

**Curso:** Ciência de Computação

**Data:** 25./05./2022

**Prova:** P1

**Período:** 3º

**Disciplina:** Estrutura de dados I

**Professor:** Fermín Alfredo Tang

**Turno:** Diurno

**Nome do Aluno:** ..... **Matrícula:** .....

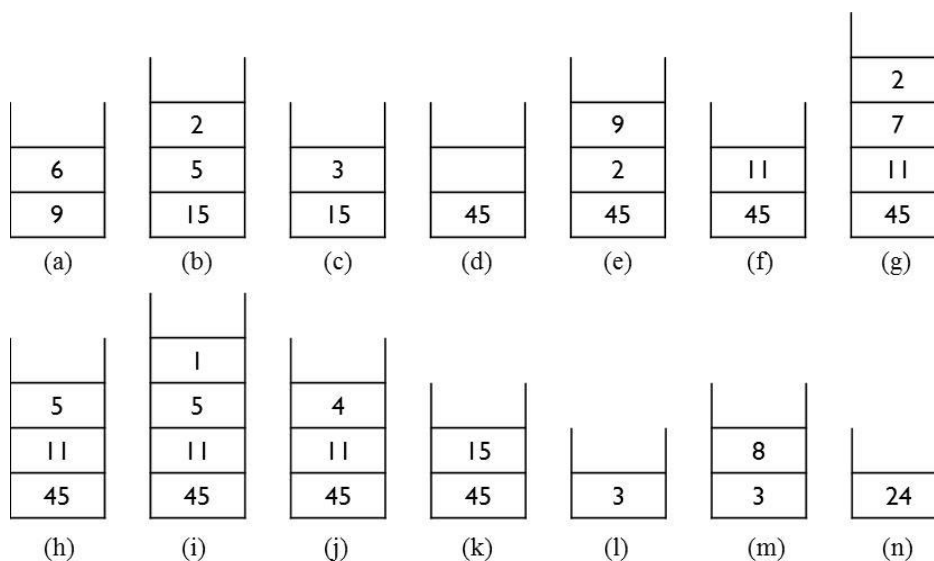
1. [2,0 Pontos] Dada a seguinte expressão pós-fixa:

$$AB + CD - \times EF + GH - I - + / J \times$$

onde:  $A = 9$ ,  $B = 6$ ,  $C = 5$ ,  $D = 2$ ,  $E = 2$ ,  $F = 9$ ,  $G = 7$ ,  $H = 2$ ,  $I = 1$ ,  $J = 8$ .

Ilustre passo a passo o processo de cálculo do valor da expressão, usando uma **pilha** para armazenar os operandos assim como os resultados parciais até o resultado final.

**Resposta 1.-** Cada vez que um operando é lido ele é armazenado na pilha. Quando um operador é lido os dois últimos operandos são removidos da pilha e a operação é realizada, o resultado é armazenado na pilha. O uso da pilha é o seguinte:



O resultado final da expressão fica armazenado na pilha. O resultado é 24.

2. [2,0 Pontos] Desenvolva um algoritmo para testar se duas pilhas P1 e P2 são iguais. Duas pilhas são iguais se possuem os mesmos elementos, exatamente na mesma ordem. Considere que as pilhas P1 e P2 já existem e possuem elementos do tipo inteiro. As pilhas precisam ser preservadas após o teste.

a) Descreva a ideia do algoritmo e as operações necessárias. Considere os casos possíveis. Ilustre a situação das pilhas P1 e P2, e a pilhas auxiliares usando exemplos. [1,0 ponto]

b) Escreva o seu algoritmo com base em operações básicas de pilha. [1,0 ponto]

