

Instituto de Computação - Unicamp

MC102 - Algoritmos e Programação de Computadores

Laboratório 12 - Editor de Discursos da Aliança Rebelde

Prazo de entrega: **28/04/2017 23:59:59**

Peso: **1**

Professor: Eduardo C. Xavier

Professor: Guido Araújo

Monitor: Arthur Pratti Dadalto

Monitor: Cristina Cavalcante

Monitor: Klairton de Lima Brito

Monitor: Luís Felipe Mattos

Monitor: Paulo Finardi

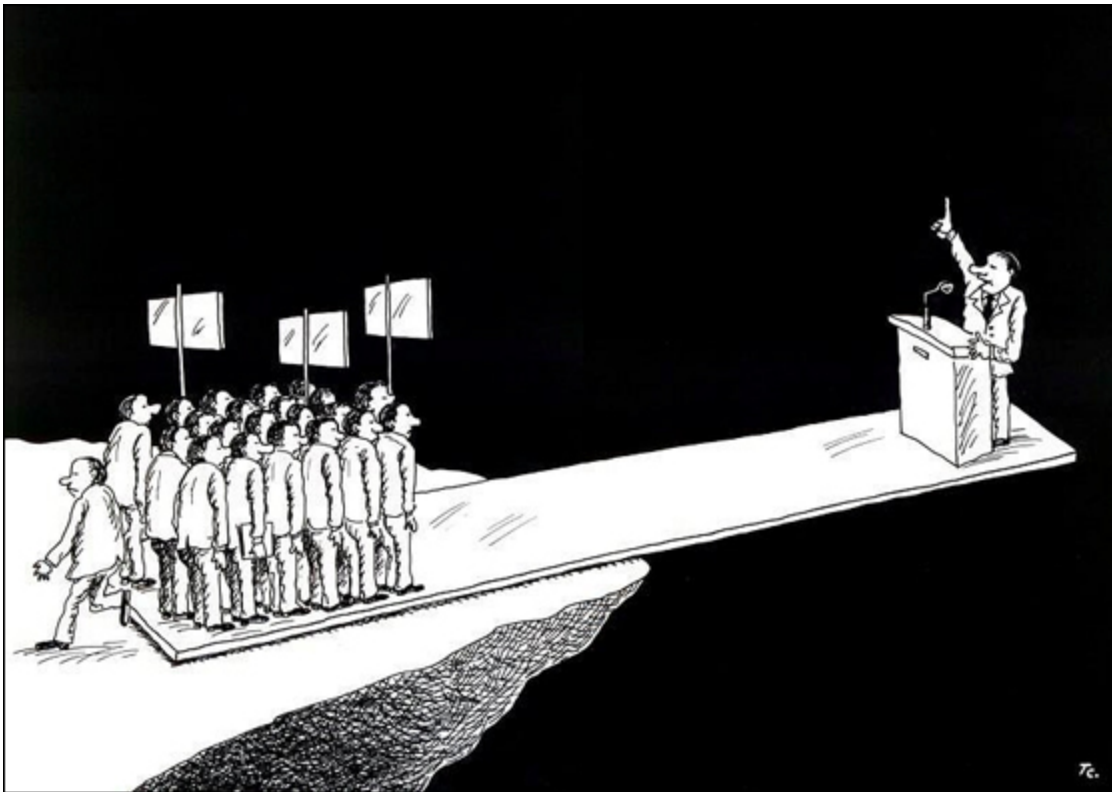
Monitor: Paulo Lucas Rodrigues Lacerda

Monitor: Pedro Alves

Monitor: Renan Vilas Novas

Monitor: Vinicius de Novaes Guimarães Pereira

Descrição



Os políticos da Aliança Rebelde realizam muitos discursos pela galáxia, e o processo de criação dos discursos demanda muito tempo, principalmente a parte de escrevê-los. Os redatores dos discursos notaram que as operações mais utilizadas por eles durante a escrita eram: apagar, substituir e inverter uma palavra. Para agilizar o processo de criação dos seus discursos, os políticos da Aliança Rebelde precisam de um editor de texto com estas três funções.

Objetivo

Como programador da Aliança Rebelde, você deve criar um programa que, dado um texto, um identificador de uma operação e seus parâmetros, realiza as modificações necessárias no texto. Os identificadores de cada operação são representados por uma letra, sendo elas:

- D - Apagar
- I - Inverter
- R - Substituir

- Q - Sair do programa

Os parâmetros de cada identificador são informados imediatamente depois do mesmo, sendo que os parâmetros são:

- O identificador D é seguido de uma palavra que denotamos por `palavra_deletar` com tamanho máximo de 50 caracteres. TODAS as ocorrências da palavra no texto devem ser removidas.
- O identificador I é seguido de uma palavra que denotamos por `palavra_inverter` com tamanho máximo de 50 caracteres. TODAS as ocorrências da palavra no texto devem ser invertidas.
- O identificador R é seguido de duas palavras denotadas por `palavra_velha` e `palavra_nova`, ambas com tamanho máximo de 50 caracteres. TODAS as ocorrências da palavra `palavra_velha` no texto devem ser substituídas por `palavra_nova`.

