Noções iniciais de programação

Lista de exercícios

1. Escreva um programa em Ruby que receba um valor inteiro do usuário, some a constante 5 ao valor recebido e mostre na tela.
   Exemplo de resposta:
   Entrada: 3
   Saída:
   Recebi: 3
   Resultado: 8

2. Escreva um programa em Ruby que leia uma string do usuário e cheque se ela contém apenas caracteres ASCII.
   Exemplos:
   Entrada: tenho 20 anos
   Saída: Contém somente caracteres ASCII
   Entrada 2: meu nome é Graça
   Saída 2: Contém caracteres não ASCII

3. Escreva um programa em Ruby com uma variável para cada um dos tipos citados nos slides (integer, float, string, array, symbol e hash) e utilize o método .class para imprimir na tela o conteúdo que comprove isso.
   Saída:
   73 -> integer
   5.883 -> float
   trainee -> string
   ...

4. Deve-se criar as variáveis **nome**, **idade**, **peso** e **altura**. Armazene dentro dessas variáveis os valores referentes às suas informações pessoais respeitando o tipo que faz sentido (string, integer ou float). Tendo as variáveis e seus respectivos conteúdos, concatenar tudo formando a seguinte frase:

"Meu nome é **nome**, tenho **idade** anos, peso **peso** kg e minha altura é **altura** m"

5. Escreva um programa em Ruby que, dada a nota de um aluno, imprima a menção dele de acordo com as regras da UnB. (http://www.dan.unb.br/images/pdf/graduacao/documentos_uteis/porcentagemdefrequncia.pdf)
   Exemplo:
   Entrada: 4.7
   Saída: MI

6. Faça um programa que imprima todos os números pares de 1 até 100.
   Saída:
   2, 4, 6, 8, 10, …, 100

7. Desenvolva um programa que receba do usuário um número entre 1 e 10 e escreva a tabuada do valor lido. O programa deve pedir para o usuário inserir um novo número se a entrada não estiver no intervalo entre 1 e 10 quantas vezes for preciso.
   Exemplo:
   Entrada: 5
   Saída: 5, 10, 15, …, 50

8. Escreva uma função que diz se o parâmetro de entrada é par ou ímpar.
   Exemplo:
   Entrada: 5
   Saída: 5 é Ímpar

9. Utilizando a função criada no exercício anterior, escreva um programa que receba 5 valores de entrada e imprima quantos deles são pares e quantos são ímpares.
   Exemplo:
   Entrada:
   (5 números enviados um a um)
   1
   3
   5
   7
   9
   Saída: 7 são ímpares, 3, pares

10. Utilizando a estrutura de repetição for, faça um programa em Ruby que receba 10 números e conte quantos deles estão no intervalo [10,20] e quantos deles estão fora do intervalo, escrevendo estas informações.
    Exemplo:
    Entrada: 1 3 5 7 9 11 12 13 18 20 (10 números separados por espaço)
    Saída: 5 estão no intervalo [10,20], 5, fora

11. Defina um array com o nome de 5 alunos e utilize algum método para imprimir aleatoriamente o nome de um dos alunos.

12. Escreva um código em Ruby que gere um array com todos os anos do século 20 e coloque em outro array apenas os que forem anos bissextos (recomendamos fazer uma função que verifica se é bissexto). Por fim, imprima esse array.
    Saída: [1904, 1908, 1912, …,1996]

13. Tendo como entrada o hash **aluno1 = {"nome" => "Pedro", "idade" => 20}**, imprima na tela apenas o nome do aluno1.
    Saída: Pedro

14. Ainda sobre o hash do exercício 4, crie uma nova chave **altura** com o valor **1.72**. Imprima o hash final gerado.
    Saída: {"nome" => "Pedro", "idade" => 20, “altura” => 1.72}

