

UNIVERSIDADE FEDERAL DO PIAUÍ - CAMPUS SHNB  
SISTEMAS DE INFORMAÇÃO  
ALGORITMOS E PROGRAMAÇÃO II  
Trabalho Prático Final

## 1 Editor de Texto Básico

Você está escrevendo um editor de texto simples que suporta apenas os dois comandos a seguir:

- "type c" onde c é um caractere: Anexa o caractere c ao final do texto atual.
- "undo t" onde t é um inteiro: Desfaz todas as operações que foram executadas nos t segundos anteriores na ordem inversa.

Por exemplo, considere a seguinte sequência de comandos:

- Segundo 1: type a
- Segundo 2: type b
- Segundo 3: type c
- Segundo 5: undo 3

No final do segundo 3, o texto é "abc". No segundo 5, todos os comandos realizados nos 3 segundos anteriores são desfeitos na ordem inversa. Isso significa que 'c' é removido e, em seguida, 'b' é removido. O texto se torna apenas "a".

Observe que os comandos "desfazer" também podem ser desfeitos. Por exemplo:

- Segundo 1: type a
- Segundo 2: type b
- Segundo 3: undo 2
- Segundo 4: undo 2

Depois do segundo 2, o texto é "ab". Depois do segundo 3, tudo é desfeito e o texto fica vazio. No segundo 4, o "desfazer" anterior é desfeito, então o texto se torna "ab" novamente. Então, o "tipo b" também é desfeito e o texto se torna apenas "a".

## 2 Instruções sobre Implementação

O programa deve ser implementado na linguagem C. O código deve receber dois parâmetros de entradas diretamente da entrada principal (stdin). O primeiro parâmetro é um arquivo contendo informações de entrada. O segundo parâmetro também é um arquivo, onde o deve ser escrito o resultado.

Para padronizar a correção do trabalho prático, o código deve ser compilado e executado em terminal do sistema linux nas máquinas dos laboratórios 813 ou 814. A execução do programa será via linha de comando, por exemplo, *./tp entrada.txt saida.txt*, onde *entrada.txt* e *saida.txt* são, respectivamente, os arquivos de entrada e saída requeridos como o primeiro e segundo parâmetros.

### 3 Entrada e Saída

Os arquivos de entrada e saída mencionados na seção anterior devem seguir rigorosamente o formato descrito a seguir.

O arquivo de entrada pode ter várias linhas. Cada linha representa uma instância do problema e contém dois conjuntos de dados, cada conjunto entre chaves e separados por uma vírgula: (1) um conjunto de comandos separados por vírgulas e (2) um conjunto de tempos em segundos para cada comando do primeiro conjunto, também separados por vírgulas. Segue detalhes sobre a entrada:

- Cada instância do problema contém entre 1 e 50 comandos.
- O caractere  $c$  do comando "type  $c$ " é um *asc* entre 'a' e 'z'.
- O tempo  $t$  do comando "undo  $t$ " é um valor inteiro entre 1 e  $10^9$ .
- Os elementos do conjunto de tempos possui o mesmo tamanho do conjunto de comandos e estão em ordem crescente.
- Cada elemento do conjunto de tempos é um valor inteiro entre 1 e  $10^9$ .

O arquivo de saída contém uma linha de resposta para cada instância do problema respectivamente. Cada linha representa uma *string* resultante dos comandos de instância de problema.

Seguem exemplos de entrada e saída do programa:

Entrada:	Saída:
{type a,type b,type c,undo 3},{1,2,3,5}	a
{type a,type b,undo 2,undo 2},{1,2,3,4}	a
{type a,undo 1,undo 1},{1,2,3}	a
{undo 1},{1}	
{type a,type b,type c,undo 10},{1,2,3,1000}	abc

### 4 Documentação

A documentação deve conter pelo menos quatro seções, organizadas da seguinte forma:

- Uma visão geral da solução proposta
- A organização da solução, ou seja, como a solução foi modularizada.
  - Não inclua código em C!
  - Use algoritmos genéricos (exemplo, portugol) ou diagramas de fluxos para descrever a principal função ou módulo da solução.
- Análise de custo da solução proposta.
- Conclusão curta em um parágrafo.

A análise de custo deve explicar quantas vezes a função mais importante do programa (ou trecho dessa função) é executada em relação à quantidade de comandos. Gráficos e/ou tabelas podem ser utilizados nessa análise.

A documentação deve conter no mínimo 7 e no máximo 10 páginas, seguindo os modelos doc ou latex a serem disponibilizados no compartilhamento. A partir desses modelos deve ser gerado um arquivo PDF para a entrega.

## 5 Avaliação

O trabalho prático pode ser feito em grupos de até **3 componentes** e os pontos desse trabalho terão a seguinte organização:

- Funcionamento correto do programa (6 pontos)
- Documentação (4 pontos)

A documentação (um arquivo PDF) e códigos fonte do trabalho devem ser compactados em arquivo do tipo zip e submetidos no sistema Sigaa. Após a submissão, será agendado entrevistas com os grupos. O objetivo da entrevista é verificar se cada membro do grupo participou do desenvolvimento do trabalho. **Caso seja observado falta de conhecimento do trabalho entregue, haverá reduções da nota do membro, podendo alcançar nota 0 o membro que desconhecer totalmente o trabalho.**

**Atenção:** cópias de código ou soluções idênticas do problema, levará a **nota zero** para todos os membros dos grupos envolvidos nas cópias e soluções.

## 6 Datas

- Submissão no Sigaa na sexta-feira (02/08/2018) até as 23:59.
- Entrevistas a serem agendadas posteriormente com grupos.

**Boa sorte!**