## Resposta 2.-

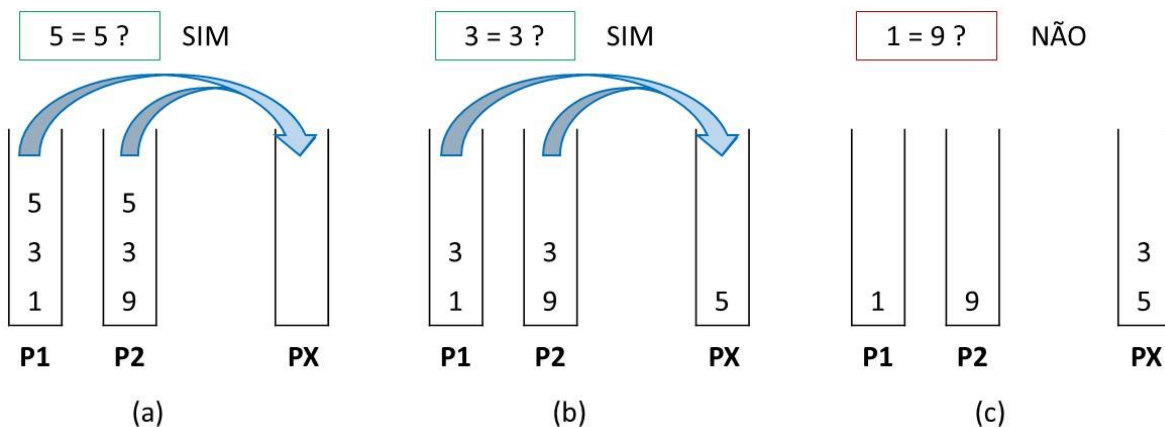
a) Tratando-se de pilhas, a única maneira de acessar os seus elementos é extrair o elemento do topo. Como as pilhas precisam ser preservadas, os elementos extraídos precisam ser armazenados temporariamente em uma pilha auxiliar PX, para poder reconstruir as pilhas originais. Assim, os passos do algoritmo para testar se duas pilhas P1 e P2 são iguais seriam:

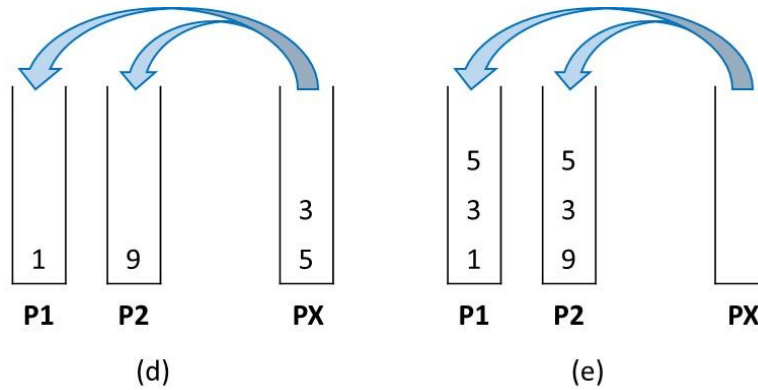
- Verificar se ambas as pilhas possuem o mesmo número de elementos; Caso o número seja diferente, as pilhas não serão iguais. Caso contrário, continuar.
- Examinar o conteúdo de ambas as pilhas P1 e P2, extraíndo um a um seus elementos, mediante um loop; Comparar ambos elementos.
- Repetir o loop enquanto a pilha P1 não estiver vazia (P2 também não estará), e os elementos extraídos forem iguais. Caso os elementos sejam diferentes, as pilhas não serão iguais.
- Caso os elementos sejam iguais, inserir o elemento topo na pilha auxiliar PX.
- Em qualquer caso, para restaurar o conteúdo das pilhas P1 e P2, extrair um a um os elementos da pilha auxiliar PX, e inserir nas pilhas P1 e P2.

Os casos possíveis seriam:

- Duas pilhas com diferente número de elementos;
- Duas pilhas com igual número de elementos, mas com conteúdos diferentes, seja pelo valor dos elementos ou pela ordem em que os elementos aparecem;
- Duas pilhas iguais.

