

Atividade Avaliativa

Disciplina: Fundamentos de Programação

Data de Entrega: 29/04/2021

Professor: Marcos Roberto dos Santos

Avaliação Bimestral

(2,0) #obrigatório #optativo
(3,5) Atividade Avaliativa
(0,5) Participação
(4) Prova Teórica

Descrição:

O aluno deverá enviar os aplicativos através do Ambiente Virtual Acadêmico – AVA.

Objetivo:

Aplicar todas as metodologias vistas em sala de aula, melhorando cada algoritmo para obter respostas adequadas e dados mais pontuais para as soluções apresentadas.

Analise as questões e desenvolva os algoritmos abaixo, escolhendo ao mínimo duas tags (Ex.: 1 Hard e um Medium, ou 2 Easy e 1 Medium), totalizando 3,5 pontos.

✓ Very Easy 0.5 pontos

✓ Easy – 1 ponto

✓ Medium – 1.5 pontos

✓ Hard – 2 pontos

- 1) [very easy] Leia uma quantidade indeterminada de duplas de valores inteiros X e Y. Escreva para cada X e Y uma mensagem que indique se estes valores foram digitados em ordem crescente ou decrescente. A entrada contém vários casos de teste. Cada caso contém dois valores inteiros X e Y. A leitura deve ser encerrada ao ser fornecido valores iguais para X e Y. Para cada caso de teste imprima “Crescente”, caso os valores tenham sido digitados na ordem crescente, caso contrário imprima a mensagem “Decrescente”. *[abaixo um exemplo de saída do code]*

Exemplo de Entrada	Exemplo de Saída
5 4	Decrescente
7 2	Decrescente
3 8	Crescente
2 2	

- 2) [very easy] Leia um número inteiro que representa um código de DDD para discagem interurbana. Em seguida, informe à qual cidade o DDD pertence, considerando a tabela abaixo. Se a entrada for qualquer outro DDD que não esteja presente na tabela acima, o programa deverá informar: DDD não cadastrado

DDD	Destination
61	Brasília
71	Salvador
11	Sao Paulo
21	Rio de Janeiro
32	Juiz de Fora
19	Campinas
27	Vitoria
31	Belo Horizonte

[abaixo um exemplo de saída do code]

Exemplo de Entrada	Exemplo de Saída
11	Sao Paulo

- 3) [easy] Leia 4 valores inteiros A, B, C e D. Analise após, se B for maior do que C e se D for maior do que A, e a soma de C com D for maior que a soma de A e B escrever a mensagem "Valores aceitos", senão escrever "Valores não aceitos". [abaixo um exemplo de saída do code]

```
Informe o valor de A: 1
Informe o valor de B: 2
Informe o valor de C: 3
Informe o valor de D: 4
Valores Não Aceitos
```

- 4) [easy] Utilizando o comando 'for', leia 5 valores Inteiros. A seguir mostre quantos valores digitados foram pares, quantos valores digitados foram ímpares, quantos valores digitados foram positivos e quantos valores digitados foram negativos. [abaixo um exemplo de saída do code]

Exemplo de Entrada	Exemplo de Saída
-5	3 valor(es) par(es)
0	2 valor(es) impar(es)
-3	1 valor(es) positivo(s)
-4	3 valor(es) negativo(s)
12	

- 5) [easy] Você deve fazer um programa que leia um valor qualquer e apresente uma mensagem dizendo em qual dos seguintes intervalos ([0,25], (25,50], (50,75], (75,100]) este valor se encontra. Obviamente se o valor não estiver em nenhum destes intervalos, deverá ser impressa a mensagem "Fora de intervalo". O símbolo (representa "maior que". Por exemplo: [0,25] indica valores entre 0 e 25, inclusive ele. (25,50] indica valores maiores que 25 Ex: 25.00001 até o valor 50. [abaixo um exemplo de saída do code]

