

## Introdução Teórica

### Descrição da estrutura de dados PILHA

Uma estrutura de dados é agrupamento de valores que são armazenados em uma única variável, por exemplo, vetores e matrizes.

Uma lista linear é uma estrutura de dados que além de armazenar vários valores de elementos, impõe que a posição de cada elemento deve respeitar algum tipo de ordem.

Por exemplo, em uma lista de pacientes que estão na sala de espera para atendimento deve ser respeitada a ordem cronológica de chegada de cada paciente. Portanto, uma lista deve não apenas conter as informações de cada paciente, mas também oferecer uma ordenação, de forma que sempre se saiba quem foi o primeiro a chegar, o segundo e assim por diante.

Existe todo um formalismo para tratar listas lineares que será devidamente abordado, posteriormente, nessa disciplina.

Para o trabalho proposto deve-se utilizar um tipo especial de lista linear conhecido como **PILHA**.

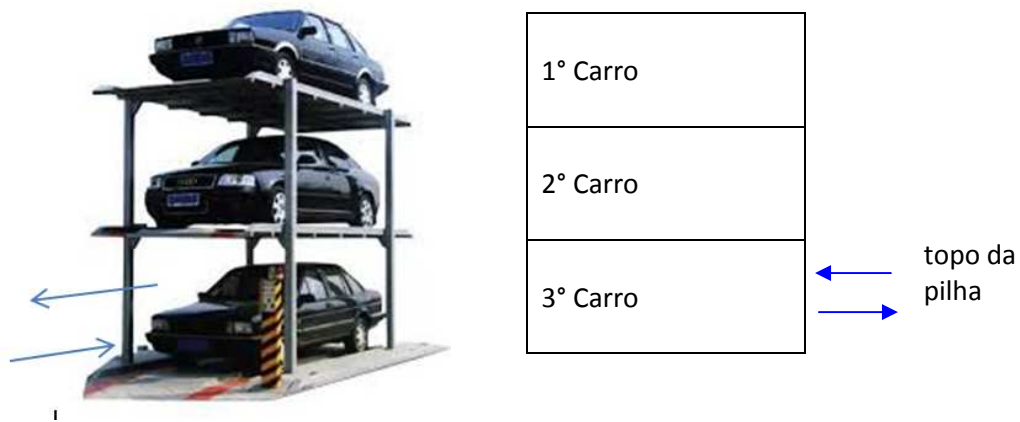
Uma pilha (*stack*) é uma lista linear onde as operações de inserção e de remoção são efetuadas apenas em uma extremidade, denominada topo da pilha. Estruturas deste tipo são conhecidas como LIFO (*last in first out* – último a entrar primeiro a sair).

A figura a seguir mostra uma representação de pilha.



Exemplo **PILHA**:

“Analogia” com estacionamento vertical.



Mas, qual a diferença entre os conceitos de VETOR e **PILHA**?



Em um estacionamento térreo, cada carro poderá ocupar qualquer vaga disponível do estacionamento. Assim, a entrada e saída é feita de forma aleatória e não se pode saber qual carro foi o primeiro em nem qual foi o último a entrar no estacionamento.

E se por engano se tratar pilha como vetor?



Pilhas são usadas em aplicações onde é necessário que a ordem de saída dos elementos seja inversa da ordem de entrada.

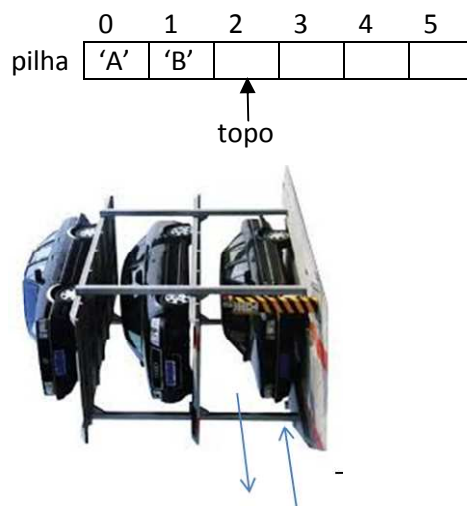
Neste momento, será demonstrado como uma **pilha** de caracteres pode ser **implementada** usando um **vetor** para armazenar os vários caracteres (ou elementos). Além do vetor, a pilha necessita de uma variável inteira, que defina quem é o **topo**. Esta variável será utilizada como único índice de acesso à posição dos elementos do vetor.

Portanto, uma pilha para ser completamente definida precisa de um vetor de elementos e da indicação da extremidade de inserção e remoção denominada de topo da pilha.

