Sumário

[1. Introdução: 2](#_Toc144310519)

[2. Implementação: 2](#_Toc144310520)

[3. Testes 2](#_Toc144310521)

[4. Conclusão 3](#_Toc144310522)

[Referências 3](#_Toc144310523)

[Anexos 4](#_Toc144310524)

[principal.c 4](#_Toc144310525)

## Introdução:

O presente projeto visa solucionar um problema em que muitos professores têm ao longo de suas carreiras profissionais: A demora para lançar o resultado final de cada aluno de acordo com suas notas. Dessa forma, visando facilitar a vida dos professores, foi feito um programa utilizando a linguagem de programação C. Esse aplicativo tem como objetivo ler um arquivo CSV lançado pelo usuário (normalmente um professor), em que conterá as informações necessárias para nosso programa funcionar: Nome, telefone, curso, notas 1 e 2. Após a entrada, será processado e o programa irá gerar um novo arquivo chamado “SituacaoFinal.csv”.

#### GitHub:

https://github.com/marioprneto/Trabalho-de-Estrutura-de-dados

## Implementação:

Em primeira instância foi criada a variável composta do tipo struct para o aluno. Após isso, foi criado duas variáveis do tipo FILE para ler e escrever os arquivos, uma do tipo char com capacidade de até 1024 caracteres (valor máximo para ser lido por linha) e uma variável do tipo aluno. Após isso, foi recebido os valores do tipo FILE por meio do méotod fopen(), e caso eles forem nulos, exibirá uma mensagem de erro identificando este problema. Após isso, caso o programa não seja encerrado, ele irá executar um laço de repetição que só irá terminar quando a última linha for lida. Desse modo, ele irá percorrer cada linha e fazer a soma da média, optei por fazer dentro do laço principal mesmo pois é algo simples de entender, uma verificação que retorna uma situação, que é definida por variável, em que é “APROVADO” ou “REPROVADO”. Após isso, o método fprintf() serve para gravar no novo arquivo gerado o resultado com os parâmetros do nome, a média e o resultado final. Por fim, só fechei os arquivos com o método fclose() e encerrei a execução do meu programa.

## Testes

Para meus testes, eu optei por gerar vários arquivos: Um que iria conter somente dados de alunos que foram REPROVADOS, outro que iria conter dados somente de alunos APROVADOS, e por fim, mesclado. Confesso que o que mais retornou resultado foi o mesclado, pois pude ver que o aplicativo estava funcionando corretamente, dessa forma, segue em anexo o resultado dos meus testes ao ser executado pelo Code blocks, IDE de minha preferência:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

## Conclusão

Com base no apresentado, é possível tirar por conclusão que está não é, de longe, a melhor forma de fazer a resolução do problema. Porém, é uma forma bem eficiente de se resolver a questão proposta, utilizando a lógica. Mas, esse é um algoritmo que por ser simples, permite implementações constantes, tais como separar as funções específicas da parte principal do algoritmo.

## Referências

<https://chat.openai.com/>?

https://stackoverflow.com/questions/8115944/writing-user-input-to-a-file-in-c-programming

## Anexos

### principal.c

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente