

Instituto Federal de Educação Ciência e Tecnologia do RN - IFRN

Exercícios

- 001** Escreva um programa que leia dois números inteiros e informe se os mesmos são iguais ou não. Caso os números sejam iguais o programa deve mostrar a palavra **Iguais** (apenas o 'i' é maiúsculo). Caso os números sejam diferentes o programa deve mostrar a palavra **Diferentes**. Abaixo segue os exemplos de execução, com os arquivos de entrada e saída.

Exemplo de execução	Entrada (entrada.txt)	Saída (saida.txt)
<pre>\$ ruby iguais.rb 123 123 Iguais \$</pre>	<pre>123 123</pre>	<pre>Iguais</pre>

Exemplo de execução	Entrada (entrada.txt)	Saída (saida.txt)
<pre>\$ ruby iguais.rb 20 21 Diferentes \$</pre>	<pre>20 21</pre>	<pre>Diferentes</pre>

- 002** Escreva um programa que leia dois números inteiros e mostre o menor dos dois. O programa deve mostrar **APENAS** o número, nada mais.
- 003** Escreva um programa que leia três números inteiros e mostre o menor dos três. O programa deve mostrar apenas o número, nada mais.
- 004** Escreva um programa que leia uma nota e informe se a nota é válida. O programa deve mostrar a palavra **Valida** se a nota for válida ou **Invalida** se a nota não for válida. Observe que não há acento no texto a ser mostrado. Uma nota é válida se ela é maior ou igual a 0.0 e menor ou igual a 10.0.
- 005** Escreva um programa que leia duas notas de um aluno do IFRN, verifique se as mesmas são válidas, e informe se o aluno passou ou não. Considere a média de acordo com a fórmula abaixo:

$$\frac{(n_1 \times 2) + (n_2 \times 3)}{5}$$

onde n_1 representa a primeira nota e n_2 representa a segunda nota.

O programa deve mostrar a letra **A** se o aluno atingiu uma média maior-igual do que 6.0 ou **R** se o aluno não atingiu a média 6.0. Caso pelo menos uma das notas não seja válida o programa deve mostrar **I**.

- 006** Escreva um programa que leia um número inteiro e informe se o mesmo é par ou ímpar. Se o número for par o programa deve mostrar a palavra **Par**, caso contrário o programa deve mostrar a palavra **Impar**, ambas sem acento e apenas a primeira letra maiúscula.
- 007** Escreva um programa que leia um número inteiro e escreva seu valor absoluto. Nada mais deve ser mostrado pelo programa. Abaixo segue dois exemplos de execução do programa, um para o caso de o número ser negativo e outro para o caso de ele ser positivo.

Exemplo de execução	Entrada (entrada.txt)	Saída (saida.txt)
<pre>\$ ruby absoluto.rb 123 123 \$</pre>	<pre>123</pre>	<pre>123</pre>

Exemplo de execução	Entrada (entrada.txt)	Saída (saida.txt)
<pre>\$ ruby absoluto.rb -21 21 \$</pre>	<pre>-21</pre>	<pre>21</pre>

- 008** Escreva um programa que leia dois números inteiros e verifique se o maior é múltiplo do menor. O programa deve mostrar a palavra **MULTIPL0** caso o número maior seja múltiplo do maior. Nada mais deve ser mostrado pelo programa.
- 009** Escreva um programa que leia um número e mostre se o mesmo é divisor de 12345678. O programa deve mostrar a palavra **DIVISOR** caso o número seja divisor ou **NAO DIVISOR** caso o número não seja divisor.
- 010** Escreva um programa que leia 3 valores inteiros, verifique e mostre se os mesmos formam os comprimentos dos lados de um triângulo. O programa deve apenas mostrar a palavra **Triangulo** (apenas a primeira letra maiúscula, sem acento) caso os números formem os lados de um triângulo ou **Nao triangulo** (sem acento) caso não formem.
- 011** Escreva um programa que leia 3 números, verifique se eles formam os lados de um triângulo e mostre o tipo de triângulo: **equilátero**, **isósceles** ou **escaleno**. O programa deve mostrar apenas uma palavra dentre as seguintes: **Nenhum**, **Equilatero**, **Isosceles** ou **Escaleno**. Observe que não há acentos nas palavras.
- 012** Escreva um programa que leia 3 valores inteiros e mostre-os em ordem decrescente. Cada valor deve ser mostrado em uma linha. Nada mais deve ser mostrado.
- 013** Escreva um programa que leia 4 valores inteiros, elimine o maior e o menor e mostre a soma dos 2 restantes.
- 014** Escreva um programa que leia 3 números inteiros que representam uma data (dia, mês e ano) e informe se a data é válida. O programa deve mostrar a palavra **Valida** ou **Invalida** em uma linha. Nada mais deve ser mostrado.
NÃO é necessário verificar se o ano é bissexto
- 015** Escreva um programa que leia 3 inteiros que representam uma data (dia, mês e ano, necessariamente nessa sequência) e mostra a data do dia anterior, da seguinte forma:

dd/mm/aaaa

onde **dd** é o dia (sempre com dois algarismos), **mm** é o mês (também com dois algarismos) e **aaaa** é o ano (com quatro algarismos). Considere que a entrada contenha apenas datas válidas.

