

Sistema de notas

Um sistema de notas é uma complexa atividade em um sistema educacional. O SIGAA é muito criticado pelos seus usuários, mas a natureza complexa e peculiaridades de diferentes instituições torna o seu desenvolvimento uma complexa atividade!

Uma das tarefas básicas desse sistema, é carregar os dados que estão salvos em strings e colocá-los dentro de structs para poder processar! Sua tarefa é implementar um código que contém a struct aluno com os seguintes campos:

Campo	Tipo
nota	um inteiro
nome	string de 80 caracteres

Além disso, defina no seu código a função preencherAluno:

```
struct aluno preencherAluno(char texto[240])
```

Essa função recebe uma string de texto, que contém até 240 caracteres com a nota do aluno e o nome do aluno **separados por vírgula**. Nesta função, você precisa copiar o conteúdo do argumento “texto” para uma struct aluno e, por último, retornar a struct preenchida.

Restrições

Não é permitido utilizar string.h, nem funções da stdio.h que fazem parser de strings. Mas é permitida a criação de outras funções auxiliares. **IMPORTANTE!** Defina a função com o mesmo nome, incluindo letras maiúsculas e minúsculas, mesma ordem de argumentos e mesmos nomes nos argumentos. Caso contrário irá ocorrer erro de compilação. O nome da struct e de seus campos devem ser idênticos a tabela acima (incluindo letras maiúsculas e minúsculas). Além disso, os campos strings devem conter exatamente 80 caracteres, nenhum a mais ou a menos.

Ao enviar a sua solução pro MOJ, envie somente o arquivo com a extensão C com a função exigida do enunciado, as estruturas, a inclusão das bibliotecas utilizadas e as funções auxiliares (caso existam). Não inclua neste arquivo a função main. Crie uma função main para testar seu código, mas remova-a (ou comente) antes de enviar a solução para o MOJ.

Entrada

A sua função não deve fazer nenhuma leitura da entrada padrão.

Saída

A sua função não deve imprimir nenhum dado de saída.

Exemplos

Exemplo 1

```
preencherAluno("75, Nome Sobrenome");
```

Sua função deve retornar a struct corretamente preenchida.

Exemplo 2

```
preencherAluno("100, Jose Maria Jesus");
```

Sua função deve retornar a struct corretamente preenchida.

Author: Daniel Sundfeld <daniel.sundfeld@unb.br>