15. Crie um hash vazio para armazenar as informações de um aluno. Após criar o hash vazio, peça para o usuário digitar o **nome**, **idade**, **matrícula** e **e-mail**. Guarde essas informações no hash aluno com as chaves citadas em negrito e imprima o resultado gerado.
    Entrada:
    Roberto
    18
    180125729
    roberto@struct.unb.br
    Saída: {"nome"=>"Roberto", "idade"=>18, "matricula"=>"180125729", "email"=>"roberto@struct.unb.br"}

16. Um dos maiores dilemas da computação é a dicotomia entre memória e processamento: devo ocupar minha base de dados com dados já processados (relações de porcentagem, por exemplo), ou devo processá-los toda vez que eles forem necessários? Pensando nisso, imagine que você está programando uma aplicação que gerenciará uma base de dados de DNA. Sabemos que DNA são duplas hélices ligadas pela interação entre seus nucleotídeos. Temos quatro tipos de nucleotídeos no DNA: adenina (A), timina (T), citosina (C) e guanina (G). Os nucleotídeos A só podem realizar ligações com nucleotídeos T, e os nucleotídeos C, com G. Dito isso, faça um programa que gere uma fita de DNA com 10 nucleotídeos aleatórios, para, depois de montada a primeira fita, gerar a segunda fita que completa esse DNA. Explicite na tela a primeira fita, gerada aleatoriamente, e depois a segunda fita, com as respectivas ligações.
Exemplo de resposta:
Primeira fita:
[A, T, C, A, T, G, C, T, C, G]
DNA:
[A, T]
[T, A]
[C, G]
[A, T]
[T, A]
[G, C]
[C, G]
[T, A]
[C, G]
[G, C]

17. Você foi contratado para programar um caixa eletrônico. Receba um valor inteiro inserido pelo usuário e responda quantas notas de 100, 50, 20, 10, 5 e 2 são necessárias para que o valor seja sacado. Indique se houver resto.
Exemplo:
Entrada: 193
Saída:
1 nota(s) de 100
1 nota(s) de 50
2 nota(s) de 20
0 nota(s) de 10
0 nota(s) de 5
1 nota(s) de 2
Resto: 1

Desafio 1 (Não obrigatório)

- Um restaurante tem seu cardápio organizado em um hash. O hash consiste em três chaves, “entrada”, “principal” e “sobremesa”. Você deverá colocar de 3 a 5 pratos vinculados a cada uma dessas chaves (array de pratos). Escreva um programa que retorne um array com um prato selecionado aleatoriamente de cada chave e os imprima na tela.
  {"entrada"=>["Bruschetta", "Salada","Sopa"], "principal"=> …}

  Coloquei acima o começo de como ficará uma possível versão do seu hash. Visto que temos um hash com as chaves **entrada**, **principal** e **sobremesa**. O resultado esperado no array gerado será um prato de cada uma dessas chaves, selecionado aleatoriamente. Exemplo de resultado:
  ["Bruschetta", "Bobó de Camarão", "Torta de Limão"]

Desafio 2 (Não obrigatório)

- No mesmo restaurante maluco do exercício anterior, um garçom que trabalha com dev sugeriu que os pratos deveriam ter o preço no cardápio. Ele disse: “isso não é uma EJ, não da pra trabalhar sem ganhar dinheiro”. Para isso, foi sugerido que o valor de cada chave do hash maior (**entrada**, **principal**, **sobremesa**), fosse transformado em um hash organizado da seguinte forma:
  {“Bruschetta” => 2.5, “Salada” => 3, “Sopa” => 5}
  Logo, o cardápio ficaria da seguinte maneira:
  {"entrada"=>{"Bruschetta" => 2.5, "Salada" => 3, "Sopa" => 5}, "principal"=> …}

  Com essa mudança, o restaurante poderia finalmente cobrar seus clientes! Faça então um programa que não só gere e imprima na tela um array com um prato aleatório de cada categoria, mas também que calcula a conta que deverá ser paga.

  Exemplo de resultado:
  ["Bruschetta", "Bobó de Camarão", "Torta de Limão"]. Total: R$45.00.