NÃO é necessário verificar se o ano é bissexto

- 016** Escreva um programa que leia duas datas (3 números inteiros para cada data) e mostre a quantidade de dias de diferença entre as datas. O valor mostrado deve ser inteiro positivo, sempre.

Desconsidere o dia 29 de fevereiro dos anos bissextos

- 017** Escreva um programa que leia 5 números inteiros e mostre a diferença entre o maior e o menor (*maior – menor*).
- 018** Escreva um programa que leia 2 notas de um aluno, que representam as notas do primeiro e do segundo bimestre, em seguida o programa calcula se o aluno ficou em recuperação, foi aprovado ou reprovado. Caso o aluno não tenha sido aprovado verifique se ele ficou em recuperação (nota entre 2,0 e 5,9). Se o aluno ficou em recuperação, o programa deve pedir a nota de recuperação e calcular média final do aluno. Finalmente o programa deve mostrar a nota final, com 1 casa decimal seguida da situação final do aluno: **APROVADO** ou **REPROVADO**, tudo com letras maiúsculas, como no exemplo abaixo:

8.5: APROVADO

O cálculo das médias parcial e final deve ser feito de acordo com o descrito no manual do aluno.

OBS.: TODOS os cálculos devem ser feitos considerando duas casas decimais

- 019** Escreva um programa que calcule se existe combustível suficiente e em quanto tempo um carro C_1 alcança um carro C_2 .

O programa deve ler a quantidade de combustível em C_1 , em litros (inteiro), o consumo em quilômetros por litro (real) e a velocidade de deslocamento (inteiro, em km/h). O programa deve também ler a velocidade de deslocamento de C_2 (inteiro, em km/h) e a distância entre C_1 e C_2 (inteiro, em km).

Com estes dados o programa deve calcular se C_1 alcança C_2 antes da gasolina acabar e em quanto tempo. Considere que todos os valores lidos são números inteiros. O tempo deve ser mostrado em **HH:MM:SS**, onde **HH** é a quantidade em horas, **MM** em minutos e **SS** em segundos, todos com dois algarismos. Apenas uma linha deve ser mostrada.

Exemplo de saída:

02:30:20

Esta saída informa que o C_1 alcança C_2 em duas horas, trinta minutos e 20 segundos. O programa deve desconsiderar as possíveis casas decimais dos segundos.

Caso não haja combustível suficiente ou a velocidade de C_2 seja maior que a de C_1 o programa deve mostrar o tempo --:--:--

020 Uma chave de um campeonato de futebol possui 3 times que devem jogar entre si. Para auxiliar a organização a classificar os times na chave escreva um programa que leia os nomes de 3 times de uma chave, os resultados dos jogos e mostre a classificação dos times na chave.

Regras:

- Uma vitória vale 3 (três) pontos
- Uma derrota vale 0 (zero) pontos
- Um empate vale 1 (um) ponto
- Caso haja dois times com a mesma pontuação os critérios de desempate, na ordem, são:
 1. Confronto direto.
 2. Saldo de gols.
 3. Maior quantidade de gols pró.
 4. Menor quantidade de gols contra.

O programa deve ler, primeiramente, um nome de time por linha, sendo 3 (quatro) o total de times. O programa deve, então, ler os resultados dos jogos, sendo o número de gols de cada time na partida por vez. Considere os times a seguir (digitados dessa ordem):

Baraunas
Potiguar
Campinense

O programa deve ler os resultados dos jogos na seguinte ordem:

Baraunas x Potiguar
Baraunas x Campinense
Potiguar x Campinense

Considere os seguintes resultados:

Baraunas	1 x 3	Potiguar
Baraunas	5 x 3	Campinense
Potiguar	4 x 2	Campinense

O programa deve mostrar os dados de cada time por linha, ordenado por classificação, com os seguintes dados:

Classificação:NomeTime:Pontos:QtdVitórias:QtdEmpates:QtdDerrotas:GolsPro:GolsContra:SaldoGols

Os gols pró e contra devem ser mostrados com duas casas decimais, com zeros a esquerda. O saldo de gols deve ser mostrado com 3 dígitos, onde o primeiro é o sinal de positivo/negativo (+ ou -).

Os exemplos de execução, entrada e saída seguem abaixo:

Exemplo de execução	Entrada (entrada.txt)	Saída (saida.txt)
<pre>\$ ruby futebol.rb Baraunas Potiguar Campinense 1 3 5 3 4 2 1:Potiguar:6:2:0:0:07:03:+04 2:Baraunas:3:1:0:1:06:06:+00 3:Campinense:0:0:0:2:05:09:-04 \$</pre>	<pre>Baraunas Potiguar Campinense 1 3 5 3 4 2</pre>	<pre>1:Potiguar:6:2:0:0:07:03:+04 2:Baraunas:3:1:0:1:06:06:+00 3:Campinense:0:0:0:2:05:09:-04</pre>