OBSERVAÇÃO:

- Note que uma mesma palavra pode estar escrita utilizando letras maiúsculas e/ou minúsculas, por exemplo, "Rebeldes" e "rebeldes" são palavras escritas utilizando caracteres diferentes, mas são a mesma palavra (Seu editor deverá ser CASE INSENSITIVE).
- O texto deve sempre conter um único espaço separando palavras. Desta forma, se por exemplo for substituída `bananada` por `dado` no texto `bananada de marmelo`, o resultado final deve ser
dado de marmelo
e não
dado de marmelo.

Entrada

A entrada consiste de um texto de no máximo 1000 caracteres, e uma série de identificadores de operações e seus respectivos parâmetros.

Dica para a linguagem C: leia o texto da seguinte forma:

```
fgets(texto, 1000, stdin);
```

Saída

Deverá ser impresso o texto lido como entrada e, para cada operação realizada sobre o texto, deve ser impresso o texto com a modificação decorrente da operação e o programa deve ler outra operação.

Quando o programa ler o operador `q`, que representa a operação de sair, o programa deve encerrar a execução.

Dica para a linguagem C: imprima a saída da seguinte forma:

```
printf("%s", texto);
```

Exemplos

Teste 01

Entrada

```
PMACINU Casa Vermelha de Madeira e Barro
D
madeira
D
E
R
vermelha
Estadual
R
casa
Universidade
R
barro
Campinas
I
```

PMACINU

Q

Saída

PMACINU Casa Vermelha de Madeira e Barro

PMACINU Casa Vermelha de e Barro

PMACINU Casa Vermelha de Barro

PMACINU Casa Estadual de Barro

PMACINU Universidade Estadual de Barro

PMACINU Universidade Estadual de Campinas

UNICAMP Universidade Estadual de Campinas

Teste 02

Entrada

idIOMaTico charcUTaRiA ChARCutAria debaTIdO InteGRo MEDALHAO

R

idiomatico

cOmItIva

R

debatido

PALACIO

R

medalhao

EnteAdO

D

charcutaria

R

enteado

TeoCraTICO

I

palacio

I

integro

D

oicalap

D

orgetni

Q

Saída

```
idIOMaTico charcUTaRiA ChARCutAria debaTIdO InteGRo MEDALHAO
cOmItIva charcUTaRiA ChARCutAria debaTIdO InteGRo MEDALHAO
cOmItIva charcUTaRiA ChARCutAria PALACIO InteGRo MEDALHAO
cOmItIva charcUTaRiA ChARCutAria PALACIO InteGRo EnteAdO
cOmItIva PALACIO InteGRo EnteAdO
cOmItIva PALACIO InteGRo TeoCraTICO
cOmItIva OICALAP InteGRo TeoCraTICO
cOmItIva OICALAP oRGetnI TeoCraTICO
cOmItIva oRGetnI TeoCraTICO
cOmItIva TeoCraTICO
```

Teste 03

Entrada

```
inVoCaR PEStanEjar tribUtaCAo matErnO EpIstoLAR amOViveL eQUino
D
amovivel
R
invocar
GERaDoR
D
materno
D
tributacao
Q
```

Saída

```
inVoCaR PEStanEjar tribUtaCAo matErnO EpIstoLAR amOViveL eQUino
inVoCaR PEStanEjar tribUtaCAo matErnO EpIstoLAR eQUino
GERaDoR PEStanEjar tribUtaCAo matErnO EpIstoLAR eQUino
GERaDoR PEStanEjar tribUtaCAo EpIstoLAR eQUino
GERaDoR PEStanEjar EpIstoLAR eQUino
```

Para mais exemplos, consulte os [testes abertos no Susy](#).

Observações

- O número máximo de submissões é **10**;
- O seu programa deve estar completamente contido em um único arquivo denominado `lab12.c` .
- Para a realização dos testes do SuSy, a compilação dos programas desenvolvidos em C irá considerar o comando:

```
gcc -std=c99 -pedantic -Wall -o lab12 lab12.c ;
```
- Você deve incluir, no início do seu programa, uma breve descrição dos objetivos do programa, da entrada e da saída, além do seu nome e do seu RA;
- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.

CrITÉrios importantes

Independentemente dos resultados dos testes do SuSy, o não cumprimento dos critérios abaixo implicará em nota zero nesta tarefa de laboratório.

- Para os programas em linguagem C, os headers aceitos para essa tarefa são `stdio.h` e `string.h` .