

As soluções dos exercícios a seguir pressupõem o uso de TADs, principalmente o **TAD Lista**. Uma característica do ambiente *Sharif* é que a solução de cada problema esteja em um único arquivo **.c**, para que possa ser submetida (enviada) para ser avaliada no ambiente *Sharif*. Nesse sentido, ao trabalhar com TADs, antes de submeter uma solução, crie um único arquivo **.c**, cujo conteúdo segue o formato abaixo:

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

<TADs: conteúdo do(s) arquivo(s) .h>

<TADs: conteúdo do(s) arquivo(s) .c , sem #include "TAD.h">

int main () {
    <seu código>
}
```

O “uso do TAD” significa que a função **main** deve utilizar somente as operações presentes no arquivo **TAD.h**, ou seja, conhecer que operações estão previstas para o TAD.

Problema 01) A lista de inimigos de Arya (série “Game of Thrones”)

Lista de Arya: “Cersei. Walder Frey. Montanha. Meryn Trant.”

Para se manter motivada, Arya sempre se lembra da lista de inimigos que ela mais odeia. O principal objetivo de sua jornada é acabar com todos na sua lista!

Entretanto, às vezes algum inimigo dela pode ser morto por outra pessoa. Quando ela descobre que tal inimigo morreu, ela o remove da sua lista. Além disso, Arya também pode fazer novos inimigos durante sua jornada. Quando ela faz um novo inimigo, tal inimigo é incluído na sua lista.

Arya quer acabar com seus inimigos um por um, na mesma ordem em que aparecem na sua lista. A qualquer momento, ela pode se perguntar sobre o conteúdo da lista. Para tal, dados dois inimigos **a** e **b**, ela deve determinar quantos inimigos estão na lista entre **a** e **b**, excluindo ambos. Ajude Arya respondendo essas perguntas.

Entrada: A primeira linha contém um o número inteiro N ($1 \leq N \leq 100$), que se refere ao número de inimigos inicialmente em sua lista. Considere que os inimigos são identificados por números de 1 a 100, inclusive. A próxima linha contém N números inteiros, separados entre si por um espaço em branco, com o conteúdo inicial da lista de Arya. As próximas linhas descrevem as operações possíveis, segundo os formatos:

- **I p1 p2** ($1 \leq p1, p2 \leq 100$) : inserir o inimigo **p2** após o inimigo **p1**; é garantido que **p1** está na lista, e **p2** não está na lista;
- **R p1** ($1 \leq p1 \leq 100$) : remover o inimigo **p1** da lista; é garantido que **p1** está na lista;
- **Q p1 p2** ($1 \leq p1, p2 \leq 100$) : determinar quantos inimigos estão na lista entre **p1** e **p2**, excluindo ambos; é garantido que **p1** e **p2** estão na lista.
- **F** : termina as operações.

Saída: imprima uma linha com o resultado de cada operação do tipo **Q**, na ordem em que aparecem na entrada; ao final, imprimir o conteúdo da lista.

Importante 1: Use o **TAD Lista**, para suportar listas de números inteiros:

- arquivo **Lista.h**;
- arquivo **Lista_encadeamento_simples.c** (implementação por encadeamento simples);
- arquivo **programa.c** para a função **main**.
- para compilar: **gcc -o programa Lista.h Lista_encadeamento_simples.c programa.c**

Importante 2: Não altere o arquivo **Lista.h**. Esteja atento(a) se para concluir/corrigir a implementação no arquivo **Lista_encadeamento_simples.c**.

Entrada	Saída
3	1
3 8 2	2
Q 3 2	1
I 8 9	3 9 2 1
Q 3 2	
R 8	
I 2 1	
Q 9 1	
F	

Problema 02) Implementação 2 A lista de inimigos de Arya (série “Game of Thrones”)

Refere-se à outra solução para o **Problema 01**, pois requer uma outra forma de implementar.

Importante 1: Use o **TAD Lista**, para suportar listas de números inteiros:

- arquivo **Lista.h**;
- arquivo **Lista_vetor.c** (implementação por vetor);
- arquivo **programa.c** para a função **main**.
- para compilar: **gcc -o programa Lista.h Lista_vetor.c programa.c**

Importante 2: Não altere o arquivo **Lista.h**. Esteja atento(a) se para concluir/corrigir a implementação no arquivo **Lista_vetor.c**.

Problema 03) Agenda de contatos

Em uma cidade pequenina, seus habitantes ainda possuem o hábito de ‘anotar’ contatos em uma agenda de papel. Um desses habitantes, Fulana, também anota em sua agenda de contatos o número de vezes em que ela ligou para uma pessoa. Como a modernidade está chegando na cidade, Fulana precisa de um programa que a permitisse: inserir um novo contato; remover um contato; registrar para quem ela fez cada ligação.

Entrada: Cada linha do arquivo de entrada pode possuir algum dos seguintes formatos:

- **I nome fone ligação** : incluir uma pessoa com os seguintes dados, separados por um espaço em branco: primeiro nome do contato (**nome**), não há nome composto; telefone do contato (**fone**); e o número de vezes que ligou para esse contato (**ligação**); sempre inserir um novo contato no início da lista;
- **R nome** : remover um contato que possui o nome na lista (caso esteja presente);
- **L nome** : incrementar o número de ligações para um contato que possui o nome na lista (caso esteja presente).
- **F** : terminar as operações.

Saída: Imprimir toda a lista de contatos, cada contato em linha separada, com os dados: nome (primeiro nome), telefone e número de vezes que ligou para esse contato, separados por um espaço em branco. Antes de imprimir, ordene a lista pelo nome do contato (ordenação estável). É garantido que não existe mais de uma pessoa com o mesmo nome.

Importante 1: Use o **TAD Lista**, para suportar listas de números inteiros:

- arquivo **Elemento.h**;
- arquivo **Elemento.c**;
- arquivo **Lista2.h**;
- arquivo **Lista2_encadeamento_simples.c** (implementação por encadeamento simples);
- arquivo **programa.c** para a função **main**.

- para compilar: `gcc -o programa Elemento.h Elemento.c Listas.h Lista2_encadeamento_simples.c programa.c`

Importante 2: Não altere os arquivos **Elemento.h** e **Lista2.h**. Esteja atento(a) se para concluir/corrigir a implementação no arquivo `Lista_encadeamento_simples2.c`.

Importante 3: Qual a diferença entre os arquivos **Lista2.h** e **Lista.h** (dos Problema 01e 02) ?

Entrada	Saída
I jose 333-33 3 I bia 999-99 9 I bela 777-77 7 I maria 555-55 5 I rui 888-88 8 L maria L pedro R pedro R jose I junior 111-11 1 F	bela 777-77 7 bia 999-99 9 junior 111-11 1 maria 555-55 6 rui 888-88 8