

UENF

Universidade Estadual do Norte Fluminense Darcy Ribeiro

Curso: Ciência de Computação

Data: 15/04/2021

Atividade: Questões Módulo 2 – AM2

Período: 3º

Disciplina: Estrutura de Dados I

Professor: Fermín Alfredo Tang

Turno: Diurno

Nome do aluno:

Matrícula:

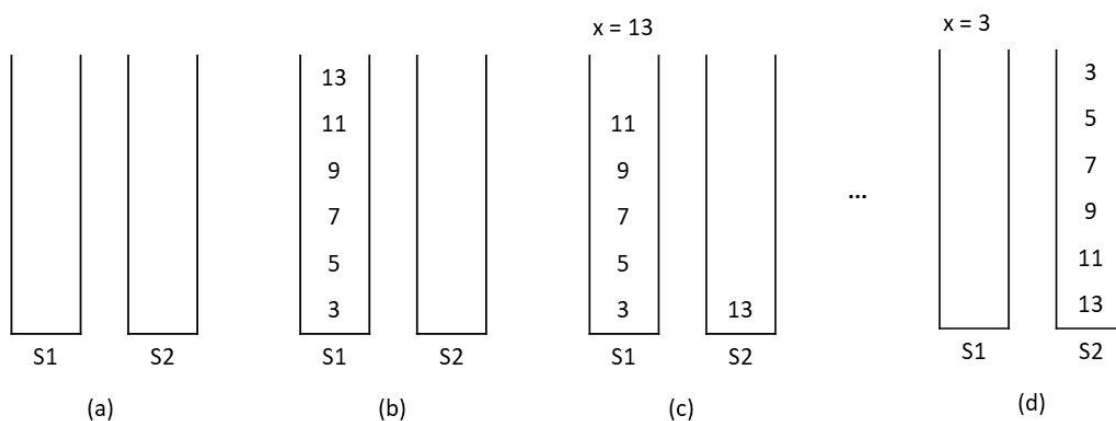
Questões para AM2

Considerando os conceitos sobre Pilhas:

1. Considere que temos duas pilhas de inteiros vazias, s1 e s2, mostre em uma figura a evolução de cada pilha após as seguintes operações:

```
1 pushStack (s1, 3)
2 pushStack (s1, 5)
3 pushStack (s1, 7)
4 pushStack (s1, 9)
5 pushStack (s1, 11)
6 pushStack (s1, 13)
7 loop not emptyStack (s1)
  1 popStack (s1, x)
  2 pushStack (s2, x)
8 end loop
```

Resposta1.- Temos inicialmente duas pilhas vazias, s1 e s2.



Sabemos que a operação pushStack() corresponde a inserção de um elemento na pilha e a operação popStack() corresponde a remoção de um elemento da pilha.

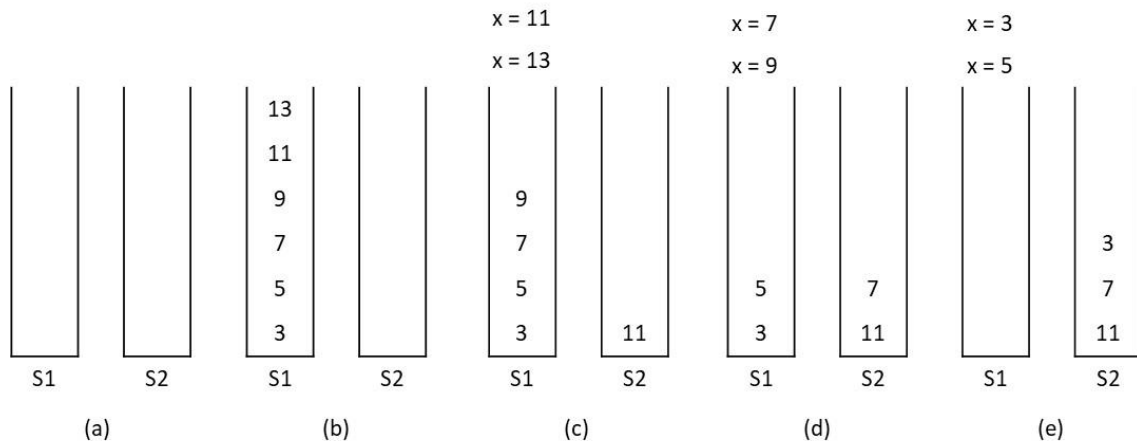
2. Considere que temos duas pilhas de inteiros, s1 e s2, mostre em uma figura a evolução de cada pilha após as seguintes operações:

```

1 pushStack (s1, 3)
2 pushStack (s1, 5)
3 pushStack (s1, 7)
4 pushStack (s1, 9)
5 pushStack (s1, 11)
6 pushStack (s1, 13)
7 loop not emptyStack (s1)
  1 popStack (s1, x)
  2 popStack (s1, x)
  3 pushStack (s2, x)
8 end loop

```

Resposta2.- Temos inicialmente duas pilhas vazias, s1 e s2.



