#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<stdbool.h>

#include<math.h>

// Function prototypes

int circle(char a[]);

int triangle(char a[]);

int square(char a[]);

int rectangle(char a[]);

int pentagon(char a[]);

int hexagon(char a[]);

int heptagon(char a[]);

int octagon(char a[]);

int nanogon(char a[]);

int decagon(char a[]);

int trapezium(char a[]);

float circleArea(float radius);

float squareArea(float side);

float rectangleArea(float length, float width);

float triangleArea(float base, float height);

float pentagonArea(float side);

float hexagonArea(float side);

float heptagonArea(float side);

float octagonArea(float side);

float nonagonArea(float side);

float decagonArea(float side);

float trapeziumArea(float base1, float base2, float height);

double sin\_func(double x);

double cos\_func(double x);

double tan\_func(double x);

double exp\_func(double x);

double trapezoidal\_rule(double (\*func)(double), double a, double b, int n);

double polynomial\_func\_1(double x, int degree, double coefficients[]); // Polynomial function

double trapezoidal\_rule\_1(double (\*func)(double, int, double[]), double a, double b, int degree, double coeff[], int n);

int main() {

int choice;

float radius, side, length, width, base, height, base1, base2;

double a, b;

int n;

int expression;

int degree1, degree2;

double a1, b1;

int n1;

printf("Enter the number of choice\n");

printf("1-For regular shapes\n2-For area enclosed by curves\n3-For area enclosed by polynomials\n");

scanf("%d",&expression);

if(expression==1){ printf("Choose a shape to calculate its area:\n");

printf("1. Circle\n");

printf("2. Square\n");

printf("3. Rectangle\n");

printf("4. Triangle\n");

printf("5. Pentagon\n");

printf("6. Hexagon\n");

printf("7. Heptagon\n");

printf("8. Octagon\n");

printf("9. Nonagon\n");

printf("10. Decagon\n");

printf("11. Trapezium\n");

printf("Enter the name of the shape :\n");

char m[20];

scanf("%s",m);

if (circle(m)==strlen(m)){

printf("Enter the value of Radius : ");

scanf("%f",&radius);

printf("%0.2f",circleArea(radius));

}

else if(square(m)==strlen(m)){

printf("Enter the value of Side : ");

scanf("%f",&side);

printf("%0.2f",squareArea(side));

}

else if(rectangle(m)==strlen(m)){

printf("Enter the length and width of the rectangle: ");

scanf("%f %f", &length, &width);

printf("Area of the rectangle: %.2f\n", rectangleArea(length, width));

}

else if(triangle(m)==strlen(m)){

printf("Enter the base and height of the triangle: ");

scanf("%f %f", &base, &height);

printf("Area of the triangle: %.2f\n", triangleArea(base, height));

}

else if(pentagon(m)==strlen(m)){

printf("Enter the side of the pentagon: ");

scanf("%f", &side);

printf("Area of the pentagon: %.2f\n", pentagonArea(side));

}

else if(hexagon(m)==strlen(m)){

printf("Enter the side of the hexagon: ");

scanf("%f", &side);

printf("Area of the hexagon: %.2f\n", hexagonArea(side));

}

else if(heptagon(m)==strlen(m)){

printf("Enter the side of the heptagon: ");

scanf("%f", &side);

printf("Area of the heptagon: %.2f\n", heptagonArea(side));

}

else if(octagon(m)==strlen(m)){

printf("Enter the side of the octagon: ");

scanf("%f", &side);

printf("Area of the octagon: %.2f\n", octagonArea(side));

}

else if(nanogon(m)==strlen(m)){

printf("Enter the side of the nonagon: ");

scanf("%f", &side);

printf("Area of the nonagon: %.2f\n", nonagonArea(side));

}

else if(decagon(m)==strlen(m)){

printf("Enter the side of the decagon: ");

scanf("%f", &side);

printf("Area of the decagon: %.2f\n", decagonArea(side));

}

else if(trapezium(m)==strlen(m)){

printf("Enter the lengths of the two bases and the height of the trapezium: ");

scanf("%f %f %f", &base1, &base2, &height);

printf("Area of the trapezium: %.2f\n", trapeziumArea(base1, base2, height));

}

else{

printf("Invalid choice\n");

}

return 0;

