

TRABALHO PRÁTICO DE AQUECIMENTO**Observações:**

1. Comece a fazer este trabalho imediatamente. Você nunca terá tanto tempo para resolvê-lo quanto agora!
2. **Data de Entrega:** até 11 de abril de 2011, às **23:59 horas**, ou antes. Após essa data haverá uma penalização por atraso: 2^d , onde d é o número de dias de atraso.
3. Envie este trabalho para o endereço eletrônico `esub.para.loureiro@gmail.com` tendo como assunto `[PAA 2011/1 TPA: "seu nome completo"]` e como anexo um arquivo zip, descrito abaixo, com o nome `TPA-"SeuNomeCompleto".zip` onde o string "SeuNomeCompleto" é o seu nome completo sem espaços em branco.
Exemplo para o aluno Zoroastro Felizardo e Sortudo:
 - Assunto: `[PAA 2011/1 TPA: Zoroastro Felizardo e Sortudo]`
 - Arquivo zip: `TPA_ZoroastroFelizardoESortudo.zip`
4. O seu trabalho deve ser executado em alguma máquina do ambiente computacional do Departamento de Ciência da Computação da UFMG, onde os monitores irão avaliá-lo. No arquivo `leiam.txt`, a ser incluído no arquivo zip, você deve dizer qual é o ambiente computacional para executar o seu TP bem como todas as instruções necessárias.
5. Linguagem de programação do trabalho: C ou C++.

Especificação do Problema: Auxiliar de Redação

A publicação dos resultados de pesquisa é uma parte muito importante de todo o trabalho científico. Normalmente a publicação desses resultados ocorre na forma de artigos em congressos e periódicos depois de serem revisados quanto à forma e conteúdo. Infelizmente, na academia é comum encontrar artigos difíceis de ler e com erros de ortografia e de gramática.

Para ajudar na parte específica da forma, existem algumas ferramentas que podem nos auxiliar. Por exemplo, no Unix existe um utilitário chamado `ispell` que faz a verificação da grafia ("*spelling checker*") de textos escritos em inglês. Para outros ambientes e línguas existem ferramentas específicas. Existem também ferramentas comerciais baseadas em técnicas de inteligência artificial que tentam descobrir erros gramaticais, estilo literário e outras informações como estatísticas sobre parágrafos.

O objetivo deste exercício de programação é especificar, projetar e implementar um sistema que o ajude no processo de redação. Esta é uma tarefa que exige um projeto cuidadoso das estruturas de dados e de seus respectivos algoritmos para poderem ser utilizados de forma eficiente na prática.

Considerações a serem feitas

- Este programa deverá ler um arquivo contendo um texto sem nenhuma formatação ("arquivo ASCII") onde cada sentença termina por um sinal de pontuação (".", "?", "!").
- Cada parágrafo é separado por pelo menos uma linha em branco.
- Desconsidere na análise a ser feita pelo seu programa "*stop words*" como artigos (a, o, as, os) e conjunções (e, ou) que não possuem conteúdo semântico. A lista de *stop words* deve ser fornecida no arquivo `stopwords` a ser criado pelo aluno, sendo que cada *stop words* deve aparecer em uma linha.

- A saída do seu programa irá fazer referência ao arquivo de entrada. Considere a primeira linha do arquivo de entrada como “1”, a segunda como “2” e assim por diante, para fazer essa referência.
- Uma palavra é uma sequência de letras delimitada por branco, “coluna da esquerda”, “coluna da direita” e símbolos de pontuação.
- Faz parte do projeto do seu programa fornecer uma saída legível.
- Os arquivos a serem usados para teste serão disponibilizados na página da disciplina.
- Outras suposições podem ser feitas desde que devidamente justificadas e não alterem as que foram descritas acima.

Saída

Este programa deverá fornecer as seguintes informações:

1. Para cada parágrafo do texto:

- (a) O número da linha no arquivo fonte onde começa;
- (b) O número de sentenças no parágrafo;
- (c) O número de palavras de cada sentença incluindo e não incluindo “*stop words*”.

Razão para incluir esta informação: Um problema que ocorre na redação são parágrafos formados por uma única sentença (o que deve ser evitado), sentenças muito longas ou curtas, ou mesmo uma mistura dessas situações.

2. Para cada palavra que não é “*stop word*”:

- (a) Cada palavra listada em ordem alfabética com a frequência de sua ocorrência no texto indicando as linhas no arquivo fonte onde aparecem;
- (b) No caso de uma palavra aparecer mais de uma vez no mesmo parágrafo indicar também a sua ocorrência no parágrafo e as distâncias entre elas incluindo “*stop words*”.

Razão para incluir esta informação: é comum repetir a mesma palavra várias vezes ao longo do texto ou num mesmo parágrafo. Os exemplos a seguir foram tirados de textos técnicos publicados na literatura:

- ... acerca de alimentação, moradia, saneamento, etc., acerca das condições de vida e trabalho. ... sobre nascimentos, doenças transmissíveis, etc. ... população por sexo, idade, procedência, etc. ... água, saneamento básico, recolhimento de lixo, etc. ... escolaridade, renda média, etc. ... ambulatórios, hospitais, etc. ... categoria profissional, lotação, etc. ... consultas, exames, vacinas, etc. ... custos, gastos, fontes financiadoras, etc.

[Normalmente num trabalho técnico deve-se evitar o uso de “etc” e ser mais claro enumerando ou listando os itens. Neste trecho “etc” aparece nove vezes.]

- Uma das tarefas mais importantes de todo o trabalho científico é a publicação dos resultados do trabalho de pesquisa.

[Palavra “trabalho” repetido duas vezes na mesma sentença separada por apenas sete vocábulos.]

3. Ocorrência de certas expressões sendo que para cada expressão deve ser indicada a linha no arquivo fonte onde aparece. O conjunto de expressões a serem analisadas deve ser lido do arquivo **expressoes** a ser criado pelo aluno, sendo que cada expressão deve aparecer em uma linha.

Razão para incluir esta informação: Existem certas expressões ou palavras em português que são frequentemente usadas mas não existem. O exemplo mais comum é “em nível de” ou “a nível de” como é bem discutido pelo filólogo Vianney Mesquita, professor da Universidade Federal do Ceará em um de seus livros.

Também temos exemplos de palavras que não existem em português ou são usadas com sentidos inadequados como “deletar” no sentido de remover, e “suportar” no sentido de ter uma funcionalidade. Os exemplos a seguir foram tirados de textos técnicos publicados na literatura:

- Garantir maior disponibilidade dos sistemas, na medida em que problemas locais (em nível de hardware, software) não afetam o conjunto.
- Este trabalho analisou apenas alguns órgãos responsáveis pela segurança pública a nível do Estado de Minas Gerais e a Polícia Federal.

Documentação

A sua documentação deve conter as seguintes partes (a serem incluídas no arquivo zip):

1. Uma documentação “mínima” que explique as fases de especificação, projeto e implementação (`doc1.pdf`).
2. Para cada algoritmo principal, uma descrição de seu projeto e análise de complexidade (`doc2.pdf`).
3. Para os arquivos a serem usados para teste (incluir os arquivos de teste, i.e., `teste1.in`, `teste2.in`, ...), a saída de execução de cada um deles (`teste1.out`, `teste2.out`, ...).
4. Um arquivo `leiam.txt` com informações sobre o ambiente computacional para executar o seu TP bem como todas as instruções necessárias.

Você pode optar por ter apenas um arquivo `doc.pdf` que inclui os dois arquivos de documentação.