3. Use a **transformação manual** para transformar as seguintes expressões infixas nas suas respectivas formas posfixas e prefixas:

- $D - B + C$
- $A * B + C * D$
- $(A + B) * C - D * F + C$
- $(A - 2 * (B + C) - D * E) * F$

Resposta3.- Na transformação manual identifica-se as expressões infixas mais internas de acordo com precedência dos operadores (parênteses tem maior precedência, depois multiplicação e divisão, depois soma e subtração) e substitui-se elas primeiro com expressões posfixas (ou prefixas equivalentes).

c) Fazendo a conversão para posfixa, temos:

$$(A + B) * C - D * F + C$$

$$(AB+) * C - DF * +C$$

$$AB + C * - DF * +C$$

$$AB + C * DF * - + C$$

$$AB + C * DF * -C +$$

Fazendo a conversão para prefixa, temos:

$$(A + B) * C - D * F + C$$

$$(+AB) * C -* DF + C$$

$$\begin{aligned}
 &* + ABC - * DF + C \\
 &- * + ABC * DF + C \\
 &+ - * + ABC * DFC
 \end{aligned}$$

4. Mude as expressões infixas da questão anterior em expressões posfixas usando o **método algorítmico** com base em uma pilha. Ilustre passo a evolução da pilha.

Resposta4.-

Algoritmo no Slide 10-S6 Pilhas aplicações parte 2.

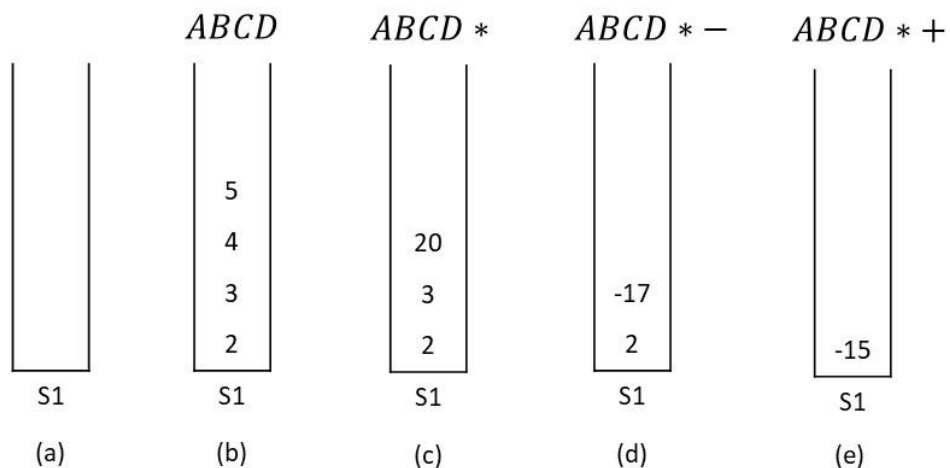
5. Pense em um método algoritmo para transformar a expressão em prefixa.

Resposta5.-

6. Determine o valor das seguintes expressões posfixas, considerando que as variáveis possuem os seguintes valores: A=2, B=3, C=4 e D=5. Ilustre o procedimento usando uma pilha para armazenar os valores intermediários.

- a. $A B * C - D +$
 b. $A B C + * D -$
 c. $A B C D * - +$
 d. $D C * B A + -$

Resposta6.- O método algorítmico neste caso consiste em processar a expressão de esquerda para direita. Sempre que um operando é encontrado empilhá-lo. Sempre que um operador é encontrado desempilhar os dois últimos operandos da pilha, considerando que o primeiro desempilhado será segundo operando e o segundo desempilhado será o primeiro operando. Executar a operação e empilhar o resultado novamente. O algoritmo termina com um único elemento na pilha.

