

Sumário

1. Introdução:	2
2. Implementação:	2
3. Testes	2
4. Conclusão	3
Referências	3
Anexos	3
ponto.h	3
ponto.c	4
main.c	6

1. Introdução:

Esse código tem como objetivo lê as coordenadas de um triângulo poligonal, logo em seguida calcular a sua área e mostrá-la no terminal ou no prompt de comando, as coordenadas então em um arquivo de texto (.txt), então foi utilizado a manipulação de arquivos para poder lê-las. Além disso foi implementado as operações que manipulam TADs em um arquivo ponto.c.

GitHub:

https://github.com/Diogo-Fr/Estrutura_de_Dados.git

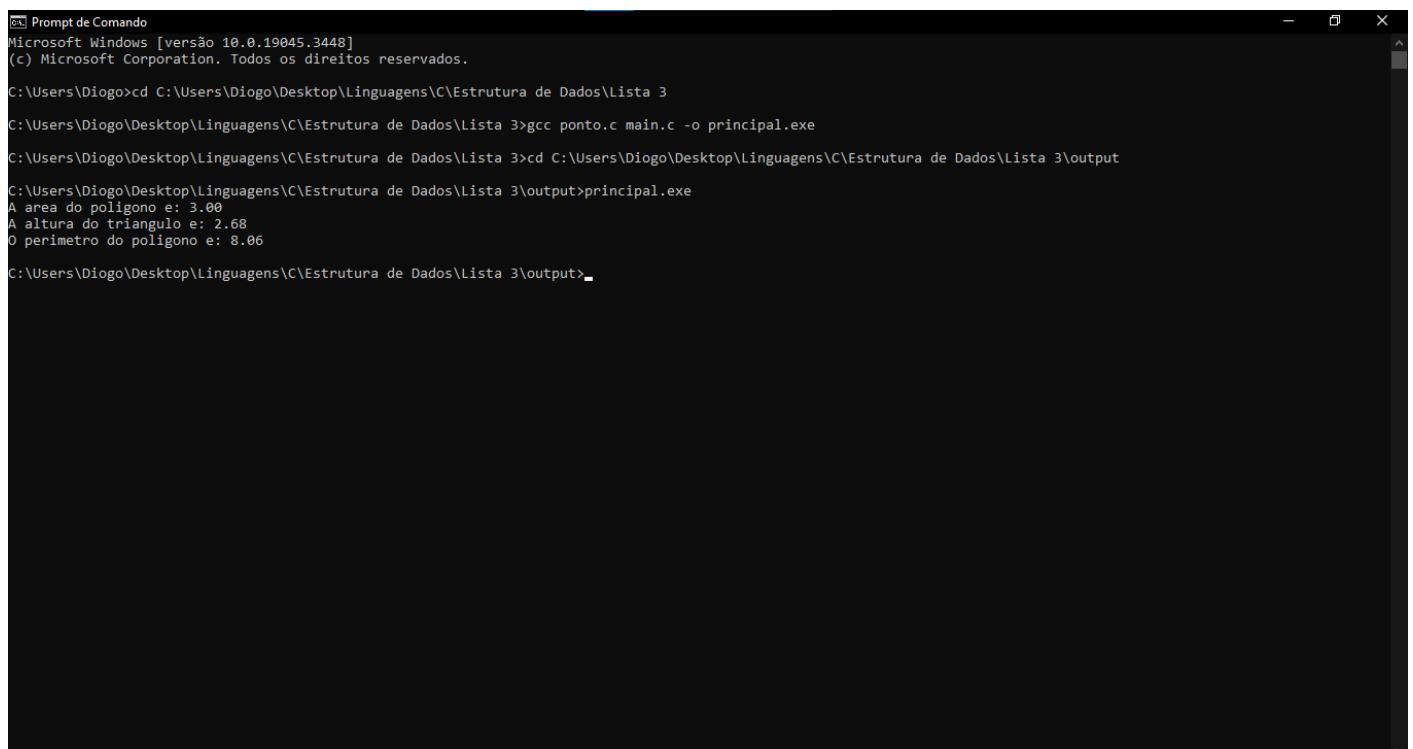
2. Implementação:

Ponto.h: É um arquivo de cabeçalho onde declara duas variáveis do tipo float X e Y onde estão armazenadas coordenadas de um ponto cartesiano, além disso, ele declara três funções, calcular a área, o perímetro e a altura.

Ponto.c: É onde fica os cálculos das funções que foram declaradas no ponto.h. No cálculo da área ele está fazendo a soma da área dos triângulos formados pelos pontos, no cálculo da altura está sendo utilizado o teorema de Heron ($A = b \cdot h / 2$) e por fim no cálculo do perímetro está somando as distâncias entre os pontos adjacentes.

Main.c: O código começa abrindo o arquivo poligono.txt para leitura e lê os dados das coordenadas nele e armazena nas variáveis X e Y. utilizando essas variáveis, ele chama as funções declaradas nos arquivos ponto.c e ponto.h e em seguida ele mostra os resultados na tela.

