

Trabalho Prático

1 Regras Básicas

1. Estude bastante cada par de entrada/saída.
2. Todos os programas serão desenvolvidos nas linguagens Java ou C. Quando não especificada a linguagem, considere que a mesma é Java.
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. Para cada questão, devemos submeter apenas um arquivo (.java ou .c). Essa regra será necessária para a submissão de trabalhos no Verde e no identificador de plágio utilizado na disciplina.
6. Use o padrão Java para comentários.
7. Na leitura de um número inteiro, se seu valor for vazio ou não número, ele recebe zero.
8. Nos exercícios de ordenação ou estruturas de dados, se dois registros tiverem a mesma chave de pesquisa, eles serão ordenados pelo atributo nome.
9. 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'.
10. Para cada exercício: faça (entende-se análise, implementação e comentários), teste (várias vezes) e submeta no Verde. Os exercícios não testados/comentados serão penalizados.
11. A correção será automática através do Verde e de entrevista com o aluno nas aulas de laboratório.
12. Fique atento ao Charset dos arquivos de entrada e saída.

2 Dicas sobre Recursividade

1. Um método recursivo pode ser usado como uma estrutura de repetição e um erro básico consiste em confundir tais estruturas com os métodos recursivos. Para evitar esse erro, neste trabalho, não podemos usar estruturas de repetição nos métodos recursivos.
2. Outro erro básico consiste em criar variáveis globais (em Java, atributos) para darem suporte à recursividade. Para evitar esse erro, neste trabalho, não podemos usar variáveis globais.
3. Um terceiro erro básico consiste em passar o valor de retorno como parâmetro. Para evitar esse erro, neste trabalho, não podemos passar parâmetros extras aos do enunciado, exceto um contador para recursividade. O método abaixo contém um exemplo desse erro.

```
1 boolean contarLetrasMinusculas(String s, int i, int resp){
2     int resp;
3     if(i == s.length()){
4         resp = 0;
5     } else if ((s.charAt(i) >= 'a' && s.charAt(i) <= 'z') == false){
6         resp = contarLetrasMinusculas(s, i + 1, resp + 1);
7     } else {
8         resp = contarLetrasMinusculas(s, i + 1, resp);
9     }
10    return resp;
11 }
```

4. Quando criamos um método recursivo, normalmente, passamos um contador como parâmetro. Uma forma elegante de implementar essa solução recursiva consiste em criar dois métodos: o recursivo; e outro que chama o recursivo pela primeira vez e inicializa o valor do contador. Neste trabalho, devemos usar essa técnica em todos os métodos recursivos. Veja o exemplo abaixo:

```
1
2 boolean somenteLetrasMinusculas(String s){
3     return somenteLetrasMinusculas(s, 0);
4 }
5 boolean somenteLetrasMinusculas(String s, int i){
6     boolean resp;
7     if(i == s.length()){
8         resp = true;
9     } else if ((s.charAt(i) >= 'a' && s.charAt(i) <= 'z') == false){
10        resp = false;
11    } else {
12        resp = somenteLetrasMinusculas(s, i + 1);
13    }
14    return resp;
15 }
```

5. Uma boa prática de programação é colocar um único *return* em cada método. Por exemplo, o método anterior é mais indicado que a versão abaixo.

```

1
2 boolean somenteLetrasMinusculas(String s, int i){
3     if(i == s.length()){
4         return true;
5     } else if ((s.charAt(i) >= 'a' && s.charAt(i) <= 'z') == false){
6         return false;
7     } else {
8         return somenteLetrasMinusculas(s, i + 1);
9     }
10 }

```

3 Questões

1. **Arquivo em Java:** Faça um programa que leia um número inteiro n indicando o número de valores reais que devem ser lidos e salvos sequencialmente em um arquivo texto. Após a leitura dos valores, devemos fechar o arquivo. Em seguida, reabri-lo e fazer a leitura de trás para frente e mostre todos os valores lidos na tela. A entrada padrão é composta por um número inteiro n e mais n números reais. A saída padrão corresponde a n números reais mostrados um por linha de saída.
2. **Arquivo em C:** Refaça a questão anterior na linguagem C.
3. **Leitura de Página HTML em Java** - Leia duas strings sendo que a primeira é o nome de uma página web e a segunda, seu endereço. Por exemplo, “Pontificia 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}) nomepagina(x_{26}).