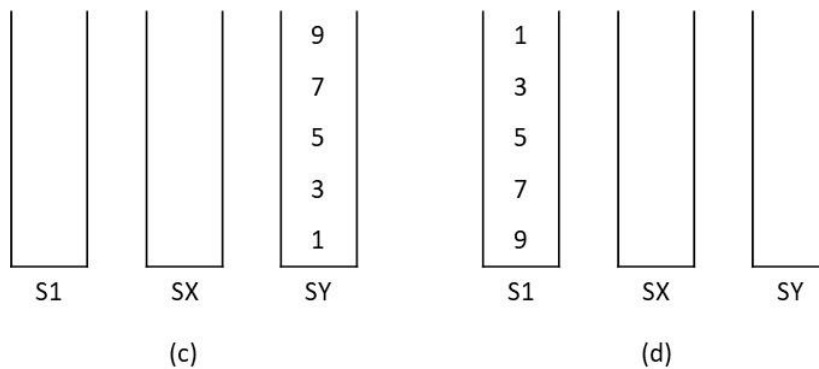
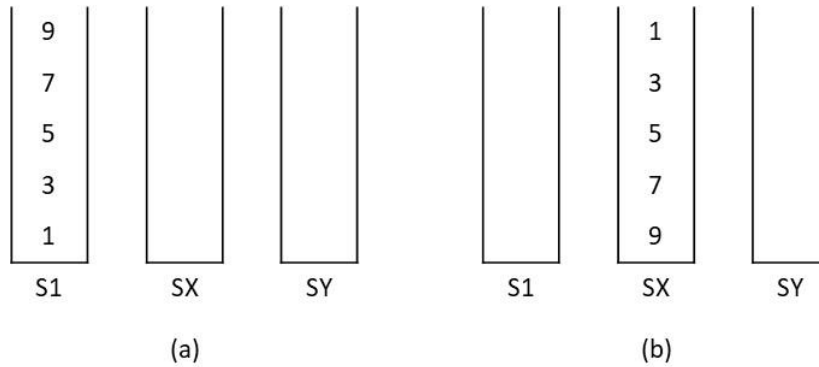


7. Desenvolva um algoritmo para inverter o conteúdo de uma pilha. *Dica.* Use pilhas temporárias para armazenar o conteúdo da pilha original e fazer a inversão. Ilustre com exemplos.

Resposta7.-

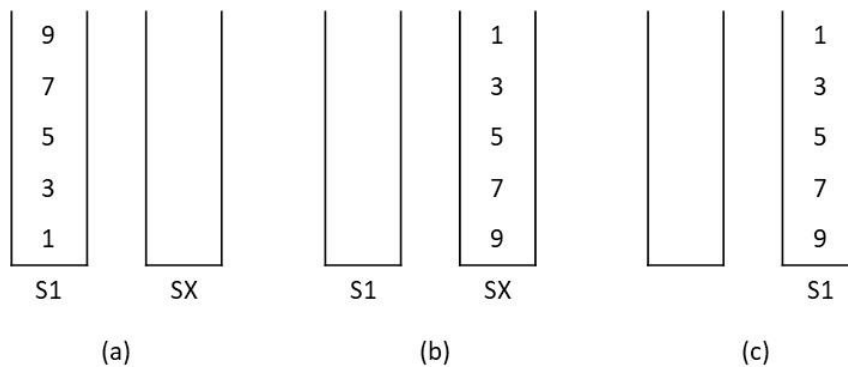
Foram discutidas duas soluções possíveis:

A primeira usando apenas operações definidas no TAD Pilha, `popStack()` e `pushStack()`. Seriam necessárias duas pilhas auxiliares SX e SY tendo em vista que a remoção dos elementos da pilha original em outra auxiliar não seria suficiente para inverter os dados de volta na pilha original. A situação é ilustrada no exemplo.



O algoritmo seria formado por três loops para realizar a remoção de elementos de uma pilha e inserção em outra enquanto a pilha não estiver vazia `emptyStack()`.

A segunda ideia utilizaria operações fora do TAD Pilha original, ou como uma extensão das operações do TAD, visando a eficiência da tarefa. Neste caso utilizaria apenas uma pilha auxiliar SX, para realizar a inversão dos dados. Destruiria a pilha vazia S1, `destroyStack()`. E utilizaria a mesma variável ponteiro S1 para armazenar a pilha auxiliar, mudando o nome da pilha auxiliar para S1.



