

# Instituto Federal de Educação Ciência e Tecnologia do RN - IFRN

## Exercícios

- 001** Escreva um programa que leia dois números inteiros e informe se os mesmo são iguais ou não. Caso os números sejam iguais o programa deve mostrar a palavra **SIM**. Caso os números sejam diferentes o programa deve mostrar a palavra **NAO**. Abaixo segue os exemplos de execução, com os arquivos de entrada e saída.

Exemplo de execução	Entrada (entrada.txt)	Saída (saida.txt)
\$ ruby iguais.rb 123 123 SIM \$	123 123	SIM

Exemplo de execução	Entrada (entrada.txt)	Saída (saida.txt)
\$ ruby iguais.rb 20 21 NAO \$	20 21	NAO

- 002** Escreva um programa que leia dois números inteiros e mostre a diferença entre o maior e o menor (*maior – menor*). O programa deve mostrar **APENAS** o número, nada mais.
- 003** Escreva um programa que leia três números inteiros e mostre o maior 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 mostra 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 a média e a situação do aluno. 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 média do aluno, com um casa decimal, seguida de dois pontos (:), com a situação atual, que pode ser: **APROVADO**, **RECUPERACAO** ou **REPROVADO**. O aluno é aprovado se sua média for maior-igual a 6.0. Ele é reprovado se sua média for menor do que 2.0. Caso, pelo menos, uma das notas não seja válida o programa não mostra a média, deve apenas mostrar a palavra **INVALIDA**.

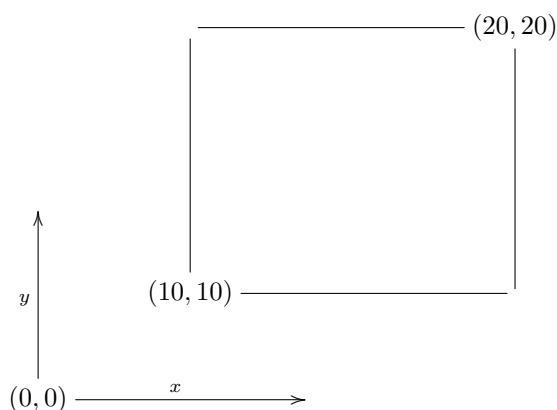
Exemplo de saída:

7.1: APROVADO

- 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.
- 007** Escreva um programa que leia 3 números e verifique se a soma dos dois menores é maior do que o maior. O programa deve mostrar **MAIOR** caso positivo ou **MENOR** caso negativo.

Exemplo de execução	Entrada (entrada.txt)	Saída (saida.txt)
\$ ruby somamaior.rb 10 18 15 MAIOR \$	10 18 15	MAIOR

- 008** Escreva um programa que leia 3 números e verifique se o menor é divisor dos dois outros números. O programa deve mostrar a palavra **DIVISOR** caso o menor seja divisor dos dois. Nada mais deverá ser mostrado pelo programa.
- 009** 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.
- 010** Escreva um programa que leia 3 números inteiros e informe se a soma dos dois menores é maior do que o maior. o programa deve mostrar a palavra **maior** (toda em minúscula) se a soma dos dois menores for maior do que o maior número. Nada deve ser mostrado se a soma não for maior.
- 011** Escreva um programa que leia 3 números, verifique se eles formam os lados de um triangulo e mostre o tipo de triangulo: 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 números inteiros, que representam dois pontos no plano cartesiano (mostrado na figura abaixo) e informe se é possível formar um quadrado ou um retângulo com esses pontos. Caso seja possível formar um quadrado o programa deve mostrar a palavra **Quadrado**, senão o programa deve mostrar a palavra **Retangulo** (sem acentos).



- 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 sequencia) 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 uma data, representada com 3 números inteiros: dia, mês e ano. Em seguida o programa deve ler um número inteiro, que corresponde a uma quantidade de dias, e deve mostrar uma data calculada somando-se a data de entrada com a quantidade de dias.
- Se a entrada for um número negativo o programa não mostra nada.*
- 017** Escreva um programa que leia 6 números e mostre a maior diferença entre eles.
- Exemplo: Se os números digitados forem 1, 15, 3, 8, 9, 10 a maior diferença é 14 (15 – 1).*
- 018** Escreva um programa para calcular a situação final de um aluno, da seguinte forma: primeiro o programa deve ler 2 notas, que correspondem a nota da prova e do trabalho do primeiro bimestre, em seguida o programa deve ler dois números inteiros, onde o primeiro é o peso da prova e o segundo o peso do trabalho. Com esses valores o programa deve calcular a nota do primeiro bimestre. Em seguida o programa deve ler mais 2 notas e dois

pesos, que correspondem as notas da prova e do trabalho do segundo bimestre e seus respectivos pesos, e calcular a nota do segundo bimestre. De posse das duas notas o programa deve calcular a média parcial do aluno, de acordo com a fórmula da média parcial do IF.

O programa deve verificar se o aluno está em recuperação e, caso esteja, deve ler a nota da recuperação e calcular a média final. Ao final o programa deve mostrar a nota do primeiro bimestre, a nota do segundo bimestre, a nota da recuperação, caso exista, a média final e a situação: **APROVADO** ou **REPROVADO**. Se o aluno não passou pela recuperação, o programa deve mostrar um hífen (-) para a nota da recuperação. As notas devem ser mostradas com uma casa decimal e todos os valores devem ser separados por dois pontos (:), como nos exemplos abaixo:

1. Exemplo de aprovado sem recuperação

6.0:6.0:--6.0:APROVADO

2. Exemplo de reprovado com recuperação

2.0:2.0:2.0:2.0:REPROVADO

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 uma casa decimal*

- 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 em dois dígitos, 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>