

<b>1ª PROVA</b>	<b>Valor: 30,00</b>	<b>Nota Obtida:</b>	<b>Vista:</b>
<b>Curso:</b> Sistemas para Internet	<b>Disciplina:</b> Algoritmos e Programação		<b>Professor:</b> Bruno Queiroz Pinto
<b>Aluno:</b>	<b>Turma:</b>		<b>Data:</b>

### (7.5)Exercício 1: Modelo de Crescimento Populacional com Limitações

Crie um programa em Java e implemente um método recursivo *calcularPopulacao(int n)* que calcula o tamanho da população na geração **n**, segundo o modelo:

$$P_n = \begin{cases} 100 & \text{se } n = 0 \\ 300 & \text{se } n = 1 \\ 3P_{n-1} - 2P_{n-2} & \text{se } n \geq 2 \end{cases}$$

**Entrada:** Um número inteiro **n**, correspondente a geração a ser analisada.

**Saída:** O texto “População da geração **n** é **P<sub>n</sub>**”, onde **P<sub>n</sub>** será calculado pela função recursiva.

<b>Entrada</b>	<b>Saída</b>
1	População da geração 1 é 300
8	População da geração 8 é 51100

### (7.5)Exercício 2: Sistema de Evolução de Tecnologias em Inteligência Artificial

Em um modelo de evolução de IA, duas métricas principais (Capacidade e Estabilidade) evoluem de forma interdependente através de gerações. Cada geração depende das duas anteriores de maneira distinta.

**Relações Recursivas:**

**Capacidade:**

$$C(n) = \begin{cases} 1.0 & \text{se } n = 0 \\ 2.0 & \text{se } n = 1 \\ 1.5 \times C(n-1) - 0.3 \times E(n-1) + 0.2 \times C(n-2) & \text{se } n \geq 2 \end{cases}$$

**Estabilidade:**

$$E(n) = \begin{cases} 2.0 & \text{se } n = 0 \\ 1.0 & \text{se } n = 1 \\ 0.8 \times E(n-1) + 0.4 \times C(n-1) - 0.1 \times E(n-2) & \text{se } n \geq 2 \end{cases}$$

**Sinergia:**

$$S(n) = \sqrt{C(n) \times E(n)} \times \frac{C(n) + E(n)}{2}$$

Crie um programa em Java e implemente os métodos recursivos *capacidade(int n)* e *estabilidade(int e)* que permitem calcular a capacidade, estabilidade e sinergia da evolução de um modelo de IA na geração **n**, segundo os modelos acima.

**Entrada:**

Um número inteiro **n** que representa a geração da tecnologia.

**Saída:**

Imprima a saída conforme modelo abaixo:

*Geracao n:*

*Capacidade: C*

*Estabilidade: E*

*Sinergia: S*

Onde **C**, **E** e **S** são valores reais com 2 casas decimais.

**Entrada**

1

**Saída**

Geracao 1:

Capacidade: 2.00

Estabilidade: 1.00

Sinergia: 2.12

7

Geracao 7:

Capacidade: 20.69

Estabilidade: 11.09

Sinergia: 240.74

### (7.5)Exercício 3 :

Baseado no enunciado abaixo, crie um programa em Java para resolvê-lo. Você não deverá utilizar métodos da classe String, apenas os métodos length, charAt, toCharArray. Não pode utilizar as classes StringBuilder, StringBuffer e Character.

## STR18 - Contar ocorrência de letras

Por BRUNO QUEIROZ PINTO,  Brazil

Timelimit: 1

Faça um programa em Java que recebe uma frase e duas letras e imprima a letra que tenha a maior quantidade de ocorrências na frase. Caso, tenham a mesma quantidade de ocorrências, imprimir empate.

Obs.: É case-sensitive, ou seja, letras maiúsculas são diferentes das minúsculas.

#### Entrada

Uma String contendo a frase a ser analisada, dois caracteres a serem buscados.

#### Saída

Imprimir o caractere que tenha a maior quantidade de ocorrências na Frase, ou empate.

Samples Input	Samples Output
Uberlândia e b	empate
Diz-se que um software é case-sensitive ou possui "case sensitivity" quando ele é capaz de analisar uma cadeia de caracteres, avaliar a existência de caixa alta e caixa baixa e comportar-se de diferentes maneiras em função disso. a e	a

### (7.5)Exercício 4:

Resolva utilizando qualquer método da classe String. Não utilizar StringBuilder e StringBuffer

beecrowd | 5584

## STR20 - Multiplicar a soma dos números das palavras

Por BRUNO QUEIROZ PINTO,  Brazil

Timelimit: 1

Faça um programa que recebe uma frase, com diversas palavras. Some os dígitos numéricos em cada uma das palavras e então multiplique os valores encontrados. Se uma palavra não conter números ou o resultado da soma for 0, considere o valor 1 para essa palavra.

#### Entrada

Uma String contendo uma frase.

#### Saída

O resultado da multiplicação da soma dos dígitos numéricos presentes em cada palavra, no formato : "Resultado = " + valor

Samples Input	Samples Output
342fdas 332dfs 11dd dfd dfsf fdsf 123fsd	Resultado = 864

Obs.: Passar nos testes não significa estar correto, os testes não são completos.



**INSTITUTO FEDERAL**  
Triângulo Mineiro  
Campus Uberlândia Centro

**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
TRIÂNGULO MINEIRO – Campus Uberlândia Centro



**INSTITUTO FEDERAL**  
Triângulo Mineiro  
Campus Uberlândia Centro

**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
TRIÂNGULO MINEIRO – Campus Uberlândia Centro



**INSTITUTO FEDERAL**  
Triângulo Mineiro  
Campus Uberlândia Centro

**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
TRIÂNGULO MINEIRO – Campus Uberlândia Centro