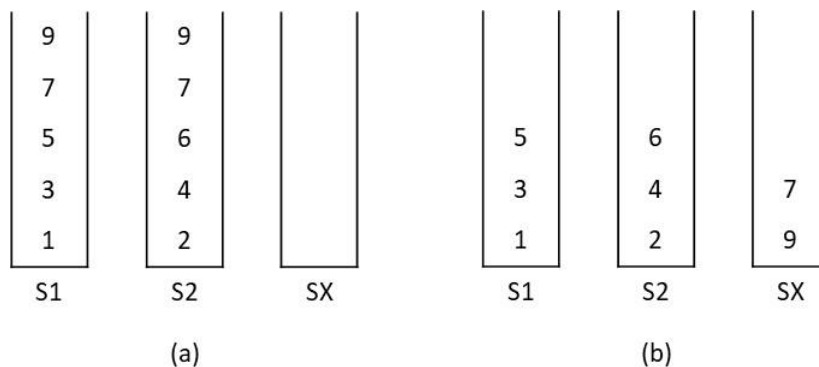
8. Desenvolva um algoritmo para testar se duas pilhas P1 e P2 são iguais. Duas pilhas são iguais se possuem os mesmos elementos, exatamente na mesma ordem. Considere que as pilhas P1 e P2 já existem e são passadas como parâmetro.

Considere que as pilhas possuem elementos do tipo inteiro. Caso seja necessário, criar pilhas auxiliares. Ilustre com exemplos.

Resposta8.-

O algoritmo realizaria primeiro a comparação pelo número de elementos, `stackCount()`. Caso o número fosse igual, prosseguiria comparando o topo de ambas pilhas `stackTop(P1)` e `stackTop(P2)`. Caso forem iguais, prosseguiria extraíndo o topo de ambas as pilhas para ter acesso ao próximo elemento, `popStack(P1)` e `popStack(P2)`, repetindo isso dentro de um loop. Se durante as comparações os elementos correspondentes forem diferentes o loop termina e conclui-se que as pilhas são diferentes.

Durante o processo de remoção de elementos da ambas as pilhas, foi observado que, deve-se usar uma pilha auxiliar PX, para armazenar os elementos removidos (e iguais em ambas pilhas) para poder reconstruir as pilhas após a verificação, mediante um segundo loop. A situação é ilustrada no exemplo.



Observe que em (b) já podemos concluir que as pilhas são diferentes e usar a pilha PX para reconstruir ambas a pilhas.

- Desenvolva uma função chamada *copystack()* que copie o conteúdo de uma pilha em outra. A função deve receber duas pilhas como parâmetros: P1, a pilha origem e P2, a pilha destino. A pilha destino terá seu conteúdo substituído. A ordem dos dados na pilha origem dever ser preservada. *Dica.* Use uma pilha auxiliar para armazenar o conteúdo da pilha origem. Ilustre com exemplos.

Resposta9.-

- Desenvolva uma função chamada *catstack()* que concatene (anexar) o conteúdo de uma pilha no topo da outra. A função deve receber duas pilhas como parâmetros: P1, a pilha origem e P2, a pilha destino. A ordem dos dados na pilha origem dever ser preservada. *Dica.* Use uma pilha auxiliar para armazenar o conteúdo da pilha origem. Ilustre com exemplos.

Resposta10.-

Considerando os conceitos sobre Filas:

- Imagine que temos uma pilha de inteiros, S, e uma fila de inteiros, Q. Desenhe uma figura de S e Q após as seguintes operações:

```

1 pushstack (S, 3)
2 pushstack (S, 12)
3 enqueue (Q, 5)
4 enqueue (Q, 8)
5 popstack (S, x)
6 pushstack (S, 2)
7 enqueue (Q, x)
8 dequeue (Q, y)
9 pushstack (S, x)
10 pushstack (S, y)

```

12. Qual seria o conteúdo da fila Q, após o seguinte código ser executado:

```

1 Q = createqueue
2 loop (not end of file)
  1 read number
  2 if (number not 0)
    1 enqueue (Q, number)
  3 else
    1 x = queuerear (Q)
    2 enqueue (Q, x)
  4 end if
3 end loop

```

Considerando que a seguinte sequência de dados é ingressada:

5, 7, 12, 4, 0, 4, 6, 8, 67, 34, 23, 5, 0, 44, 33, 22, 6, 0.

13. 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.

