

Trabalho Final Algoritmos

Professores: Guilherme Dal Bianco e Priscila Delabetha

Apresentação:

Portal do Professor

Você foi contratado para desenvolver um sistema para os professores da UFFS controlarem as notas de seus alunos, e gerenciarem seus diários.

O sistema deve possuir as seguintes funções:

Funções principais:

- Cadastro de Turma
- Cadastros de Alunos
- Inserção de notas (NP1 e NP2)
- Calcular Médias Finais
- Encerrar Diário

Funções de exibição:

- Listar alunos da turma
- Exibir dados da turma
- Exibir notas parciais
- Exibir notas finais

Detalhes das funções principais:

Cadastros de Turma:

Permite o usuário cadastrar turmas com as informações:

- Código da turma (aleatório ou gerado sequencialmente)
- Componente curricular ministrado
- Fase
- Horário das aulas

Cadastrar Alunos:

Deve possibilitar ao usuário cadastrar as informações básicas de um aluno:

- Nome completo do aluno
- Número de matrícula
- CPF (deve ser validado)
- Turma a qual ele pertence

Inserção de notas (NP1 e NP2):

Essa função permite ao professor informar as notas parciais de NP1 ou NP2 de uma turma.

Calcular médias finais:

Essa função irá utilizar o peso informado pelo professor para calcular a média final de cada aluno da turma.

Encerrar diário:

A função encerrar diário deve verificar os alunos aprovados e reprovados, e exibir na tela as médias finais da turma toda, juntamente com a informação de aprovação ou reprovação dos alunos.

Utilização do programa

O sistema deve exibir um menu com as opções para que o usuário possa selecionar o que deseja fazer.

Recursos da Linguagem C obrigatórios no projeto:

Funções - as operações detalhadas neste trabalho devem ser feitas com funções da Linguagem C.

Vetores - estruturas de vetores precisam estar presentes no trabalho.

Structs – Utilize Structs para pelo menos uma das informações que precisam ser armazenadas.

Ponteiro - Pelo menos uma função no projeto precisa utilizar ponteiros.

Avaliação:

1. Compilação: 10%
2. Execução correta: 60%
 - Serão feitos vários testes.
 - Cada teste com um nível de dificuldade maior, onde o arquivo de saída do programa será comparado com um "gabarito". O aluno receberá nota máxima se ambos forem idênticos.
3. Estilo de programação: 10%
 - Código bem indentado, legível, comentado (sem excesso), bem estruturado, nomes de variáveis significativos, etc.
4. Documentação: 10%

5. Apresentação do andamento com a estrutura do código pronta e a lógica para resolver o problema (função já implementada): 10%