

Nome:_____

Prova 3º Bimestre**Informações**

- Quantidade de questões: **3**
- Total de pontos: **100**
- Peso no bimestre: **40%**

Instruções

1. Antes de iniciar, confira se esta cópia contém todas as questões e se estão legíveis.
2. Prova individual e com consulta.
3. A correta interpretação das questões faz parte da avaliação.
4. Todas as questões devem ser desenvolvidas e resolvidas em código, no projeto BlueJ que acompanha a disciplina.

Boa Prova!Tabela de Pontuações – **Uso exclusivo do professor!**

Questões	1	2	3	Total
Valor	50	40	10	100
Pontuação				

Questões

1. Sobre *máximo divisor comum*, leia:

Um modo de calcular o m.d.c. (máximo divisor comum) de dois ou mais números é utilizar a decomposição desses números em fatores primos.

1. decomponamos os números em fatores primos;
2. o m.d.c. é o produto dos fatores primos comuns.

Acompanhe o cálculo do m.d.c. entre 36 e 90:

$$36 = 2 \times \boxed{2 \times 3 \times 3}$$

$$90 = \boxed{2 \times 3 \times 3} \times 5$$

O m.d.c. é o produto dos fatores primos comuns, tal que $\text{m.d.c.}(36, 90) = 2 \times 3 \times 3$, portanto $\text{m.d.c.}(36, 90) = 18$. Escrevendo a fatoração dos números na forma de potência, temos:

$$36 = 2^2 \times 3^2$$

$$90 = 2 \times 3^2 \times 5$$

Portanto $\text{m.d.c.}(36, 90) = 2 \times 3^2 = 18$.

Com base nessas instruções, implemente em Java o que se pede:

- (a) [15 pontos] Um método que recebe um número inteiro n como parâmetro e retorna o primeiro número primo p tal que $p > n$. Por exemplo, se $n = 3$ então p será 5, e se $n = 14$ então p será 17.
- (b) [15 pontos] Um método que recebe um número inteiro como parâmetro e retorna uma lista com seus fatores primos decompostos.
- (c) [10 pontos] Um método que recebe dois números inteiros como parâmetros e calcule o m.d.c. entre eles utilizando o algoritmo dos fatores primos comuns.
- (d) [10 pontos] Faça a análise de complexidade assintótica para cada um dos métodos escritos anteriormente, classificando-os nos conjuntos *Big-O*, Ω e Θ .

2. Programe a classe `Histograma`, que recebe uma `String` como parâmetro de construção, e realiza uma contagem de quantidade de cada caractere presente na `String`, ignorando apenas os espaços em branco. A classe escreve essa relação através do método `toString`. A saída deve estar formatada da seguinte forma:

String do construtor:
Vê se tira notas boas!

String do método `toString()`:
a: 3
b: 1
e: 1
ê: 1
i: 1
n: 1
o: 2
s: 3
t: 2
V: 1
!: 1

Critérios de avaliação:

- (a) [20 pontos] Computação do histograma funciona.
- (b) [10 pontos] Computação do histograma funciona com complexidade $O(n)$. *Dica: use alguma TAD com tabelas de dispersão.*
- (c) [10 pontos] Formatação da saída (a ordem dos caracteres não é importante).

3. [10 pontos] No projeto `prova0` do BlueJ, há um arquivo texto sobre autoria e descrição do projeto. Além de preencher os campos adequadamente, crie um novo campo denominado “**PLANO DE ENSINO:**”. Preencha-o com toda a bibliografia deste curso, conforme indicado pelo **Plano de Ensino**.