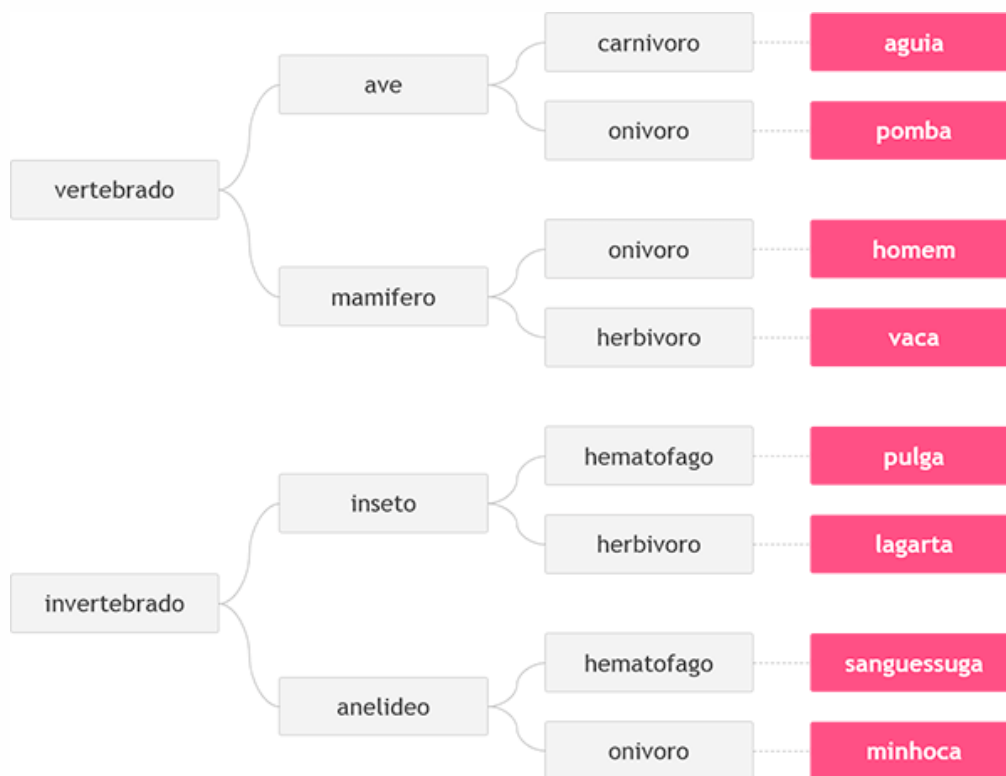
Exemplo de Entrada	Exemplo de Saída
25.01	Intervalo (25,50]
25.00	Intervalo [0,25]
100.00	Intervalo (75,100]
-25.02	Fora de intervalo

- 6) [medium] Desenvolva um algoritmo com a capacidade de receber a KM de um veículo e a quantidade de litros de gasolina que o mesmo consumiu. Após deve ser calculado, indicando a média de consumo do veículo e o nível de consumo conforme tabela abaixo (Km/l). [1.5]

Indicação de Consumo de Veículos

A	Mais de 12 Km/l
B	Até 12 Km/l
C	Até 10 Km/l
D	Até 8 Km/l
E	Até 4 Km/l

- 7) [medium] Neste problema, você deverá ler 3 palavras que definem o tipo de animal possível segundo o esquema abaixo, da esquerda para a direita. Em seguida conclua qual dos animais seguintes foi escolhido, através das três palavras fornecidas. [abaixo um exemplo de saída do code]



Exemplos de Entrada	Exemplos de Saída
vertebrado mamifero onivoro	homem
vertebrado ave carnivoro	aguia
invertebrado anelideo onivoro	minhoca


- 8) [medium] Crie um algoritmo que receba a velocidade do carro e analise se o mesmo deve ser multado e valor que ele deve pagar. [abaixo um exemplo de saída do code]
- Se for menor que 80 não haverá multa.
 - Se estiver entre 80 e 95, a multa é infração leve e o valor a ser pago é de R\$ 67,00
 - Se estiver acima de 95, a multa é infração grave e o valor a ser pago é de R\$ 187,00

```
Informe a velocidade do carro em KM/H: 91
Infração Leve, R$: 67,00
```

- 9) **[hard]** Implemente o seguinte algoritmo:

URI Online Judge | 1160

Crescimento Populacional

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Mariazinha quer resolver um problema interessante. Dadas as informações de população e a taxa de crescimento de duas cidades quaisquer (A e B), ela gostaria de saber quantos anos levará para que a cidade menor (sempre é a cidade A) ultrapasse a cidade B em população. Claro que ela quer saber apenas para as cidades cuja taxa de crescimento da cidade A é maior do que a taxa de crescimento da cidade B, portanto, previamente já separou para você apenas os casos de teste que tem a taxa de crescimento maior para a cidade A. Sua tarefa é construir um programa que apresente o tempo em anos para cada caso de teste.