A figura abaixo ilustra o procedimento.





b) Um algoritmo para a operação de comparação seria o seguinte:

```

PX = CreateStack(PX);
pilhas_iguais = verdadeiro
se ( stackcount(P1) != stackcount(P2) ) então
    pilhas_iguais = falso
senão
    elem1 = StackTop(P1)
    elem2 = StackTop(P2)
    enquanto não(emptyStack(P1)) e (elem1 == elem2) fazer
        popStack(P1);
        popStack(P2);
        pushStack(PX, elem1);
        se não(emptyStack(P1)) então
            elem1 = StackTop(P1)
            elem2 = StackTop(P2)
        fim-se
    fim-fazer
    se (elem1 != elem2) então
        pilhas_iguais = falso
    fim-se

    enquanto não(emptyStack(PX)) fazer
        elemx = popStack(PX);
        pushStack(P1, elemx);
        pushStack(P2, elemx);
    fim-fazer
retorna pilhas_iguais

```

3. [2,0 Pontos] Qual seria o conteúdo das filas Q1 e Q2 após o seguinte código ser executado:

```

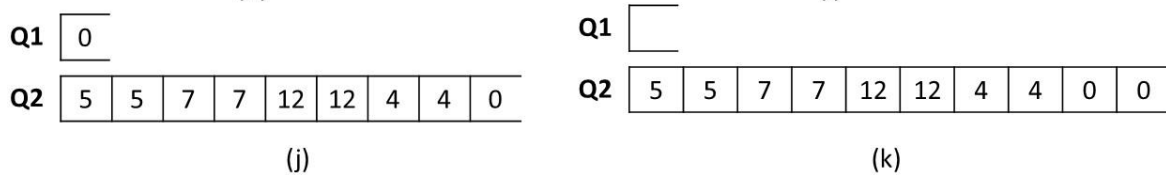
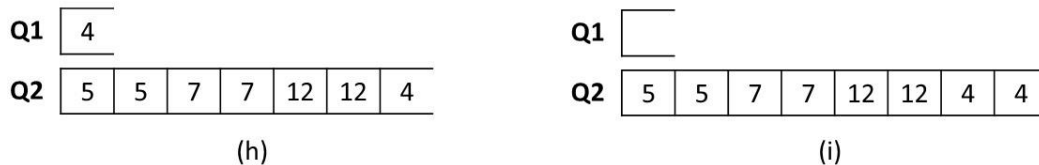
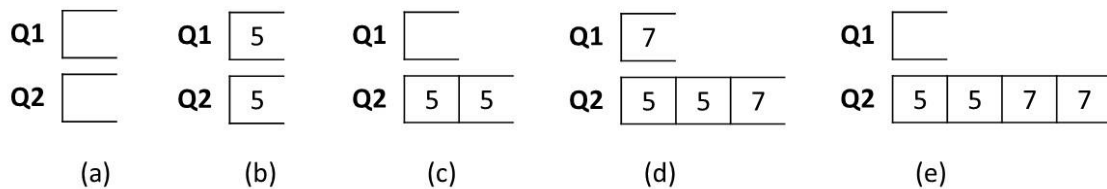
1  q1 = createqueue
2  q2 = createqueue
3  loop (not end of file)
  1  read number
  2  enqueue (q1, number)
  3  enqueue (q2, number)
  4  loop (not empty q1)
    1  dequeue (q1, x)
    2  enqueue (q2, x)
  5  end loop
4  end loop

```

Considerando que a seguinte sequência de dados é ingressada:

5, 7, 12, 4, 0, 4, 6.

**Resposta 3.-** Inicialmente duas filas Q1 e Q2 são criadas. O loop externo faz a leitura dos números de um arquivo. Após lido, o número é inserido em ambas as filas. O loop interno se repete enquanto a fila Q1 não estiver vazia, removendo um elemento da fila Q1, colocando em x e inserindo x na fila Q2.



