

1 -

```
C/C++
bool Media(TipoRegistroEstrela *x, TipoApontadorEstrela *Ap, long * media){

    TipoApontadorEstrela Pag = *Ap;

    if(Pag == NULL) return false;

    if(Pag->Pt == Interna){

        int i = 1;

        while(i < Pag->UU.U0.ni && x->Chave > Pag->UU.U0.ri[i-1]) i++;

        if(x->Chave >= Pag->UU.U0.ri[i-1]) return Media(x,
&Pag->UU.U0.pi[i], media);

        else return Media(x, &Pag->UU.U0.pi[i-1], media);
    }
    else{
        //Verificando se o item existe na pagina
        int i;
        long somatorio = 0;
        bool existe = false;
        for(i = 0; i < Pag->UU.U1.ne; i++){
            somatorio += Pag->UU.U1.re[i].dado1;

            if(Pag->UU.U1.re[i].Chave == x->Chave)
                existe = true;
        }
        if(existe){
            *media = somatorio/i;
            return true;
        }
        return false;
    }
}
```

C/C++

```
int main(){
    TipoChave inserir [] = {11, 36, 53, 95, 8, 91, 81, 48, 25, 27, 31, 16,
21, 59, 63, 72};
    int valor [] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16};

    TipoApontadorEstrela Ap;
    inicializa_b_estrela(&Ap);

    for(int i = 0; i<16; i++){
        TipoRegistroEstrela reg;
        reg.Chave = inserir[i];
        reg.dado1 = valor[i];

        Insere_b_estrela(reg, &Ap);
    }

    exibirItensArvoreBEstrela(&Ap);

    long media;
    TipoRegistroEstrela pesq;
    pesq.Chave = 36;
    if(Media(&pesq, &Ap, &media))
        printf("Média: %ld\n", media);
    else
        printf("Chave não presente na árvore\n");

    return 0;
}
```

```
8 : 5
11 : 1
16 : 12
21 : 13
Chave interna: 25
25 : 9
27 : 10
31 : 11
Chave interna: 36
36 : 2
48 : 8
Chave interna: 53
53 : 3
59 : 14
63 : 15
72 : 16
Chave interna: 91
81 : 7
91 : 6
95 : 4
Média: 5
```

2 - Limite inferior e limite superior: Delimitam entre quais valores os itens da partição devem estar para saber se irão entrar no ou não no pivô.

Leitura inferior e superior: É a partir destes ponteiros que selecionamos o item que será lido de forma alternada.

Escrita inferior e superior: São os ponteiros que serão responsáveis para determinar onde os dados serão escritos no arquivo após saírem do pivô ou quando eles vão sair direto do arquivo e ir para outra posição.

i & j: Serão os delimitadores para onde as próximas partições do quicksort começarão (j até o fim) ou terminarão (começo até o i).

Pivô: É o item que estará ordenado ao terminar a partição, ficando armazenado na memória principal, sendo que seus valores poderão ser sobrescritos no arquivo.

3) 10 20 45 50 87 11 10 19 25 30

It 1 10 20 45 50 87 19

It 2 10 11 19 25 30 25 30

It 3 10 10 11 19 20 25 30 45 50 87

It 4