3. Testes



```
Prompt de Comando
Microsoft Windows [versão 10.0.19045.3448]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Diogo>cd C:\Users\Diogo\Desktop\Linguagens\C\Estrutura de Dados\Lista 3

C:\Users\Diogo\Desktop\Linguagens\C\Estrutura de Dados\Lista 3>gcc ponto.c main.c -o principal.exe

C:\Users\Diogo\Desktop\Linguagens\C\Estrutura de Dados\Lista 3>cd C:\Users\Diogo\Desktop\Linguagens\C\Estrutura de Dados\Lista 3\output

C:\Users\Diogo\Desktop\Linguagens\C\Estrutura de Dados\Lista 3\output>principal.exe
A area do poligono e: 3.00
A altura do triangulo e: 2.68
O perimetro do poligono e: 8.06

C:\Users\Diogo\Desktop\Linguagens\C\Estrutura de Dados\Lista 3\output>
```

4. Conclusão

O código em si não foi de extrema dificuldade, o maior problema foi compreender como fazê-lo (não estava presente na aula de explicação do tema) e executar o código, depois de pesquisar um pouco, consegui ter uma compreensão melhor das TADs e como desenvolvê-la, depois disso só foi executar alguns dos meus conhecimentos para realizar o que estava sendo pedido neste trabalho.

Referências

Sites virtuais

Gaspar, Wagner. Como criar arquivos de cabeçalho .h na linguagem C?

- <https://wagnergaspar.com/como-criar-arquivos-de-cabecalho-h-na-linguagem-c/>

Vídeos

https://www.youtube.com/watch?v=nX_3zUmNFFs

Anexos

ponto.h

```
#ifndef PONTO_H
```

```
#define PONTO_H
```

```
typedef struct {
```

```
    float x;
```

```
    float y;
```

```
} Ponto;
```

```
float calcularAreaPoligono(Ponto* pontos, int numPontos);
```

```
float calcularAlturaTriangulo(Ponto* pontos, int numPontos);
```

```
float calcularPerimetro(Ponto* pontos, int numPontos);
```

```
#endif
```

ponto.c

```
#include<math.h>

#include "ponto.h"

float calcularAreaPoligono(Ponto* pontos, int numPontos) {

    float area = 0.0;

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

        int j = (i + 1) % numPontos;

        area += (pontos[i].x * pontos[j].y) - (pontos[j].x * pontos[i].y);

    }

    area = 0.5 * area;

    if (area < 0) {

        area = -area; // A área não pode ser negativa.

    }

    return area;

}

float calcularAlturaTriangulo(Ponto* pontos, int numPontos) {

    if (numPontos != 3) {

        return 0.0;

    }

    // Sqrt é para calcular a raiz quadrada e pow para calcular a potência.
```

```
float a = sqrt(pow(pontos[1].x - pontos[2].x, 2) + pow(pontos[1].y - pontos[2].y, 2));  
float b = sqrt(pow(pontos[0].x - pontos[2].x, 2) + pow(pontos[0].y - pontos[2].y, 2));  
float c = sqrt(pow(pontos[0].x - pontos[1].x, 2) + pow(pontos[0].y - pontos[1].y, 2));  
  
float s = (a + b + c) / 2.0;  
float area = sqrt(s * (s - a) * (s - b) * (s - c));  
  
if (b == 0) {  
    return 0.0;  
}  
  
return (2.0 * area) / b;  
}  
  
float calcularPerimetro(Ponto* pontos, int numPontos) {  
    float perimetro = 0.0;  
  
    for (int i = 0; i < numPontos; i++) {  
        int j = (i + 1) % numPontos;  
        float distancia = sqrt(pow(pontos[i].x - pontos[j].x, 2) + pow(pontos[i].y - pontos[j].y, 2));  
        perimetro += distancia;  
    }  
  
    return perimetro;  
}
```

main.c

```
#include<stdio.h>

#include<stdlib.h>

#include "ponto.h"

int main() {

    FILE* arquivo;

    int numPontos;

    arquivo = fopen("poligono.txt", "r");

    if (arquivo == NULL) {

        printf("Erro ao abrir o arquivo.\n");

        exit(1);

    }

    fscanf(arquivo, "%d", &numPontos);

    // Malloc é usado para alocar uma memória de uma maneira dinâmica.

    Ponto* pontos = (Ponto*)malloc(numPontos * sizeof(Ponto));

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

        fscanf(arquivo, "%f %f", &pontos[i].x, &pontos[i].y);

    }

    fclose(arquivo);
```

```
float area = calcularAreaPoligono(pontos, numPontos);

printf("A area do poligono e: %.2f\n", area);


if (numPontos == 3) {

    float altura = calcularAlturaTriangulo(pontos, numPontos);

    printf("A altura do triangulo e: %.2f\n", altura);

}


float perimetro = calcularPerimetro(pontos, numPontos);

printf("O perimetro do poligono e: %.2f\n", perimetro);


return 0;

}
```