Sumário

[1. Introdução: 2](#_Toc146445554)

[2. Implementação: 2](#_Toc146445555)

[3. Testes 2](#_Toc146445556)

[4. Conclusão 2](#_Toc146445557)

[Referências 3](#_Toc146445558)

[Anexos 3](#_Toc146445559)

[Main.c 3](#_Toc146445560)

## Introdução:

O código em C tem como objetivo calcular a área de uma região poligonal com base nas coordenadas dos vértices fornecidas em um arquivo de entrada. Ele utiliza a fórmula de Shoelace para realizar esse cálculo e imprime o resultado no formato correto, com números de ponto flutuante, no console. Esse código pode ser útil em diversas aplicações envolvendo geometria computacional e é independente do local (locale) para formatação de números.

#### GitHub:

https://github.com/GuilFranca/estrutura\_dados/tree/main/TP03

## Implementação:

O código em C calcula a área de um polígono com base nas coordenadas de seus vértices lidos de um arquivo de entrada chamado "vertices.txt". Ele utiliza a fórmula de Shoelace para efetuar o cálculo e imprime o resultado no formato correto de ponto flutuante no console. O código é compilado em um ambiente C padrão e executado após a compilação. Ele aloca memória dinamicamente para armazenar as coordenadas dos vértices, realiza o cálculo da área e libera a memória antes de encerrar. O código fornece uma solução prática para o cálculo de áreas de polígonos a partir de dados em arquivo.

## Testes

Houve um problemas com a leitura do arquivo, por conta do nome do mesmo estar errado.

A screenshot of a computer

Description automatically generated

## Conclusão

Neste trabalho, desenvolvemos um programa Java que lê um número inteiro positivo par N e imprime todos os números pares de 0 até N em ordem crescente. O programa atende ao seu objetivo, oferecendo uma lista de números pares após validar a entrada do usuário. Possíveis melhorias incluem adicionar uma interface gráfica, aprimorar o tratamento de erros, otimizar o desempenho para números grandes, permitir ordenação personalizada e suporte a números ímpares.

## Referências

<https://ava.catolica.edu.br/d2l/lms/dropbox/user/folder_submit_files.d2l?ou=59528&db=68652>

<https://www.youtube.com/watch?v=y_euDUgoND8>

## Anexos

Vertices.txt

Main.c

### Main.c

#include <stdio.h>

#include <stdlib.h>

typedef struct {

float x;

float y;

} Point;

int main() {

FILE \*file;

int numVertices, i;

float area = 0.0;

// Abre o arquivo para leitura

file = fopen("vertices.txt", "r");

if (file == NULL) {

perror("Erro ao abrir o arquivo");

return 1;

}

// Lê o número de vértices

fscanf(file, "%d", &numVertices);

// Aloca espaço para armazenar os vértices

Point \*vertices = (Point \*)malloc(numVertices \* sizeof(Point));

if (vertices == NULL) {

perror("Erro ao alocar memória");

fclose(file);

return 1;

}

// Lê as coordenadas dos vértices

for (i = 0; i < numVertices; i++) {

fscanf(file, "%f %f", &vertices[i].x, &vertices[i].y);

}

// Fecha o arquivo

fclose(file);

// Calcula a área do polígono usando a fórmula de Shoelace

for (i = 0; i < numVertices - 1; i++) {

area += (vertices[i].x \* vertices[i+1].y - vertices[i+1].x \* vertices[i].y);

}

area += (vertices[numVertices-1].x \* vertices[0].y - vertices[0].x \* vertices[numVertices-1].y);

area = fabsf(area) / 2.0;

// Exibe a área do polígono

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

// Libera a memória alocada

free(vertices);

return 0;

}