Porém, em alguns casos o tempo pode ser muito grande, e Mariazinha não se interessa em saber exatamente o tempo para estes casos. Basta que você informe, nesta situação, a mensagem "Mais de 1 século."

Entrada

A primeira linha da entrada contém um único inteiro **T**, indicando o número de casos de teste ($1 \leq T \leq 3000$). Cada caso de teste contém 4 números: dois inteiros **PA** e **PB** ($100 \leq PA \leq 1000000$, $PA < PB \leq 1000000$) indicando respectivamente a população de A e B, e dois valores **G1** e **G2** ($0.1 \leq G1 \leq 10.0$, $0.0 \leq G2 \leq 10.0$, $G2 < G1$) com um dígito após o ponto decimal cada, indicando respectivamente o crescimento populacional de A e B (em percentual).

Atenção: A população é sempre um valor inteiro, portanto, um crescimento de 2.5 % sobre uma população de 100 pessoas resultará em 102 pessoas, e não 102.5 pessoas, enquanto um crescimento de 2.5% sobre uma população de 1000 pessoas resultará em 1025 pessoas. Além disso, não utilize variáveis de precisão simples para as taxas de crescimento.

Saída

Imprima, para cada caso de teste, quantos anos levará para que a cidade A ultrapasse a cidade B em número de habitantes. Obs.: se o tempo for mais do que 100 anos o programa deve apresentar a mensagem: Mais de 1 século. Neste caso, acredito que seja melhor interromper o programa imediatamente após passar de 100 anos, caso contrário você poderá receber como resposta da submissão deste problema "Time Limit Exceeded".

Exemplo de Entrada	Exemplo de Saída
6	51 anos.
100 150 1.0 0	16 anos.
90000 120000 5.5 3.5	12 anos.
56700 72000 5.2 3.0	Mais de 1 século.
123 2000 3.0 2.0	10 anos.
100000 110000 1.5 0.5	100 anos.
62422 484317 3.1 1.0	

- 10) **[hard]** Paulo e Pedro fizeram uma longa jornada desde que partiram do Brasil para competir na Final Mundial da Maratona, em Phuket, Tailândia. Notaram que a cada escala que faziam, tinham que ajustar seus relógios por causa do fuso horário. Assim, para melhor se organizarem para as próximas viagens, eles pediram que você faça um aplicativo que, dada a hora de saída, tempo de viagem e o fuso do destino com relação à origem, você informe a hora de chegada de cada vôo no destino. Por exemplo, se eles partiram às 10 horas da manhã para uma viagem de 4 horas rumo a um destino que fica à leste, em um fuso horário com uma hora a mais com relação ao fuso horário do ponto de partida, a hora de chegada terá que ser: 10 horas + 4 horas de viagem + 1 hora de deslocamento pelo fuso, ou seja, chegarão às 15 horas. Note que se a hora calculada for igual a 24, seu programa deverá imprimir 0 (zero). As entradas deverão indicar respectivamente a hora da saída, o tempo de viagem e o fuso horário do destino com relação à origem. *[abaixo um exemplo de saída do code]*

Exemplos de Entrada	Exemplos de Saída
10 7 3	20
22 6 -2	2
0 3 -4	23

- 11) [very easy] Implemente um aplicativo ao qual só possa ser acessado se o usuário responder 3 questões matemáticas (Ex.: $1+5$, $6*6$, $900/10$), e caso acerte as 3, ele será encaminhado a uma mensagem de boas vindas. Caso contrário, deve ser informado que ele errou e não pode acessar o sistema, solicitado novamente as questões até que ele acerte.