

# Trabalho de implementação

Um índice remissivo é uma lista que relaciona termo com suas respectivas localizações no interior de um livro ou de uma publicação. Em livros, o índice é ordenado alfabeticamente com base nos termos e cada termo, ou entrada do índice, é associado a um número de página. Segue um exemplo de índice remissivo:

```
Argumento, 23, 70, 166
Atribuição, 129
Classe abstrata, 191
Classe amiga, 207
Destruitor, 25, 26
Endereço de bytes, 122
Endereço de memória, 122
Função amiga, 184, 204
```

Sua tarefa é Implementar uma estrutura de dados dinâmica (sem limite de tamanho) capaz de armazenar um índice remissivo e que permita realização das operações definidas a seguir. Você deve implementar o índice usando arranjo ou lista ligada.

Operações que devem ser implementadas:

- `indice* criar_indice()` : cria e retorna um índice sem entradas.
- `unsigned int tamanho(indice *ind)` : retorna a quantidade de entradas de um índice.
- `entrada* localizar(indice *ind, char *termo_de_busca)` : busca um termo em um índice. Retorna a entrada correspondente ao termo de busca ou NULL caso o índice não contenha entrada referente ao termo de busca.
- `void inserir_entrada(indice *ind, char *termo, unsigned int *paginas, int n_paginas)` : cria e insere uma entrada em um índice.
- `void atualizar_entrada(indice *ind, char *termo, unsigned int *paginas, int n_paginas)` : atualiza uma entrada em um índice sobrescrevendo os dados relacionados às páginas de ocorrência pelos dados informados como parâmetros da função. A operação deve ignorar um termo fornecido sem entrada no índice.
- `void imprimir(indice *ind)` : imprime todas as entradas do índice em ordem alfabética crescente. Siga o formato apresentado no exemplo anterior de índice remissivo.

São disponibilizados os seguintes arquivos:

- `indice.h` : este arquivo contém os protótipos das funções que operam sobre o índice remissivo. Ele também define as estruturas utilizadas por estas funções. O próprio arquivo contém instruções em forma de comentários.
- `indice.c` : este arquivo deverá conter a sua implementação das funções definidas no arquivo `indice.h`. Você é livre para criar funções adicionais neste arquivo.

- `teste.c` : este arquivo contém uma bateria de testes que verificam a corretude das funções implementadas em `indice.h`.

Use o seguinte comando para compilar os arquivos:

```
gcc -Wall -ansi -o teste indice.c teste.c
```

Critérios de pontuação:

- Implementação de estrutura dinâmica.
- Quantidade de funções implementadas.
- Manipulação adequada de ponteiros e estruturas.
- Quantidade de testes bem-sucedidos (presentes no arquivo `teste.c`).