

Trabalho Prático – Organização e Recuperação da Informação

Prof. André R. Backes

Instruções

- **Número de integrantes por grupo 3 pessoas. Grupos com mais ou menos pessoas somente serão aceitos se não houver número suficiente de alunos para formar outro grupo.**
- **Atenção: alguns grupos poderão ser sorteados para apresentar o trabalho quando houver dúvidas quanto a sua implementação.**

Data de Entrega:

24/08/2023

Forma de Entrega:

O trabalho deverá ser entregue por e-mail. Deverá ser enviado apenas o código fonte do programa. No campo **assunto** do e-mail, colocar **TrabalhoORI**.

No arquivo **main** do programa, e no corpo do e-mail, deverá constar os nomes dos integrantes do grupo.

Escolha do Trabalho

Considere que

A = 1	E = 5	I = 9	M = 13	Q = 17	U = 21	Y = 25
B = 2	F = 6	J = 10	N = 14	R = 18	V = 22	Z = 26
C = 3	G = 7	K = 11	O = 15	S = 19	W = 23	
D = 4	H = 8	L = 12	P = 16	T = 20	X = 24	

Agora some o valor da primeira letra do primeiro nome de cada membro do grupo. O resto da divisão dessa soma por 2 indica a TAD a ser implementada.

Problema

O grupo deverá implementar a TAD correspondente (arquivos **.c** e **.h**) descrita abaixo, incluindo um arquivo **main.c** para testes.

- **TAD 0: árvore B (B tree)**
- **TAD 1: árvore B* (B* tree)**

O grupo deverá implementar a TAD da árvore correspondente para armazenar valores inteiros (não é preciso implementar a parte de manipulação de arquivos da estrutura). A TAD deverá seguir o seguinte protótipo no .h:

```
#define ordem 8 //número máximo de filhos
typedef struct NO* ArvB;
ArvB* arvB_cria();
void arvB_destroi(ArvB *raiz);
int arvB_insere(ArvB* raiz, int valor);
int arvB_remove(ArvB *raiz, int valor);
int arvB_busca(ArvB *raiz, int valor);
int arvB_qtd_nos(ArvB *raiz);
int arvB_qtd_chaves(ArvB *raiz);
```

Obs: a TAD pode conter outras funções auxiliares no .c

Instruções:

O trabalho será avaliado principalmente levando em consideração:

- 1) Realização das tarefas do trabalho.
- 2) Representação correta da entrada e saída dos dados.
- 3) Uso correto das variáveis e estruturas de dados.
- 4) Uso adequado dos conceitos aprendidos em sala (modularização, hash, árvores, etc, quando for o caso).
- 5) Boa indentação e uso de comentários no código. Evite utilizar comentários excessivamente.