

Listas

Listas Estáticas Sequenciais

Inserção: desloca para direita

Remoção: desloca para esquerda

(Fonte: Material adaptado dos Slides do prof. Monael.)

LES.c

```
int main () {
    int i, x, q, v[30000], quantidade = 0;
    scanf ("%d", &q);
    for (i = 0; i < q; i++) {
        char op;
        // note o espaco em branco antes do caractere lido
        scanf (" %c %d", &op, &x);
        switch (op) {
            case 'I': inserir (v, &quantidade, x);
                       break;
            case 'E': remover (v, &quantidade, x);
                       break;
        }
    }
    imprime (v, quantidade);
    return 0;
}
```

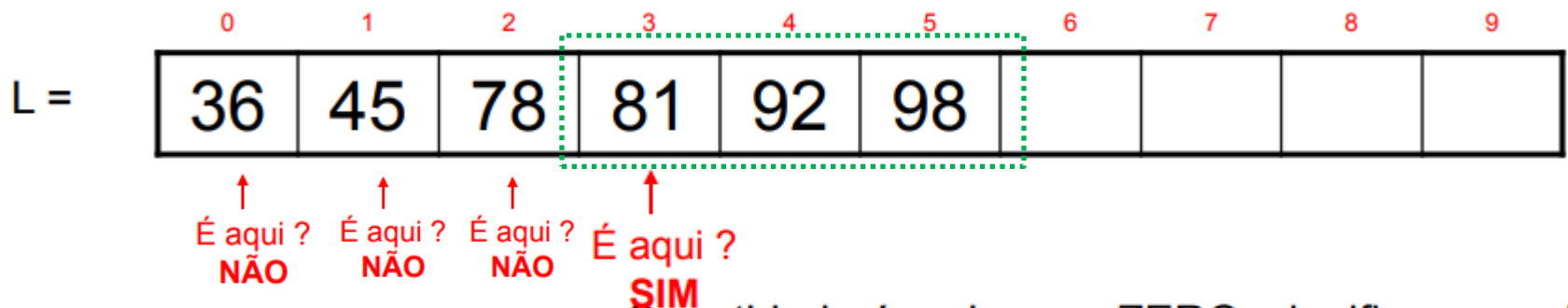
LES.c

```
// Recebe um vetor e a quantidade de elementos.  
// Imprime o conteudo do vetor, um elemento por linha.  
void imprime (int *v, int n) {  
    int i;  
    for (i = 0; i < n; i++) {  
        printf ("%d\n", v[i]);  
    }  
}
```

LES: Listas Estáticas Sequenciais

- **Inserção:** desloca para direita

Quantidade: 6



Quantidade é maior que ZERO, significa que a há itens na lista.

Portanto:

