Start IDLE and create a new window.
2. Import the random library

import random

1. Initialize variables

SIZE = 100 #define number of trials

count = 0

1. Conduct the simulation

for i in range(1, SIZE):

#generate random (x, y) coordinate

x = random.random()

y = random.random()

#test to see if random coordinate is inside unit circle

if x\*x + y\*y < 1:

count = count + 1

1. Calculate Pi

pi = 4.0 \* count/SIZE

1. Print results. Notice the commas in the print statement. The purpose of these is to suppress CRLF (Carriage Return Line Feed).

print( "With ", SIZE, "trials Pi = ", pi)

1. Now run your program. Try adjusting SIZE and fill out this table.

|  |  |
| --- | --- |
| **Size** | **Value of Pi** |
| 10 |  |
| 100 |  |
| 1000 |  |
| 10000 |  |
| 100000 |  |
| 1000000 |  |

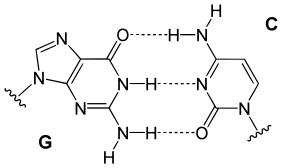
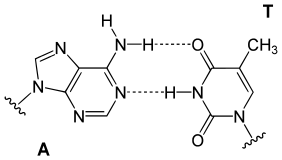
1. **Creating a complement DNA sequence**

Last week we generated a million base DNA sequence. In reality, a DNA strand is to DNA base chain tied to another chain consisting of its complement (base pairing). Guanine pairs with Cytosine and Adenine with Thymine. For example:

A base-paired DNA sequence:

ATCGATTGAGCTCTAGCG

TAGCTAACTCGAGATCGC

Your task is to take a million base DNA strand and build its complementary strand. This process takes place in cell production. In testing your program you should build a 100 base strand and it’s complement to print. **Do not print the 1 million base strand or complement**.