14. Qual seria o conteúdo da fila Q1 após o seguinte código ser executado:

```

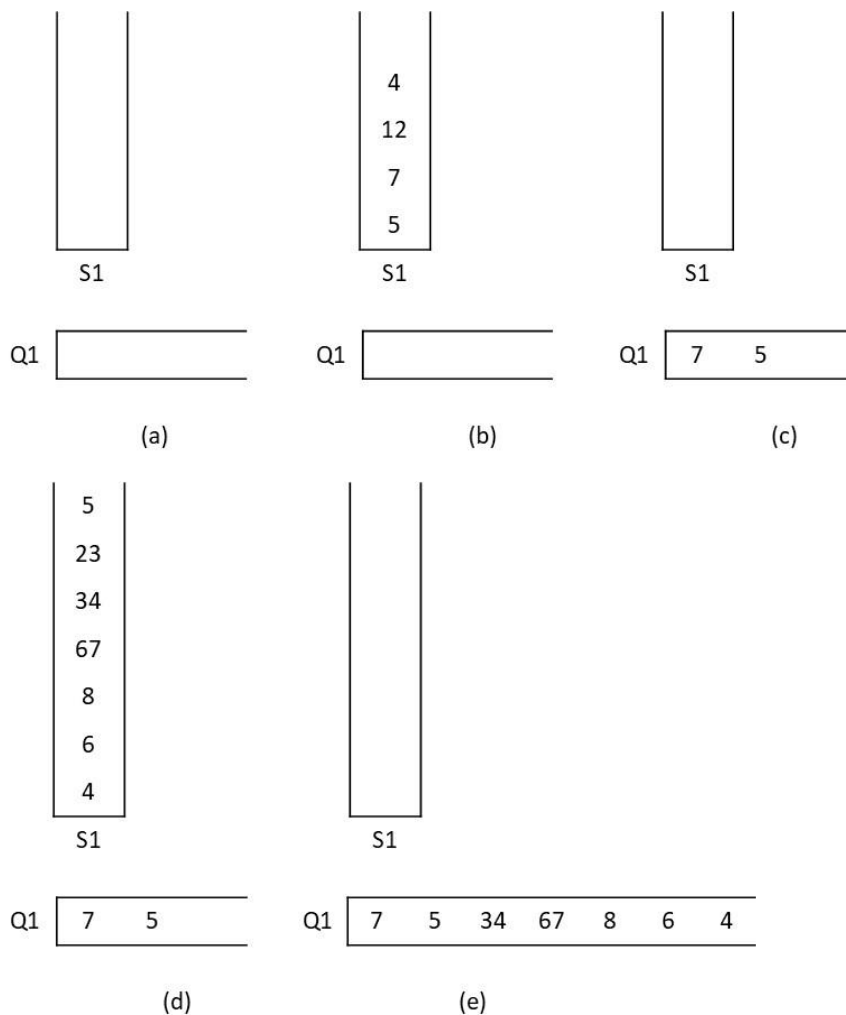
1 q1 = createqueue
2 s1 = createstack
3 loop (not end of file)
  1 read number
  2 if (number not 0)
    1 pushstack (s1, number)
  3 else
    1 popstack (s1, x)
    2 popstack (s1, x)
    3 loop (not empty s1)
      1 popstack (s1, x)
      2 enqueue (q1, x)
    4 end loop
  4 end if
4 end loop

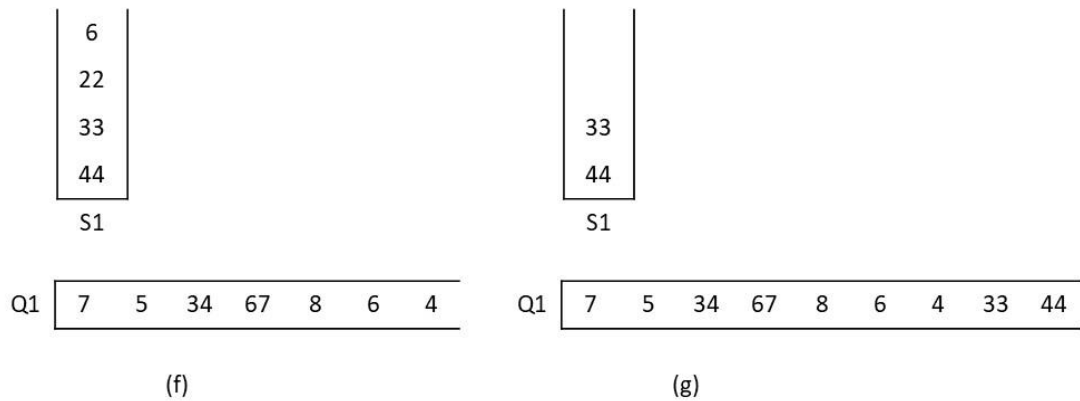
```

Considerando que, a seguinte sequência de dados é ingressada:

5, 7, 12, 4, 0, 4, 6, 8, 67, 34, 23, 5, 0, 44, 33, 22, 6, 0.

