

---

## Laboratório #1 - lab1a.c e lab1b.c

Estacionamento da Rua Esquerda

---

**Prof. Valério Rosset**

### 1 DEFINIÇÃO DO PROBLEMA A

O estacionamento da Rua Esquerda, localizado São José dos Campos é composto por uma única alameda e pode estacionar, no máximo, 10 carros. Existe apenas uma entrada/saída, que justamente dá acesso à Rua Esquerda. Alastor, proprietário do estacionamento, deseja obter informações adicionais a respeito das movimentações realizadas durante um dia no seu estabelecimento.

#### TAREFA A

Desenvolva um algoritmo em linguagem C - **utilizando lista estática sequencial** - que reproduza as atividades executadas em um dia de trabalho no estacionamento da Rua Esquerda. O programa deverá imprimir uma mensagem sempre que um carro chegar ou sair. Se um carro chegar e não houver vaga no estacionamento, o programa deverá imprimir esta informação. Se não houver vaga, o carro partirá sem entrar no estacionamento. Quando um carro sair do estacionamento, a mensagem deverá incluir o número de carros que precisaram ser manobrados para fora do estacionamento para permitir que esse carro específico saia.

#### ENTRADA

A entrada contém um único conjunto de teste, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). O conjunto de teste é composto por  $L + 1$  linhas,  $1 \leq L \leq 100$ .

1. A primeira linha contém um valor inteiro  $L$ ,  $1 \leq L \leq 100$ , que representa a quantidade de movimentações (entradas e saídas) serão realizadas.
2. Em cada uma das  $L$  linhas seguintes são dadas as informações contém uma letra - 'E' para entrada ou 'S' para saída seguida de um espaço e o número da placa do carro que é composto por 4 algarismos.

Exemplo de entrada:

```
6
E 1234
E 6543
E 5432
S 1234
E 6542
E 8974
```

Para ler os caracteres (E e S) é preciso observar que o `\n` (próxima linha) é considerado um caractere, portanto a leitura dos caracteres E e S pode ser feita da seguinte forma:

```
scanf("\n%c",& tipo)
```

## SAÍDA

Como saída padrão, seu programa deve imprimir as seguintes informações:

1. entradas bem sucedidas: "O carro xxxx foi estacionado"
2. saídas bem sucedidas juntamente com o número de carros que precisaram ser manobrados para fora do estacionamento(sem contar a manobra realizada no carro que sairá):"O carro xxxx foi retirado com x manobras"
3. entradas que não foram realizadas por falta de espaço: "Estacionamento lotado"

O exemplo abaixo mostra como deve ser a saída do programa e qual a sintaxe em C utilizada para exibi-lá.

### Exemplo:

```
O carro 1234 foi estacionado
O carro 6543 foi estacionado
O carro 5432 foi estacionado
O carro 1234 foi retirado com 2 manobras
O carro 6542 foi estacionado
O carro 8974 foi estacionado
```

### Sintaxe :

```
printf("O carro %d foi estacionado", numcarro)
printf("O carro %d foi retirado com %d manobras", numcarro, cont)
printf("Estacionamento lotado")
```

Obs: Cada informação estará em um linha.

## EXEMPLOS

Entrada	Saída
13	O carro 1234 foi estacionado
E 1234	O carro 4567 foi estacionado
E 4567	O carro 8910 foi estacionado
E 8910	O carro 5678 foi estacionado
E 5678	O carro 9873 foi estacionado
E 9873	O carro 7891 foi estacionado
E 7891	O carro 4567 foi retirado com 4 manobras
S 4567	O carro 5432 foi estacionado
E 5432	O carro 3490 foi estacionado
E 3490	O carro 4387 foi estacionado
E 4387	O carro 7543 foi estacionado
E 7543	O carro 1290 foi estacionado
E 1290	Estacionamento lotado
E 4567	

## 2 DEFINIÇÃO DO PROBLEMA B

O escritório Viva possui um convênio com o estacionamento da Rua Esquerda, por isso os funcionários do escritório pagam uma quantia menor para estacionar. O estacionamento da Rua Esquerda possui um registro com os nomes dos funcionários do escritório e um número de identificação de cada um.

### TAREFA B

Desenvolva um algoritmo em linguagem C - **utilizando lista estática encadeada** - que forneça uma lista com os números de identificação **ordenados** dos funcionários do escritório Viva que estão utilizando o estacionamento da Rua Esquerda.

### ENTRADA

A entrada contém um único conjunto de teste, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). O conjunto de teste é composto por  $L + 1$  linhas,  $1 \leq L \leq 100$ .

1. A primeira linha contém um valor inteiro  $L$ ,  $1 \leq L \leq 100$ , que representa a quantidade de movimentações (entradas e saídas) serão realizadas.
2. Em cada uma das  $L$  linhas seguintes são dadas as informações contém uma letra - 'E' para entrada ou 'S' para saída seguida de um espaço e o número de identificação do funcionário que é composto por 2 algarismos.

Exemplo de entrada:

6  
E 12

E 34  
E 89  
S 12  
E 65  
E 10  
S 89

Para ler os caracteres (E e S) é preciso considerar o n (alteração de linha) na leitura (pois o n também é considerado um carácter), portanto a leitura pode ser feita da seguinte forma:

```
scanf("\n%c",& tipo)
```

## SAÍDA

Como saída padrão, seu programa deve imprimir uma lista com os números de identificação dos funcionários que estão no estacionamento. Quando um funcionário entra (E) seu número de identificação deve ser inserido na lista e quando um funcionário sai do estacionamento (S) seu número de identificação deve ser retirado da lista. A lista deve estar ordenada. Quando ocorrer uma entrada(E) ou uma saída (S) a lista deve ser impressa.

O exemplo abaixo mostra como deve ser a saída do programa e qual a sintaxe em C utilizada para exibi-lá.

### Exemplo:

12  
12 34  
12 34 89  
34 89  
34 65 89  
10 34 65 89  
10 34 65

**Sintaxe para impressão de cada elemento da lista:** printf("%d ", listaAux->elemento[i].valor)

Obs: É preciso que tenha um espaço entre os elementos.

## EXEMPLOS

Entrada	Saída
12	12
E 12	12 45
E 45	12 45 89
E 89	12 45 56 89
E 56	12 45 56 89 98
E 98	12 45 56 78 89 98
E 78	12 56 78 89 98
S 45	12 54 56 78 89 98
E 54	12 34 54 56 78 89 98
E 34	12 34 43 54 56 78 89 98
E 43	12 34 43 54 56 75 78 89 98
E 75	34 43 54 56 75 78 89 98
S 12	

#### **OBSERVAÇÕES**

- É fundamental que o programa imprima apenas o valor de saída.
- Os exercícios devem ser postados no site <http://judge.sjc.unifesp.br>.
- Cada aluno deverá postar apenas um exercício dentro da data limite.
- Os plágios serão penalizados. Como não será possível saber quem copiou, todos os casos envolvidos receberão punições iguais, portanto, não divulgue seu código.