

```

#include <stdio.h>
#define MAX 50

typedef int telem;
typedef struct {
    telem v[MAX];
    int n;
} tlista;

/* 1) Criar uma lista vazia*/

void criar (tlista *L) {
    L->n = 0;
}

/* 2) Verificar se uma lista está vazia*/

int vazia (tlista L) {
    return (L.n == 0);
}

/*3) Verificar se uma lista está cheia*/

int cheia (tlista L) {
    return (L.n == MAX);
}

/*4) Obter o tamanho de uma lista*/

int tamanho (tlista L) {
    return (L.n);
}

/*5) Obter o i-ésimo elemento de uma lista*/

int elemento (tlista L, int pos, telem *dado) {
    /* O parâmetro dado irá receber o elemento encontrado */
    /* Retorna 0 se a posição for inválida. Caso contrário, retorna 1 */

    if ( (pos > L.n) || (pos <= 0) ) return (0);
    *dado = L.v[pos-1];
    return (1);
}

/* 6) Pesquisar um dado elemento, retornando a sua posição */

int posicao (tlista L, telem dado) {
    /* Retorna a posição do elemento ou 0 caso não seja encontrado */

    int i;
    for (i=1; i<=L.n; i++)
        if (L.v[i-1] == dado)
            return (i);
    return (0);
}

/* 7) Inserção de um elemento em uma determinada posição
Requer o deslocamento à direita dos elementos v(i+1)...v(n) */

int inserir (tlista *L, int pos, telem dado) {
    /* Retorna 0 se a posição for inválida ou se a lista estiver cheia */

```

```

/* Caso contrário, retorna 1 */

int i;
if ( (L->n == MAX) || (pos > L->n + 1) ) return (0);
for (i=L->n; i>=pos; i--)
    L->v[i] = L->v[i-1];
L->v[i] = dado;
(L->n)++;
return (1);
}

/*8) Remoção do elemento de uma determinada posição
Requer o deslocamento à esquerda dos elementos v(p+1)...v(n)*/

int remover (tlista *L, int pos, telem *dado) {
/* O parâmetro dado irá receber o elemento encontrado */
/* Retorna 0 se a posição for inválida. Caso contrário, retorna 1 */
int i;
if ( (pos > L->n) || (pos <= 0) ) return (0);
*dado = L->v[pos-1];
for (i=pos; i<=(L->n)-1; i++)
    L->v[i-1] = L->v[i];
(L->n)--;
return (1);
}

void exibe (tlista L) {
    telem dado;
    int cont;

    printf("\n");
    for (cont=0; cont <= (L.n-1); cont++)
        printf("Elemento[%d] = %d\n", cont, L.v[cont]);
    printf("\n");
}

int main(int argc, char *argv[])
{
    tlista L1;
    int i;
    telem dado;

    criar(&L1);
    printf("\n Lista criada - Tamanho da lista    %d\n", tamanho(L1));
    inserir(&L1, 1, 20);
    inserir(&L1, 2, 13);
    inserir(&L1, 3, 14);
    printf("\n Insercao de tres elementos - Tamanho da lista    %d\n",
tamanho(L1));
    exibe(L1);
    inserir(&L1,2,15);
    printf("\n Insercao de um elemento - Tamanho da lista    %d\n",
tamanho(L1));
    exibe(L1);

    remover(&L1,4,&dado);
    printf("\n Remocao de um elemento - Tamanho da lista    %d\n",
tamanho(L1));
}

```

```
    exhibe(L1);  
    getchar();  
    return 0;  
}
```