


Nome –

Assinatura –

1ª questão (valor 2.0)

Escreva um algoritmo em **PETEQS** que permita a um grupo de quatro sapos andando para a esquerda em um caminho muito estreito (R1, R2, R3) passar por outro grupo de sapos andando para a direita (L1, L2, L3).

						
L3	L2	L1		R1	R2	R3

Os sapos não andam para trás e podem pular uns sobre os outros, um sapo por vez, se houver um espaço vazio do outro lado. Dois sapos não podem ocupar o mesmo espaço ao mesmo tempo.

Os seguintes comandos estão disponíveis:

.anda ()

Faz com que o sapo avance uma casa em frente. O comando só é bem sucedido se a posição destino estiver vazia. Exemplo: **L1.anda ()**

.pula ()







Faz com que o sapo salte sobre a casa em frente. O comando só é bem sucedido se a posição destino estiver vazia. Exemplo: **R1.sal ta ()**

Exemplo:

A partir da configuração inicial, o comando **L1.anda ()** levaria os sapos à seguinte situação:

						
L3	L2		L1	R1	R2	R3

Em seguida, o comando **R1.pula ()** levaria os sapos à nova configuração:

						
L3	L2	R1	L1		R2	R3

Sua tarefa:

Escreva o algoritmo que permita aos sapos andando em direção oposta passarem uns pelos outros, respeitando as regras dos movimentos permitidos.

Sugestão: Um tabuleiro de damas ou xadrez lhe ajudaria a visualizar a solução.

2ª questão (valor 2.0)

Escreva um algoritmo em **PETEQS** que determine se um círculo, especificado pelas coordenadas do centro (x_1, y_1) e pelo raio r_1 , intercepta ou colide com outro círculo com centro em (x_2, y_2) e raio r_2

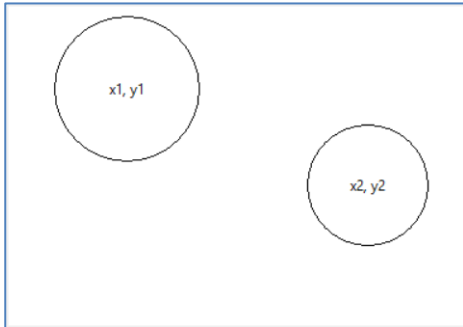


Figura 1- Círculos sem colisão

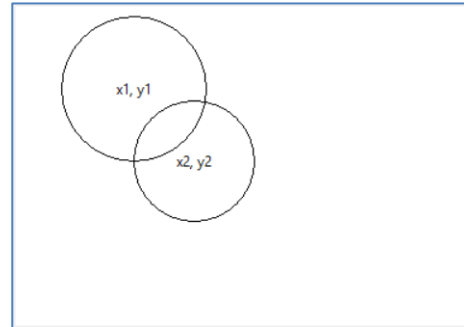


Figura 2 - Círculos com colisão

Os parâmetros $x_1, y_1, r_1, x_2, y_2, r_2$ são fornecidos nesta ordem pelo usuário ao algoritmo e lidos com o comando **leia**. O programa deve imprimir **0** quando os círculos não colidirem e **1** em caso contrário.

Exemplos:

entradas:

100 70 60 150 130 50

saída

1

entradas:

100 70 60 300 150 50

saída

0

3ª questão (valor 1.0)

Converta a sequência de códigos ASCII abaixo em texto.

```
84 74 68 32 100 101 110 117 110 99 105 97 32 111 32 70 108 117
109 105 110 101 110 115 101 32 112 111 114 32 103 114 105 116
111 115 32 100 101 32 39 116 105 109 101 32 97 115 115 97 115
115 105 110 111 39 32 99 111 110 116 114 97 32 111 32 70 108 97
109 101 110 103 111
```

Você pode usar como referência a tabela ASCII em <https://br.ccm.net/contents/54-o-codigo-ascii>

4ª questão (valor 1.25)

Escreva um algoritmo em **PETEQS** que calcule quantas horas, minutos e segundos existem em um tempo lido em segundos. Por exemplo, se o usuário fornecer o valor 3754 o algoritmo deve imprimir os valores 1, 2 e 34 indicando que em 3754 segundos há 1 hora, 2 minutos e 34 segundos. Outro exemplo, se o usuário fornecer o valor 453 o algoritmo deve imprimir os valores 0, 7 e 33 indicando que em 453 há 0 horas, 7 minutos e 33 segundos.

5ª questão (valor 1.25)

Escreva um algoritmo em **PETEQS** que leia três números inteiros e imprima quantos dos três números são ímpares. O seu algoritmo não pode usar na solução desvios condicionais. Dica: use o comando **mod**.

6ª questão (valor 1.25)

Um funcionário de uma empresa recebeu três aumentos de salário ao longo de sua carreira. Os aumentos foram, nesta ordem, 10%, 5% e 2%. Escreva um algoritmo em **PETEQS** que leia o salário inicial do funcionário e calcule o seu salário após os três aumentos. Por exemplo, se o salário inicial do funcionário é 1000.00 o salário atual é 1178.10.

7ª questão (valor 1.25)

Indique os nomes de variáveis válidos e os inválidos em **PETEQS**. Justifique suas respostas.

- (a) **salarioInicial**
- (b) **salario_inicial**
- (c) **juros\$**
- (d) **lnota**
- (e) **nota1**