Q1	4										
Q2	5	5	7	7	12	12	4	4	0	0	4

(l)

Q1	<div></div>											
Q2	5	5	7	7	12	12	4	4	0	0	4	4

(m)

Q1	6												
Q2	5	5	7	7	12	12	4	4	0	0	4	4	6

(n)

Q1	<div></div>													
Q2	5	5	7	7	12	12	4	4	0	0	4	4	6	6

(o)

4.- [2,0 Pontos] Considere que temos as duas listas encadeadas como mostradas na Figura 1, onde list1 e list2 são os ponteiros ao primeiro elemento de cada lista, respectivamente.

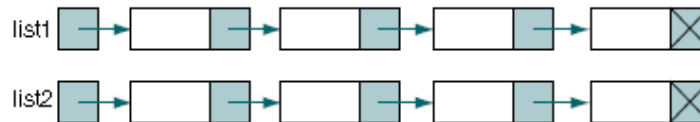


Figura 1.

i) Explique o que aconteceria com a lista 1, se aplicamos a seguinte sentença?:

```
list1 = list2
```

ii) Em cada nó, o campo link contém o ponteiro ao seguinte elemento. O acesso a esse ponteiro é dado pela sentença:  $p \rightarrow \text{link}$ .



Explique o que aconteceria se ao percorrer a lista 1, aplicamos a seguinte sentença dentro de um loop?:

```
list1 = list1 -> link
```

**Resposta 4.-**

i) A sentença indicada:

```
list1 = list2
```

consiste na atribuição de valor do ponteiro list2, no ponteiro list1. Como o ponteiro list2 aponta para o primeiro elemento da segunda lista encadeada, após se fazer essa atribuição, temos que list1 também aponta para o primeiro elemento da segunda lista.

No entanto, fazer essa atribuição significa que perderemos acesso aos elementos da primeira lista, já que o valor do ponteiro list1 é perdido.

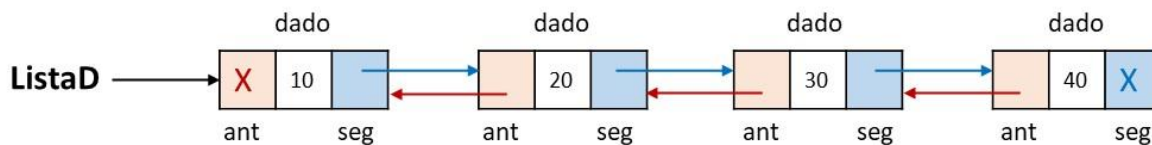
ii) Se ao percorrer a lista 1, colocamos a sentença: `list1 = list1 -> link`

dentro de um loop. Esse loop vai permitir que de fato podamos percorrer todos os elementos da lista, parando quando `list1 == NULL`. No entanto, como a variável usada para percorrer a lista é list1, que é o ponteiro ao primeiro elemento da lista, o acesso a lista será perdido.

5.- [2,0 Ponto] Explique a diferença entre um lista duplamente encadeada e uma lista multi-encadeada. Ilustre cada caso com um exemplo. No caso da lista muti-encadeada considere apenas dois campos de encadeamento.

### Resposta 5.-

Em uma lista duplamente encadeada, cada elemento (chamado de nó) possui dois campos de ligação, que são: i) um ponteiro para o elemento seguinte e ii) um ponteiro para o elemento anterior. Dessa forma os elementos da lista duplamente encadeada podem ser percorridos em duas direções para frente (*forward*) e para trás (*backward*).



Por outro lado, no caso de uma lista multi-encadeada, cada elemento (chamado de nó) possui dois ou mais campos de ligação. A diferença neste caso é que cada campo de ligação é usado para realizar um tipo de encadeamento simples entre os elementos da lista. Sendo assim, o mesmo conjunto de elementos pode ser encadeado em duas ou mais formas diferentes. Para isso, o conjunto de dados deve possuir tantas dimensões como encadeamentos desejados. A lista multi-encadeada funciona como se fossem várias listas encadeadas simples em uma só. No exemplo ilustra-se uma lista multi-encadeada com dois encadeamentos, um numérico e outro alfabético.