}

if(expression==2){

printf("Choose two functions to calculate the area between them:\n");

printf("1. sin(x)\n");

printf("2. cos(x)\n");

printf("3. tan(x)\n");

printf("4. exp(x)\n");

printf("Enter your choice for the first function (1-4): ");

scanf("%d", &choice);

double (\*func1)(double);

switch (choice) {

case 1:

func1 = sin\_func;

break;

case 2:

func1 = cos\_func;

break;

case 3:

func1 = tan\_func;

break;

case 4:

func1 = exp\_func;

break;

default:

printf("Invalid choice.\n");

return 1;

}

printf("Enter your choice for the second function (1-4): ");

scanf("%d", &choice);

double (\*func2)(double);

switch (choice) {

case 1:

func2 = sin\_func;

break;

case 2:

func2 = cos\_func;

break;

case 3:

func2 = tan\_func;

break;

case 4:

func2 = exp\_func;

break;

default:

printf("Invalid choice.\n");

return 1;

}

printf("Enter the lower bound of integration: ");

scanf("%lf", &a);

printf("Enter the upper bound of integration: ");

scanf("%lf", &b);

printf("Enter the number of subintervals: ");

scanf("%d", &n);

// Calculate the area between the curves using the trapezoidal rule

double h = (b - a) / n;

double area = 0.0;

double x1 = a;

double x2, y1;

for (int i = 0; i < n; i++) {

x2 = x1 + h;

y1 = func1(x1) - func2(x1);

if (y1 < 0) {

double c = (x1 + x2) / 2; //fabs is a function used to make the absolute floating point

double area\_interval = fabs(trapezoidal\_rule(func1, x1, c, n) - trapezoidal\_rule(func2, x1, c, n))

+ fabs(trapezoidal\_rule(func1, c, x2, n) - trapezoidal\_rule(func2, c, x2, n));

area += area\_interval;

} else {

area += 0.5 \* (func1(x1) + func1(x2)) \* h - 0.5 \* (func2(x1) + func2(x2)) \* h;

}

x1 = x2;

}

printf("Area between the curves: %.6lf\n", area);

return 0;

}

if(expression==3){

printf("Enter the degree of the first polynomial: ");

scanf("%d", &degree1);

printf("Enter the degree of the second polynomial: ");

scanf("%d", &degree2);

printf("Enter the lower bound of integration: ");

scanf("%lf", &a1);

printf("Enter the upper bound of integration: ");

scanf("%lf", &b1);

printf("Enter the number of subintervals: ");

scanf("%d", &n1);

double coeff1[degree1 + 1], coeff2[degree2 + 1];

printf("Enter the coefficients for the first polynomial (separated by spaces, starting from x^0 to x^degree1): ");

for (int i = 0; i <= degree1; i++)

scanf("%lf", &coeff1[i]);

printf("Enter the coefficients for the second polynomial (separated by spaces, starting from x^0 to x^degree2): ");

for (int i = 0; i <= degree2; i++)

scanf("%lf", &coeff2[i]);

// Calculate the area between the curves using the trapezoidal rule

double area1 = fabs(trapezoidal\_rule\_1(polynomial\_func\_1, a1, b1, degree1, coeff1, n1) - trapezoidal\_rule\_1(polynomial\_func\_1, a1, b1, degree2, coeff2, n1));

printf("Area between the curves: %.6lf\n", area1);

return 0;

}

}

// Function definitions

float circleArea(float radius) {

return M\_PI \* radius \* radius;

}

float squareArea(float side) {

return side \* side;

}

float rectangleArea(float length, float width) {

return length \* width;

}

float triangleArea(float base, float height) {

return 0.5 \* base \* height;

}

float pentagonArea(float side) {

return 0.25 \* sqrt(5 \* (5 + 2 \* sqrt(5))) \* side \* side;

}

float hexagonArea(float side) {

return (3 \* sqrt(3) / 2) \* side \* side;

}

float heptagonArea(float side) {

return 0.25 \* (7 \* tan(M\_PI / 7)) \* side \* side;

}

float octagonArea(float side) {

return 2 \* (1 + sqrt(2)) \* side \* side;

}

float nonagonArea(float side) {

return 0.25 \* (9 \* tan(M\_PI / 9)) \* side \* side;

}

float decagonArea(float side) {

return 2.5 \* side \* side \* (1 + sqrt(5));

