

SIMULADO - Av de Aprendizagem II

Bom simulado!!!

1. AproximacaoDePi - Existem diferentes formas de aproximarmos o valor de PI, https://en.wikipedia.org/wiki/Approximations_of_%CF%80. Utilizando Trigonometria, o matemático Gregory–Leibniz criou a seguinte fórmula:

$$\pi = 4 \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

Quando aberto o somatório, percebemos que nada mais é do que a soma de várias frações, onde os denominadores são a sequência de números ímpares partindo de 1, e o sinal é intercalado entre positivo e negativo. Após isso a soma é multiplicada por 4 e obtém-se valores bem próximos de PI.

$$4 \left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \right)$$

Desenvolva um algoritmo que dada a quantidade de interações, calcule a aproximação de PI baseado na fórmula de Gregory Leibniz.

Exemplo de entrada e saída do Algoritmo, note que quanto mais interações são feitas, mais próximo chegamos do valor de PI.

```

C:\dev\tads\logica-prova-i>javac AproximacaoDePi.java

C:\dev\tads\logica-prova-i>java AproximacaoDePi
Quantas interacoes serao feitas?
10
3.0418396189294032

C:\dev\tads\logica-prova-i>java AproximacaoDePi
Quantas interacoes serao feitas?
1000
3.140592653839794

C:\dev\tads\logica-prova-i>java AproximacaoDePi
Quantas interacoes serao feitas?
100000
3.1415826535897198

C:\dev\tads\logica-prova-i>java AproximacaoDePi
Quantas interacoes serao feitas?
10000000
3.1415925535897915

C:\dev\tads\logica-prova-i>java AproximacaoDePi
Quantas interacoes serao feitas?
1000000000
3.1415926525880504

C:\dev\tads\logica-prova-i>

```

2. Fatores Primos - Desenvolva o código necessário para fazer a decomposição em fatores primos. Lembrando que um número é considerado primo quando possui apenas dois divisores, 1 e ele mesmo, por exemplo, o número 17 é primo pois nenhum número entre 2 e 16 tem resto 0 quando tentamos dividir o número 17. O número decomposto em fatores primos nada mais é do que reescrever o número como uma sequência de multiplicações onde todos os valores são primos, por exemplo:

$$225 = 3 * 3 * 5 * 5$$

$$1001 = 7 * 11 * 13$$

Neste exercício, você irá solicitar ao usuário um número positivo e maior que 1 e, apresentará sua decomposição em fatores primos, mostrando um valor primo por linha, valores repetidos aparecerão um ao lado do outro, por exemplo:

```

C:\dev\tads\logica-prova-i>javac FatoresPrimos.java

C:\dev\tads\logica-prova-i>java FatoresPrimos
Informe o numero a ser decomposto em fatores primos:
225
3 3
5 5

C:\dev\tads\logica-prova-i>java FatoresPrimos
Informe o numero a ser decomposto em fatores primos:
864
2 2 2 2 2
3 3 3

C:\dev\tads\logica-prova-i>java FatoresPrimos
Informe o numero a ser decomposto em fatores primos:
1001
7
11
13

```

3. Crie um programa que receba inicialmente um valor inteiro do usuário. Após, o programa deve ficar recebendo novos valores até que seja digitado o valor 0. Para cada valor digitado, o programa deve indicar se o valor é um múltiplo, divisor ou nada do número anterior.

Por exemplo:

```

Informe um valor inicial: 5
Informe outro valor: 10
    10 é múltiplo de 5
Informe outro valor: 9
    9 não é nada de 10
Informe outro valor: 3
    3 é divisor de 9
Informe outro valor: 0

```

4. Faça um programa que receba 3 valores inteiros. Os dois primeiros valores indicam um intervalo, o terceiro valor indica um divisor. O programa deve calcular a soma dos valores dentro do intervalo que são divisíveis pelo terceiro valor.

Por exemplo:

```

Informe 3 valores: 5 10 2
Resultado: 24

```

** 6, 8 e 10 são os valores somados dentro do intervalo 5 e 10 que são múltiplos de 2.

5. A fim de entender o crescimento populacional de diversos países, cientistas do IFRS solicitaram que você criasse um programa que recebesse 4 valores. Referentes ao

tamanho da população e a taxa de crescimento anual. Para cada caso, indique em quantos anos a população de um país ultrapassará a de outro. Execute dentro de um laço de repetição até que sejam digitados os valores 0 para a população do país. Se atente para o caso em que um país nunca ultrapassará o outro.

