



Pontifícia Universidade Católica de Minas Gerais
Curso de Ciência da Computação
Disciplina: Algoritmos e Estruturas de Dados II
Prof. Felipe Domingos da Cunha / Prof. Max do Val Machado

Trabalho Prático I - Aquecimento

1 Regras Básicas

1. Todos os programas devem ser desenvolvidos na linguagem Java (exceto quando dito o contrário).
2. Todas as apresentações de trabalho devem ser feitas no Linux usando o editor VIM.
3. Como o combinado não sai caro:
 - As cópias de trabalho serão encaminhadas para o colegiado;
 - O aluno perderá 1 ponto para cada método não comentado ou com comentário inútil.
4. As exceções devem ser perguntadas/discutidas/negociadas com o professor.
5. Os únicos métodos da classe String permitidos neste trabalho são *char charAt(int)* e *int length()*.
6. O arquivo **readme.txt** contém dicas sobre Linux, VIM e redirecionamento de entrada/saída.
7. Para contar as letras, consoantes e vogais desconsidere acentos e cedilha, ou seja, consideramos somente os caracteres cujos códigos ASCII estiverem entre 'A' e 'Z' ou 'a' e 'z'.
8. Não utilize as classes IO nem Scanner.
9. Para cada exercício: faça (entende-se análise, implementação e comentários), teste (várias vezes) e submeta no Verde. Os exercícios não testados/comentados serão penalizados.
10. A correção será automática através do Verde e de entrevista com o aluno nas aulas de laboratório.
11. A correção sempre será feita na versão do Linux disponível nos laboratórios. Qualquer problema devido ao uso de outras arquiteturas será de responsabilidade do aluno.

2 Aquecimento

1. Crie um método **iterativo** que receba como parâmetro uma string e retorne seu número de caracteres maiúsculos.
2. Teste o método anterior usando redirecionamento de entrada e saída. A entrada padrão é composta por várias linhas sendo que a última contém a palavra FIM¹. A saída padrão contém um número inteiro para cada linha de entrada.

3 Descrição

1. **Palíndromo** - Crie um método **iterativo** que recebe uma string como parâmetro e retorna true se essa é um palíndromo. Na saída padrão, para cada linha de entrada, escreva uma linha de saída com SIM / NÃO indicando se a linha é um palíndromo.
2. **Ciframento de César** - O Imperador Júlio César foi um dos principais nomes do Império Romano. Entre suas contribuições, temos um algoritmo de criptografia chamado “Ciframento de César”. Segundo os historiadores, César utilizava esse algoritmo para criptografar as mensagens que enviava aos seus generais durante as batalhas. A ideia básica é um simples deslocamento de caracteres. Assim, por exemplo, se a chave utilizada para criptografar as mensagens for 3, todas as ocorrências do caractere 'a' são substituídas pelo caractere 'd', as do 'b' por 'e', e assim sucessivamente. Crie um método **iterativo** que recebe uma string como parâmetro e retorna outra contendo a entrada de forma cifrada. Neste exercício, suponha a chave de ciframento três. Na saída padrão, para cada linha de entrada, escreva uma linha com a mensagem criptografada.
3. **Alteração Aleatória** - Crie um método **iterativo** que recebe uma string, sorteia duas letras minúsculas aleatórias (código ASCII \geq 'a' e \leq 'z'), substitui todas as ocorrências da primeira letra na string pela segunda e retorna a string com as alterações efetuadas. Na saída padrão, para cada linha de entrada, execute o método desenvolvido nesta questão e mostre a string retornada como uma linha de saída. Abaixo, observamos um exemplo de entrada supondo que para a primeira linha as letras sorteados foram o 'a' e o 'q'. Para a segunda linha, foram o 'e' e o 'k'.

EXEMPLO DE ENTRADA:

*o rato roeu a roupa do rei de roma
e que que que ewq ewq ewq
FIM*

EXEMPLO DE SAÍDA:

*o rqto roeu q roupq do rei de romq
k qwk qwk qwk kwq kwq kwq*

A classe Random do Java gera números (ou letras) aleatórios e o exemplo abaixo mostra uma letra minúscula na tela. Em especial, destacamos que: i) *seed* é a semente para geração de números aleatórios; ii) nesta questão, por causa da correção automática, a *seed* será quatro; iii) a disciplina de Estatística e Probabilidade faz uma discussão sobre “aleatório”.

¹A entrada padrão dos demais exercícios é da mesma forma.

```

Random gerador = new Random();
gerador.setSeed(4);
System.out.println((char)('a' + (Math.abs(gerador.nextInt()) % 26)));

```

4. **Álgebra Booleana** - Crie um método **iterativo** que recebe uma string contendo uma expressão booleana e o valor de suas entradas e retorna um booleano indicando se a expressão é verdadeira ou falsa. Cada string de entrada é composta por um número inteiro n indicando o número de entradas da expressão booleana corrente. Em seguida, a string contém n valores binários (um para cada entrada) e a expressão booleana. Na saída padrão, para cada linha de entrada, escreva uma linha de saída com SIM / NÃO indicando se a expressão corrente é verdadeira ou falsa.
5. **Is** - Crie um método **iterativo** que recebe uma string e retorna true se a mesma é composta somente por vogais. Crie outro método **iterativo** que recebe uma string e retorna true se a mesma é composta somente por consoantes. Crie um terceiro método **iterativo** que recebe uma string e retorna true se a mesma corresponde a um número inteiro. Crie um quarto método **iterativo** que recebe uma string e retorna true se a mesma corresponde a um número real. Na saída padrão, para cada linha de entrada, escreva outra de saída da seguinte forma X1 X2 X3 X4 onde cada X_i é um booleano indicando se a entrada é: composta somente por vogais (X1); composta somente por consoantes (X2); um número inteiro (X3); um número real (X4). Se X_i for verdadeiro, seu valor será SIM, caso contrário, NÃO.
6. **Palíndromo em C** - Na linguagem C, crie um método **iterativo** que recebe uma string como parâmetro e retorna true se essa é um palíndromo. Na saída padrão, para cada linha de entrada, escreva uma linha de saída com SIM / NÃO indicando se a linha é um palíndromo.

Observação: Uma linha de entrada pode ter caracteres não letras.

7. **Leitura de Página HTTP** - Leia duas strings sendo que a primeira é o nome de uma página web e a segunda, seu endereço. Por exemplo, “Pontifícia Universidade Católica de Minas Gerais” e “www.pucminas.br”. Em seguida, mostre na tela o número de vogais (sem e com acento), consoantes e dos padrões “< br >” e “< table >” que aparecem no código dessa página. A entrada padrão é composta por várias linhas. Cada uma contém várias strings sendo que a primeira é um endereço web e as demais o nome dessa página web. A última linha da entrada padrão contém a palavra “FIM”. A saída padrão contém várias linhas sendo que cada uma apresenta o número de ocorrência (valor x_i entre parênteses) de cada caractere ou string solicitado. Cada linha de saída será da seguinte forma: a(x_1) e(x_2) i(x_3) o(x_4) u(x_5) á(x_6) é(x_7) í(x_8) ó(x_9) ú(x_{10}) à(x_{11}) è(x_{12}) ì(x_{13}) ò(x_{14}) ù(x_{15}) ã(x_{16}) õ(x_{17}) â(x_{19}) ê(x_{19}) î(x_{20}) ô(x_{21}) û(x_{22}) consoante(x_{23}) < br >(x_{24}) < table >(x_{25}) nomepágina(x_{26}).