}

float trapeziumArea(float base1, float base2, float height) {

return 0.5 \* (base1 + base2) \* height;

}

double sin\_func(double x) {

return sin(x);

}

double cos\_func(double x) {

return cos(x);

}

double tan\_func(double x) {

return tan(x);

}

double exp\_func(double x) {

return exp(x);

}

double trapezoidal\_rule(double (\*func)(double), double a, double b, int n) {

double h = (b - a) / n;

double integral = 0.5 \* (func(a) + func(b));

for (int i = 1; i < n; i++) {

double x = a + i \* h;

integral += func(x);

}

integral \*= h;

return integral;

}

double polynomial\_func\_1(double x, int degree, double coefficients[]) {

double result = 0.0;

for (int i = 0; i <= degree; i++) {

result += coefficients[i] \* pow(x, i);

}

return result;

}

// Trapezoidal rule-1 function for numerical integration

double trapezoidal\_rule\_1(double (\*func)(double, int, double[]), double a, double b, int degree, double coeff[], int n) {

double h = (b - a) / n;

double integral = 0.5 \* (func(a, degree, coeff) + func(b, degree, coeff));

for (int i = 1; i < n; i++) {

double x = a + i \* h;

integral += func(x, degree, coeff);

}

integral \*= h;

return integral;

}

int circle(char a[])

{

int k=0;

char cir[]="circle";

char CIR[]="CIRCLE";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(cir + i)||(a+i)==(CIR + i))

{

k++;

}

}

if (k==strlen(cir))

{

return k;

}

else

{

return 0;

}

}

int square(char a[])

{

int k=0;

char squ[]="square";

char SQU[]="SQUARE";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(squ + i)||(a+i)==(SQU + i))

{

k++;

}

}

if (k==strlen(squ))

{

return k;

}

else

{

return 0;

}

}

int rectangle(char a[])

{

int k=0;

char rec[]="rectangle";

char REC[]="RECTANGLE";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(rec + i)||(a+i)==(REC + i))

{

k++;

}

}

if (k==strlen(REC))

{

return k;

}

else

{

return 0;

}

}

int hexagon(char a[])

{

int k=0;

char tra[]="hexagon";

char TRA[]="HEXAGON";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(tra + i)||(a+i)==(TRA + i))

{

k++;

}

}

if (k==strlen(tra))

{

return k;

}

else

{

return 0;

}

}

int triangle(char a[])

{

int k=0;

char tri[]="triangle";

char TRI[]="TRIANGLE";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(tri + i)||(a+i)==(TRI + i))

{

k++;

}

}

if (k==strlen(tri))

{

return k;

}

else

{

return 0;

}

}

int pentagon(char a[])

{

int k=0;

char tri[]="pentagon";

char TRI[]="PENTAGON";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(tri + i)||(a+i)==(TRI + i))

{

k++;

}

}

if (k==strlen(tri))

{

return k;

}

else

{

return 0;

}

}

int heptagon(char a[])

{

int k=0;

char tri[]="heptagon";

char TRI[]="HEPTAGON";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(tri + i)||(a+i)==(TRI + i))

{

k++;

}

}

if (k==strlen(tri))

{

return k;

}

else

{

return 0;

}

}

int nanogon(char a[])

{

int k=0;

char tri[]="nanogon";

char TRI[]="NANOGON";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(tri + i)||(a+i)==(TRI + i))

{

k++;

}

}

if (k==strlen(tri))

{

return k;

}

else

{

return 0;

}

}

int decagon(char a[])

{

int k=0;

char tri[]="decagon";

char TRI[]="DECAGON";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(tri + i)||(a+i)==(TRI + i))

{

k++;

}

}

if (k==strlen(tri))

{

return k;

}

else

{

return 0;

}

}

int trapezium(char a[])

{

int k=0;

char tri[]="trapezium";

char TRI[]="TRAPEZIUM";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(tri + i)||(a+i)==(TRI + i))

{

k++;

}

}

if (k==strlen(tri))

{

return k;

}

else

{

return 0;

}

}

int octagon(char a[])

{

int k=0;

char tri[]="octagon";

char TRI[]="OCTAGON";

for (int i = 0; i < strlen(a); i++)

{

if ((a+i)==(tri + i)||(a+i)==(TRI + i))

{

k++;

}

}

if (k==strlen(tri))

{

return k;

}

else

{

return 0;

}

}