Exemplo:

Informe a população e a taxa do primeiro país:
100000 1

Informe a população e a taxa do segundo país: **80000**
10

Em 3 anos o tamanho da população do segundo país ultrapassará a do primeiro.

Informe a população e a taxa do primeiro país:
100000 10

Informe a população e a taxa do segundo país: **80000**
10

0 segundo país não ultrapassará o primeiro.

Informe a população e a taxa do primeiro país: **0 1**

Informe a população e a taxa do segundo país: **0 10**

FIM

6. Crie um programa que dado um valor inteiro, converta o número de forma que ele passe a valer seus dígitos na ordem inversa. Por exemplo, o número 1000 passa a valer 0001, ou seja, 1. O valor 1234 se torna 4321. **NÃO DEVE SER UTILIZADO STRINGS!** É um problema de repetição.

7. Faça um programa que dado um valor inicial indicando quantos valores serão digitados. Leia os valores e indique quantas quedas (quando um valor é menor que o anterior) aconteceram durante a sequência. Exemplo:

Informe a quantidade de valores: **5**

Informe os valores:

100 199 199 198 0

Resultado: **2 quedas**

**** Houve uma queda do 199 para o 198 e outro do 198 para o 0.**

8. Alunos do Campus Rio Grande desenvolveram um jogo de cartas onde cada carta representa um(a) professor@. Os professores possuem habilidades de teoria e prática que podem ir de 5 a 10. A forma de duelo entre as cartas é simples, a carta que tiver mais pontos na rodada é a vencedora. A pontuação da carta é calculada da seguinte maneira:

$$\text{pontuacao} = (\text{teoria} + \text{pratica})/2 + \text{bonus}$$

O valor do bônus é obtido através do lançamento de um dado de 6 lados.

Os alunos do TADS, desenvolveram uma versão digital do Jogo para testarem suas habilidades. No programa, primeiro deve ser indicado o número de partidas que serão simuladas. Depois, para cada partida serão indicado 4 valores referentes a teoria e prática de cada carta de professor. Para cada simulação, deve ser apresentado qual foi a carta vencedora ou se houve empate.

Exemplo:

Quantas partidas? 3

#1 Carta A: 5 8

#1 Carta B: 7 6

#1 Carta A: 6.5 + 3

#1 Carta B: 6.5 + 2

A VENCEU

#2 Carta A: 5 8

#2 Carta B: 7 8

#2 Carta A: 6.5 + 3

#2 Carta B: 15 + 6

B VENCEU

#3 Carta A: 9 5

#3 Carta B: 7 9

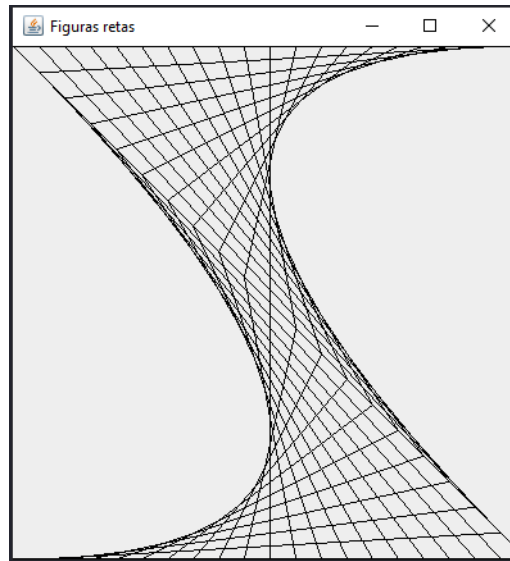
#3 Carta A: 7 + 5

#3 Carta B: 8 + 4

EMPATE

Para os exercícios 9 e 10. A instrução **g.drawLine(x0, y0, x1, y1)** desenha uma linha reta da posição (x0, y0) canto superior esquerdo, até a posição (x1, y1) canto inferior direito. Com essa instrução e as estruturas de repetição, é possível gerar as imagens solicitadas.

9. Modifique o código fonte, na região indicada, utilizando a instrução **g.drawLine(x0, y0, x1, y1)** para gerar a seguinte imagem



10. Modifique o código fonte, na região indicada, utilizando a instrução **g.drawLine(x0, y0, x1, y1)** para gerar a seguinte imagem

