

```
// IED-001 (Prof. Dr. Silvio do Lago Pereira)
```

```
// -----  
// Exemplo 1  
// -----
```

```
typedef int Chave;  
typedef char Valor[22];  
typedef struct map {  
    Chave chave;  
    Valor valor;  
    struct map *prox;  
} *Map;
```

```
// -----  
// Exemplo 2  
// -----
```

```
Map no_map(Chave c, Valor v, Map p) {  
    Map n = malloc(sizeof(struct map));  
    n->chave = c;  
    strcpy(n->valor,v);  
    n->prox = p;  
    return n;  
}
```

```
// -----  
// Exemplo 3  
// -----
```

```
void insm(Chave c, Valor v, Map *M) {  
    while( *M && c>(*M)->chave )  
        M = &(*M)->prox;  
    if( *M && c==(*M)->chave )  
        strcpy((*M)->valor,v);  
    else *M = no_map(c,v,*M);  
}
```

```
// -----  
// Exemplo 4  
// -----
```

```
void exibem(Map M) {  
    printf("[");  
    while( M ) {  
        printf("(%d,%s)",M->chave,M->valor);  
        if( M->prox ) printf(",");  
        M = M->prox;  
    }  
    printf("]\n");  
}
```

```
// -----  
// Exercício 1  
// -----
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
...  
int main(void) {  
    Map I = NULL;  
    insm(36,"Leo",&I);  
    insm(15,"Ivo",&I);  
}
```

```

    insm(42,"Eva",&I);
    insm(29,"Ana",&I);
    exibem(I);
    insm(29,"Bia",&I);
    exibem(I);
    return 0;
}

```

```

// -----
// Exemplo 5
// -----

```

```

void remm(Chave c, Map *M) {
    while( *M && c>(*M)->chave )
        M = &(*M)->prox;
    if( *M==NULL || c!=(*M)->chave ) return;
    Map n = *M;
    *M = n->prox;
    free(n);
}

```

```

// -----
// Exercício 3
// -----

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

...

```

```

int main(void) {
    Map I = NULL;
    insm(36,"Leo",&I); insm(15,"Ivo",&I);
    insm(42,"Eva",&I); insm(29,"Ana",&I);
    exibem(I);
    remm(29,&I);
    exibem(I);
    remm(42,&I);
    exibem(I);
    return 0;
}

```

```

// -----
// Exemplo 6
// -----

```

```

int pertm(Chave c, Valor v, Map M) {
    while( M && c>M->chave )
        M = M->prox;
    if( M && c==M->chave )
        strcpy(v,M->valor);
    return (M && c==M->chave);
}

```

```

// -----
// Exercício 5
// -----

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

...

```

```

int main(void) {

```

```
Valor w;  
Map I = NULL;  
insm(36,"Leo",&I); insm(15,"Ivo",&I); insm(42,"Eva",&I);  
exibem(I);  
if( pertm(42,w,I) ) printf("Valor da chave 29: %s\n",w);  
else puts("Chave 29 inexistente!");  
if( pertm(10,w,I) ) printf("Valor: %s\n",w);  
else puts("Chave 10 inexistente!");  
return 0;  
}
```