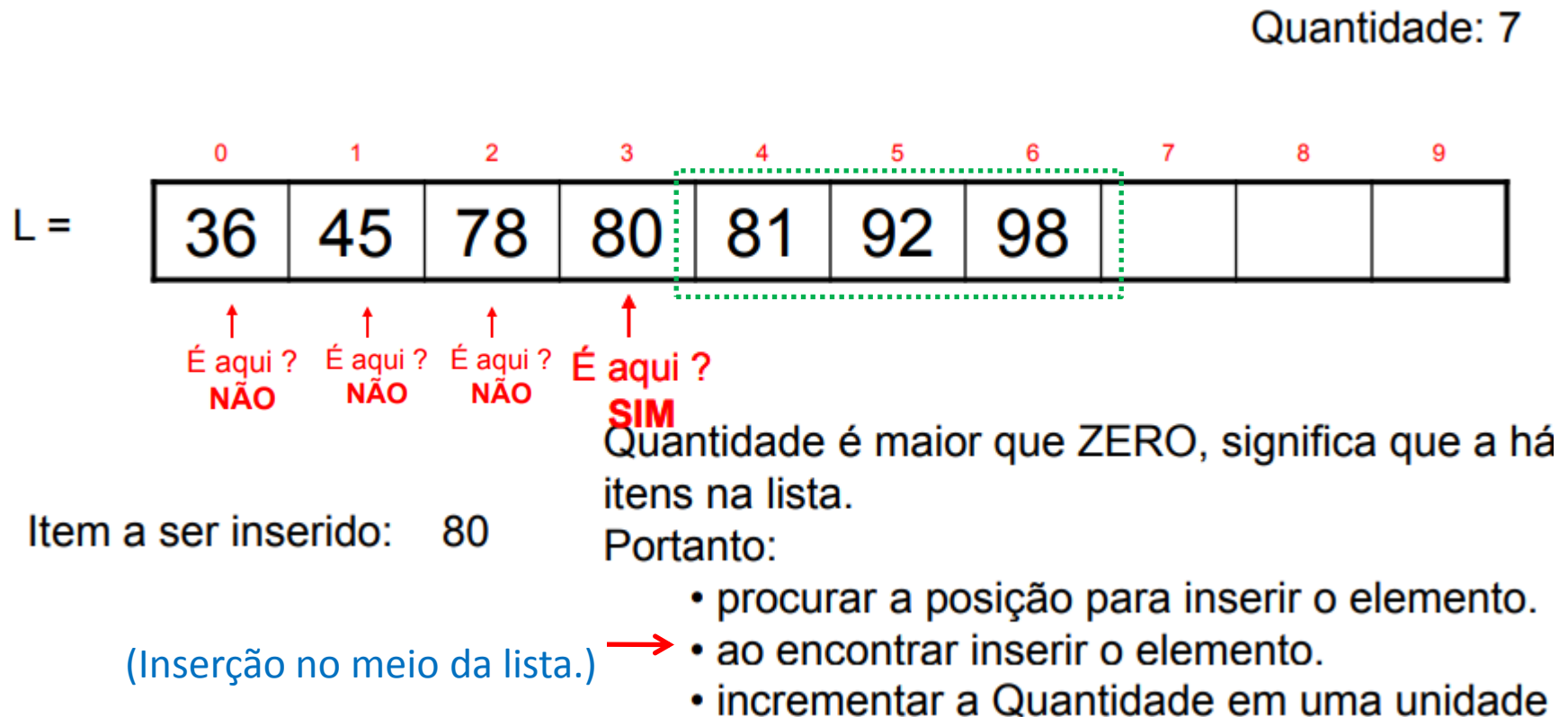
Item a ser inserido: 80

(Inserção no meio da lista.)

- procurar a posição para inserir o elemento.
- ao encontrar inserir o elemento.
- incrementar a Quantidade em uma unidade

LES: Listas Estáticas Sequenciais

- **Inserção:** desloca para direita



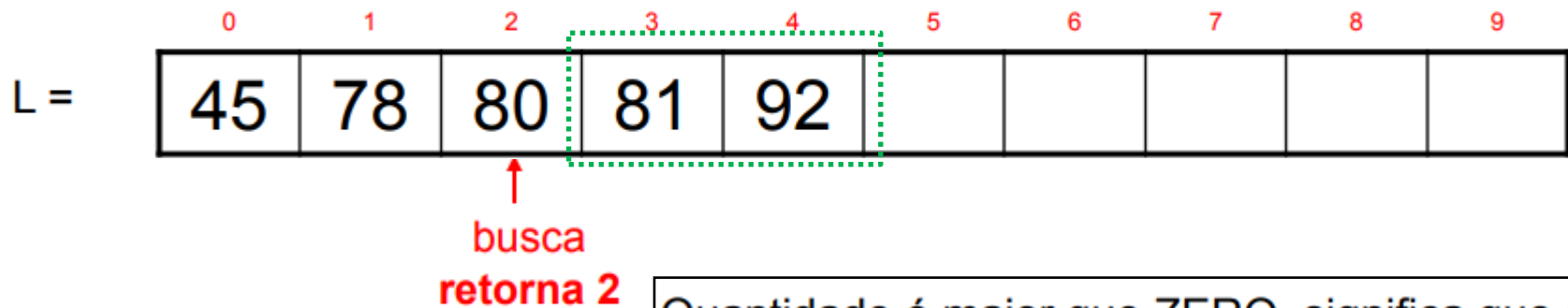
Desloca para direita

```
// Recebe um vetor, a quantidade de elementos (referencia)
// e o novo elemento a ser inserido.
// Procura a posicao de insercao i.
// Desloca os elementos para a direita caso necessário.
void inserir (int *v, int *n, int x) {
    // caso 1: inserir em uma lista vazia
    ...
    // caso 2: inserir no fim da lista
    ...
    // caso 3: inserir no inicio da lista
    // caso 4: inserir no meio da lista
    for (j = n; j > i; j--) {
        v[j] = v[j-1];
    }
    v[i] = x;
    (*n)++;
    ...
}
```

LES: Listas Estáticas Sequenciais

- **Remoção:** desloca para esquerda

Quantidade: 5



Item a ser removido: 80

(Remoção do meio da lista.)

Quantidade é maior que ZERO, significa que a há itens na lista.

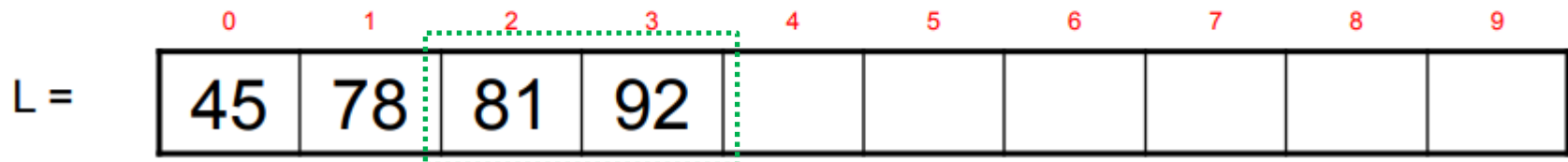
Portanto:

- Invoque a rotina de busca para o elemento a ser removido.
- • Caso o elemento esteja na lista, remova-o e decremente Quantidade de uma unidade
- Caso contrário, exiba uma mensagem.

LES: Listas Estáticas Sequenciais

- **Remoção:** desloca para esquerda

Quantidade: 4



Item a ser removido: 80

(Remoção do meio da lista.)

Quantidade é maior que ZERO, significa que a há itens na lista.

Portanto:

- Invoque a rotina de busca para o elemento a ser removido.
- • Caso o elemento esteja na lista, remova-o e decremente Quantidade de uma unidade
- Caso contrário, exiba uma mensagem.

Faça você mesmo!

```
// Recebe um vetor, a quantidade de elementos (referencia)
// e o elemento a ser removido.
// Procura a posicao de remocao i.
// Desloca os elementos para a esquerda caso necessário.
void remover (int *v, int *n, int x) {
    int i = buscaBin (v, *n, x);
    if (i != -1) {
        // desloca para a esquerda
        ...
        (*n)--;
    }
}
```