

# SCC0503 Algoritmos e Estruturas de Dados 2

Disponibilizado em: 28-03-2023

Código Turma: YLK7

Submissões até: 15-04-2023

## Trabalho 1 - Processamento de Campos e Registros em Arquivos

### Descrição

Este trabalho consiste na leitura de uma entrada (*stdin*) contendo dados de registros de livros, escrita desses dados em um arquivo (modo binário) e leitura e impressão dos  $m$  últimos registros escritos no arquivo. Vamos imaginar que você está desenvolvendo um gerenciador de livros para a biblioteca pessoal do seu professor preferido!

### Estrutura do Registro

O registro que descreve um livro é composto pelos seguintes campos:

- id - int (vamos considerar um id simplificado formado por um número pequeno)
- titulo - string (tamanho variável)
- autor - string (tamanho variável)

### Entrada

A entrada será constituída pelos registros que você deve ler e escrever no arquivo e pelo número que indicará quais registros devem ser impressos.

- 1º Entrada[int] - valor  $n$ , referente ao número de registros que serão fornecidos. ( $n > 0$ )
- 2º Entrada[n registros] -
  - id[int]
  - titulo[string]
  - autor[string]
- 3º Entrada[int] - valor  $m$ , define quais registros devem ser exibidos, por exemplo: para  $m = 3$ , imprima os 3 últimos registros do arquivo. ( $m \leq n$ )

Exemplo:

5

10

The Hunger Games

Suzanne Collins

5

The Diary of a Young Girl

Anne Frank

3

Little Women

Louisa May Alcott

8

The Fault in Our Stars

John Green

12

Anne of Green Gables

L.M. Montgomery

3

## Escrita em Arquivo

Os dados devem ser escritos no arquivo usando uma organização híbrida, o *id* é de tamanho fixo, o *título* é de tamanho variável, deve haver um símbolo para indicar o seu final, o *autor* também é de tamanho variável, deve haver um inteiro antes da string servindo como indicador de comprimento e por último, os registros devem ser separados por um símbolo.

Exemplo:

```
10The Hunger Games|15Suzanne Collins-1
5The Diary of a Young Girl|10Anne Frank-1
...
```

Vale notar que o exemplo acima mostra os números como caracteres para facilitar o entendimento. Como o arquivo será escrito em modo binário (com a função *fwrite*), somente ficarão legíveis os dados escritos como string (delimitadores, título e autor), os números vão aparecer como caracteres desconhecidos, pois estão escritos como a representação binária do número e não seu caractere correspondente. Dessa forma:

```
??The Hunger Games|???Suzanne Collins???The Diary of a Young
Girl|???Anne Frank?
```

→ int relativo ao id

→ int relativo ao comprimento da string de autor

→ char relativo ao delimitador de registro

Os delimitadores utilizados podem ser diferentes (cuidado para não escolher um símbolo que já é utilizado no domínio dos dados), o que importa é que a escrita do registro siga a estrutura proposta:

- **id (4 bytes [int])** - >
- **título (tamanho variável)** - >
- **delimitador da string de título (1 byte [char])** - >
- **indicador de comprimento da string de autor (4 bytes [int])** - >
- **autor (tamanho variável)** - >
- **delimitador de registro (1 byte [char])** - >

Sugestão de delimitadores:

- delimitador da string de título - > '|'.
- delimitador de registro - > -1 (valor armazenado em uma variável do tipo char, para ocupar 1 byte no arquivo, porém trata-se do número -1 e não dos caracteres "-1". A escolha é válida tendo em vista que o -1 não será usado como *id* nem como indicador de comprimento, o que causaria conflito).

## Saída

O programa deve imprimir no *stdout* os *m* últimos registros do arquivo e seus respectivos *byte offsets* (posição no arquivo), para o exemplo de entrada acima, a saída esperada é a seguinte (atente-se à formatação):

```
Id: 3
Titulo: Little Women
Autor: Louisa May Alcott
Byte offset: 86

Id: 8
Titulo: The Fault in Our Stars
Autor: John Green
Byte offset: 125

Id: 12
Titulo: Anne of Green Gables
Autor: L.M. Montgomery
Byte offset: 167
```

## Dicas

- Pode-se utilizar um buffer estático temporário para leitura de strings (por exemplo `char titulo[100]`), elas não passarão de 100 de comprimento.
- Se estiver tendo dificuldade para ler as strings do *stdin* em decorrência dos espaços, tente utilizar a seguinte função: `scanf("%[^(\n|\n)]", string);` ela lê uma string da entrada até encontrar um caractere de quebra de linha.