UESC – Universidade Estadual de Santa Cruz – 15/04/2025

Ciência da Computação – DEC000840-Programação II – **PROVA 01**

Sistema de Cadastro e Ordenação de Alunos

Objetivo:

Desenvolver um programa em C que gerencia dados de alunos, permitindo seu cadastro e ordenação por Coeficiente de Rendimento (CR) utilizando estruturas e algoritmos de ordenação.

Descrição do Programa:
O sistema deve:
Cadastrar Alunos

- Ler dados de 3 alunos (nome, idade, sexo e CR)
- Armazenar os registros em um vetor de estruturas

Exibir Dados

- Mostrar todos os alunos cadastrados na ordem original
- Mostrar os alunos ordenados por CR (ordem crescente)

Implementar Funções

- le_Aluno(): função sem parâmetros que retorna uma estrutura preenchida
- prn_Aluno(): recebe uma estrutura e exibe seus dados formatados
- ordena_Alunos(): ordena o vetor de alunos usando algoritmo Bubble Sort

Requisitos Técnicos:

- Usar estruturas (struct) para organizar os dados
- Implementar passagem por valor para exibição dos dados
- Implementar passagem por referência para ordenação
- Utilizar constantes para definir o número de alunos
- Formatar a saída para melhor legibilidade

Exemplo de Saída:

Cadastro do Aluno 1:
Digite o nome: Ana Costa
Digite a idade: 21
Digite o sexo: Feminino
Digite o CR: 8.2

[... outros cadastros ...]

=== ANTES DA ORDENAÇÃO ===
--Dados do Aluno--
Nome: Ana Costa
Idade: 21
Sexo: Feminino
CR: 8.20

[... outros alunos ...]

=== DEPOIS DA ORDENAÇÃO ===
--Dados do Aluno--
Nome: João Silva
Idade: 20
Sexo: Masculino
CR: 7.80

[... alunos ordenados ...]

Critérios de Avaliação:

- Correta implementação das estruturas
- Funcionamento do algoritmo de ordenação
- Manipulação adequada do vetor de alunos
- Formatação e organização do código
- Tratamento de entrada de dados

Observações:

- O programa deve usar obrigatoriamente estruturas e funções
- Comentar o código explicando a lógica de ordenação
- Validar as entradas (ex: idade positiva, CR entre 0 e 10)

- Cada função será pontuada em até 2,5 pontos (função principal e funções secundárias)
- Se o programa não compilar e executar, cada função será pontuada em até 1,25 pontos (função principal e funções secundárias)

UESC – Universidade Estadual de Santa Cruz – 15/04/2025

Ciência da Computação – DEC000840-Programação II – **PROVA 02**

Sistema de Gerenciamento de Alunos com Ordenação por CR

Objetivo:
Desenvolver um programa em C que gerencia dados de alunos, permitindo sua ordenação por Coeficiente de Rendimento (CR) sem modificar a estrutura original dos dados, utilizando vetores de ponteiros e aritmética de ponteiros.

Descrição:
O programa deve:
1. Coletar Dados
- Ler informações de 3 alunos (nome, idade, sexo e CR) e armazená-las em um vetor de estruturas Taluno
- Criar um vetor de ponteiros que aponta para esses dados

2. Exibir Dados
- Mostrar os alunos na ordem original
- Mostrar os alunos ordenados por CR (crescente)

3. Implementar
- Função le_Aluno() para leitura dos dados (sem parâmetros, retornando a struct preenchida)
- Função prn_Aluno() para exibição formatada dos dados
- Função ordena() que ordena os ponteiros por CR (usando aritmética de ponteiros)

Requisitos Técnicos:
- Usar aritmética de ponteiros em todas as operações com vetores
- Manter os dados originais intactos durante a ordenação
- Formatar a saída como tabela alinhada
- Prever/controlar estouro de buffer

Exemplo de Saída:

```
=== DADOS ORIGINAIS ===
Nome: Ana Costa         | Idade: 21 | Sexo: Feminino   | CR: 8.20
Nome: João Silva        | Idade: 20 | Sexo: Masculino  | CR: 7.80
Nome: Maria Oliveira    | Idade: 22 | Sexo: Feminino   | CR: 9.10

=== ORDENADO POR CR ===
Nome: João Silva        | Idade: 20 | Sexo: Masculino  | CR: 7.80
Nome: Ana Costa         | Idade: 21 | Sexo: Feminino   | CR: 8.20
Nome: Maria Oliveira    | Idade: 22 | Sexo: Feminino   | CR: 9.10
```

Observações:
O código deve usar obrigatoriamente:
- *(vetor + i) // Notação de ponteiros
- (*(p+i))->CR // Acesso a campos via aritmética de ponteiros

- Comentar o código explicando cada operação com ponteiros

- Cada função será pontuada em até 2,5 pontos (função principal e funções secundárias)
- Se o programa não compilar e executar, cada função será pontuada em até 1,25 pontos (função principal e funções secundárias)

Dica: A ordenação deve modificar apenas o vetor de ponteiros, preservando a posição original dos dados no vetor de estruturas.