

# Informações do aluno

---

- Nome: Gabriel Haruo Hanai Takeuchi
- NUSP: 13671636

## Como compilar

---

### Modo 1: usando um arquivo .txt com input

```
> gcc -Wall -ansi -pedantic -O2 main.c -o a.out  
> ./a.out < qualquertexto.txt
```

### Modo 2: digitando diretamente no teclado

```
> gcc -Wall -ansi -pedantic -O2 main.c -o a.out  
> ./a.out  
um texto  
qualquer.  
apenas um exemplo  
/*aperte Ctrl+d para dar o sinal EOF ao programa*/
```

## Estrutura do programa

---

O índice remissivo foi implementado com dois tipos diferentes de árvores binárias de busca.

1. Uma ABB para armazenar cada **palavra** que aparece no texto. Chamemos essa ABB apenas de *ABB*.

```
typedef struct celABB {  
    char *chave;  
    struct celABB *esq;  
    struct celABB *dir;  
    noIND *info;  
} noABB;
```

2. Para cada nó/palavra da ABB (1.), uma outra ABB que armazena as **linhas** e o **número de vezes** que a palavra ocorre no texto. Chamemos essa nova árvore de *índices* de um nó da ABB.

```
typedef struct celIND {  
    int linha;  
    int cont;
```

```
    struct celIND *esq;  
    struct celIND *dir;  
} noIND;
```

Observe que cada célula da ABB contém um ponteiro para uma célula de índices. As estruturas estão definidas no arquivo `abb.h`.

## Funções implementadas

---

Usaremos as funções já conhecidas de inserção em ABB e impressão em inordem. Entretanto, para cada função há duas versões: uma para aplicação na ABB e outra nos índices. As funções "devidamente comentadas" estão no arquivo `abb.c`.