Resposta14.- O algoritmo opera sobre uma pilha S1 e uma fila Q1. No loop empilha elementos de uma sequência sempre que forem diferentes de 0. Se o elemento for 0, remove dois elementos da pilha descartando-os, e inicia um segundo loop interno que remove um elemento e o insere na fila Q1. Temos assim:





15. Imagine que o conteúdo das filas Q1 e Q2 é o seguinte:

Q1: 42 30 41 31 19 20 25 14 10 11 12 15

Q2: 4 5 4 10 13

onde, o início da fila encontra-se no extremo esquerdo e o fim da fila encontra-se no extremo direito. Qual seria o conteúdo da fila Q3 após a execução do seguinte código.

```

1 Q3 = createqueue
2 count = 0
3 loop (not empty Q1 and not empty Q2)
  1 count = count + 1
  2 dequeue (Q1, x)
  3 dequeue (Q2, y)
  4 if (y equal count)
    1 enqueue (Q3, x)
  5 end if
1 end loop

```

16. Dada uma fila de inteiros, escreva um algoritmo que, usando somente as funções do TAD Fila (Queue ADT), calcule e imprima o soma e a média dos inteiros na fila e preserve o conteúdo da fila.

Resposta16.- O algoritmo deve acessar o número de elementos na fila, usando a operação `queueCount()`. Depois, deve usar um loop para processar os elementos da fila, um elemento de cada vez, com base no contador, extraíndo o elemento da frente usando a operação `dequeue()`. Usar um acumulador para calcular a soma. Inserir o elemento extraído de novo no final da fila, para preservar a fila, operação `enqueue()`. Finalmente, calcular a média.

17. Usando somente as funções do TAD Fila (Queue ADT), escreva o algoritmo de uma função que compare o conteúdo de duas filas e retorne True, caso as filas sejam idênticas; e False caso contrário.

18. Usando somente as funções do TAD Fila (Queue ADT), escreva o algoritmo de uma função chamada `copyQueue()` para copiar o conteúdo de uma fila origem em outra destino. Considere as seguintes situações: i) a fila origem não precisa ser preservada após a cópia; ii) a fila origem precisa ser preservada após a cópia.

Resposta18.- Podemos considerar que a fila destino não existe e precisa ser criada. Caso contrário, teríamos que lidar com a destruição dos dados pre-existent.

- i) Neste caso, basta extrair os dados da fila origem e inserir na fila destino. Tudo isso elemento a elemento, dentro de um loop.
- ii) Neste caso, extrair os dados da fila origem, e além de inserir os dados na fila destino, inserir de volta na fila origem, preservando essa fila. Tudo isso elemento a elemento, dentro de um loop.

19. Escreva um algoritmo para uma função chamada *stackToQueue()* que crie uma fila destino a partir de uma pilha origem. Após criar a fila, o topo da pilha deverá corresponder ao início da fila e a base da pilha deverá corresponder ao fim da fila. A pilha deverá ficar vazia após a execução da função.

Resposta19.- O algoritmo é um processamento direto da pilha e da fila. Extrair da pilha e inserir na fila, sem qualquer tratamento adicional.

20. Dada uma fila de inteiros, escreva um algoritmo que, usando somente as funções do TAD Fila (Queue ADT), elimine todos os inteiros negativos presentes na fila e preserve o conteúdo restante da fila sem mudar a ordem dos elementos.

Resposta20.- O algoritmo é semelhante ao da questão 16. Mas neste caso, os elementos negativos extraídos não são inseridos de volta na fila.

Considerando os conceitos sobre Listas:

21. Imagine que temos as duas listas, como mostradas na Figura 1.



Figura 1.

Responda que acontece se aplicamos as seguinte sentença:

```
list1 = list2
```

Resposta21.- A alocação de list2 em list1, teria como consequência a perda do acesso aos elementos da lista 1. Isso é uma situação que deve ser evitada uma vez que os dados permanecerão na memória mas estarão inacessíveis.

