|  |
| --- |
| **Course**: Programming Fundamental – ENSF 337  **Lab #**: Lab 5  **Instructor**: M. Moussavi  **Student Name**: Aarushi Roy Choudhury  **Lab Section**: B01  **Date submitted**: Nov 1,2021 |

**Exercise A**

Diagram, schematic

Description automatically generated

**Exercise B**

Diagram, schematic

Description automatically generated

**Exercise D**

**Exercise E**

// ENSF 337

//Lab5exe\_E.c

//Completed By: Aarushi Roy Choudhury

#include "lab5exE.h"

#include <stdio.h>

#include <math.h>

#include <string.h>

int main(void){

    Point alpha = {"A1", 2.3, 4.5, 56.00};

    Point beta = {"B1", 25.9, 30.0, 97.00};

    printf ("Display the values in alpha, and beta: ");

    display\_struct\_point(alpha);

    display\_struct\_point(beta);

    Point \*stp = &alpha;

    printf ("\n\nDisplay the values in \*stp: ");

    display\_struct\_point(\*stp);

    Point gamma = mid\_point(stp, &beta, "M1");

    printf ("\n\nDisplay the values in gamma after calling mid\_point function...");

    printf ("\nExpected result is: M1 <14.10, 17.25, 76.50>");

    printf("\n\nThe actual result of calling mid\_point function is: ");

    display\_struct\_point(gamma);

    swap (stp, &beta);

    printf ("\n\nDisplay the values in \*stp, and beta after calling swap function... ");

    printf ("Expected to be: \nB1 = <25.90, 30.00, 97.00> \nA1 = <2.30, 4.50, 56.00>");

    printf("\n\nThe actual result of calling swap function is: ");

    display\_struct\_point(\*stp);

    display\_struct\_point(beta);

    printf("\n\nThe distance between alpha and beta is: %.2f. (Expected to be: 53.74)", distance(&alpha, &beta));

    printf("\nThe distance between gamma and beta is: %.2f. (Expected to be: 26.87) \n", distance(&gamma, &beta));

    return 0;

}

void display\_struct\_point(const Point p){

    printf("\n%s <%.2lf, %.2lf, %.2lf>", p.label, p.x, p.y, p.z);

}

Point mid\_point(const Point \*p1, const Point \*p2, const char \*label){

    //YOU ARE NOT ALLOWED TO USE ANY STRING LIBRARY FUNCTIONS IN THIS FUNCTION

        //Create new point named label that is in the middle of p1 and p2

        int i;

        Point middle;

        double x=((\*p1).x+(\*p2).x)/2;

        double y=((\*p1).y+(\*p2).y)/2;

        double z=((\*p1).z+(\*p2).z)/2;

        for(i=0; label[i] != '\0'; i++){

                middle.label[i]=label[i];

        }

        middle.label[i]='\0';

        middle.x=x;

        middle.y=y;

        middle.z=z;

    return middle;

}

void swap(Point \*p1, Point \*p2){

        //Swaps the values of p1 and p2

        Point temp=\*p1;

        \*p1=\*p2;

        \*p2=temp;

}

double distance(const Point \*p1, const Point \*p2){

    //YOU ARE NOT ALLOWED TO USE THE ARROW OPERATOR (->)

        //Finds the distance between p1 and p2

        double d=sqrt(pow(((\*p1).x-(\*p2).x), 2) + pow(((\*p1).y-(\*p2).y), 2) + pow(((\*p1).z-(\*p2).z), 2));

    return d;

}

Text

Description automatically generated

**Exercise F**