22. Imagine que temos a lista mostrada na Figura 2, implementada como uma lista encadeada.

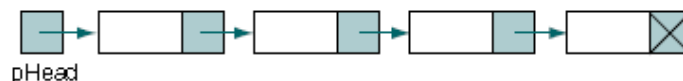


Figura 2.

Mostre que aconteceria se usássemos a seguinte sentença em uma busca percorrendo a lista?.

```
pHead = pHead->link
```

Isso, justifica o uso de dois ponteiros pPre e PLoc como descritos em aula?

Resposta22.- Entende-se que o percurso da lista significa processar dentro de um loop. Usar pHead desse jeito é errado porque perde-se o acesso ao início da lista. Neste caso, após o loop, PHead acabaria sendo nulo.

Precisamos utilizar variáveis auxiliares, para percorrer a lista. Usamos duas pPre e pLoc, porque pLoc indica a posição atual de busca o consulta, enquanto pPre é necessária em caso de inserção ou remoção.

23. Imagine que temos uma lista como a indicada na Figura 3.

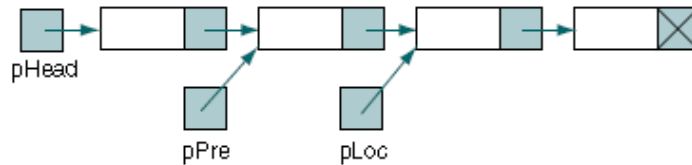


Figura 3.

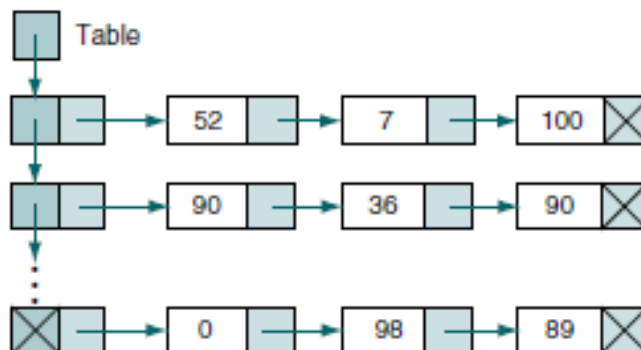
Onde, os ponteiros pPre e pLoc são usados para percorrer a lista durante uma busca e apontam para o nó predecessor e o nó atual da busca. Ao percorrer a lista devemos avançar atualizando pPre e pLoc. O seguinte código pretende definir a atualização mas tem um erro. Qual é esse erro e como pode ser corrigido?.

```
pLoc = pLoc->link
pPre = pPre->link
```

24. Escreva um algoritmo que misture duas listas ordenadas em uma única lista usando o TAD Lista.

Resposta24.- Uma forma de abordar a questão seria propor um algoritmo de intercalação que faria a leitura do primeiro elemento de cada lista, realizaria a comparação e extrairia o menor deles, inserindo-o em uma terceira lista ordenada. O processamento continuaria dentro de um loop, até esgotar os elementos de uma das duas listas. Finalmente, os elementos da outra lista que não esgotou, seriam copiados na terceira lista.

25. Escreva um algoritmo que crie uma matriz bidimensional usando listas encadeadas. Os nós na primeira coluna contém apenas ponteiros. O ponteiro esquerdo aponta para próxima linha. O ponteiro direito aponta para os dados de uma determinada fila. Observe a figura.



Resposta25.- Foram discutidas duas possibilidades para essa estrutura.

A primeira seria definir a estrutura com dimensões fixas desde o início. Número de linhas e número de colunas fixas, inicializadas com valores nulos. De forma que cada elemento da matriz ocupa uma posição em alguma lista. Por exemplo, a primeira lista corresponderia a primeira linha da matriz, e o quinto elemento nessa lista, representaria o elemento na posição (1,5). O algoritmo criaria uma lista para cada linha. No entanto, foi questionada a utilidade do uso de listas dinâmicas dessa forma.

A segunda ideia, seria utilizar esse tipo de estrutura para armazenar matrizes esparsas (sparse matrix) em que um grande número de elementos é zero e apenas poucas posições possuem valores diferentes de zero. Neste caso, somente os elementos diferentes de zero seriam armazenados economizando-se espaço. Nesta representação, além do valor número armazenado, seria necessário registrar a posição (os índices) do elemento armazenado.