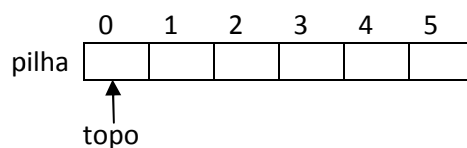
#### Conceito importante:

Topo não é a parte superior de uma pilha, mas sim sempre indica a única extremidade de inserção e/ou remoção de um elemento. Como pode ser visto na figura a seguir.

Na figura o **topo** indica que o próximo elemento a ser inserido será na posição de **índice 2**. Além disto, o índice 0 indica que o caractere 'A' foi o primeiro elemento a ser inserido.



Utilizaremos a **convenção de topo indicando a posição de índice 0** para definir que a pilha está vazia. Como mostra a figura:

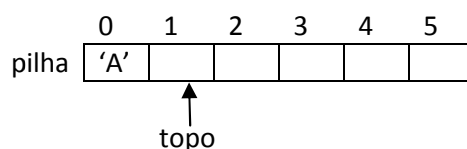


Assim, a **INSERÇÃO (EMPILHAR)** em uma pilha deverá ser feita em 2 passos:

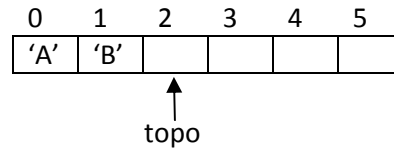
```
pilha[topo] = valor  
topo = topo + 1
```

Supondo realizar as seguintes operações na pilha:

- **EMPILHAR** o caracter 'A' na pilha:



- **EMPILHAR** o valor 'B' na pilha:

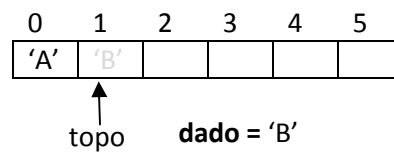


De forma análoga, a **REMOÇÃO (DESEMPILHAR)** de um valor de uma pilha deverá ser feita em 2 passos:

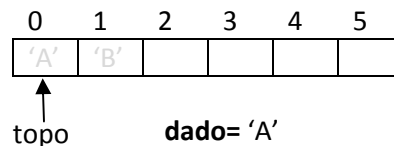
topo = topo - 1  
 dado = pilha[topo]

Supondo a seguinte sequência de operações:

- **DESEMPILHAR** o dado que está no topo da pilha:



- **DESEMPILHAR** o dado que está no topo da pilha:



**Observação:** A analogia da pilha com o estacionamento vertical tem uma diferença na saída dos dados. Na saída dos carros é necessário movimentá-los de posição para depois retirá-los. Na pilha para não desperdiçar tempo computacional com movimentações de elementos no vetor, simplesmente retira-se o elemento da vez, que é apontado pelo **"topo"**.

## Descrição da Notação Polonesa Reversa

As operações aritméticas representam uma área importante nas ciências da computação e na matemática de modo geral. Usualmente, uma expressão aritmética é formada por um ou dois operandos que podem ser (valores, exemplo 2,3, etc) ou variáveis (exemplo, x,y,z) e um operador, como, por exemplo, em A+B, onde A e B são operandos e '+' é o operador.

A representação da operação, correntemente denotada nesta forma A+B é chamada infixa. Há outras representações, igualmente válidas, a saber:

A+B    infixa  
 +AB    prefixa  
 AB+    posfixa

As duas últimas notações têm grande valor na computação automática dos valores das expressões, pois eliminam problemas de precedência de operadores (prioridade de operação). Em particular, iremos abordar apenas a notação posfixa.

Analisando apenas a conversão da notação infixa para a posfixa suponha os seguintes exemplos:

- 1) Infixa:  $A + B * C$   
Posfixa:  $A B C * +$
- 2) Seja a expressão infixa:  $(A + B) * C$   
Equivale a expressão posfixa:  $A B + C *$
- 3) Infixa:  $A * (B - C)$   
Posfixa:  $A B C - *$

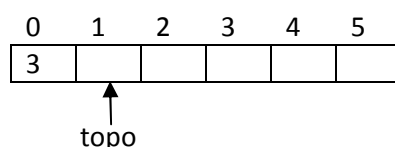
A notação posfixa também é conhecida como notação polonesa reversa (**RPN** – *Reverse Polish Notation*). Esta notação ganhou destaque na década de 70, por sua adoção nas calculadoras HP. As calculadoras de maior sucesso da empresa, a financeira HP12-C<sup>1</sup> e a gráfica HP48G são baseadas em notação polonesa reversa.

Umas das vantagens da RPN é redução do número de teclas digitadas, menos digitação menos erros.

Para usar a notação RPN é necessário o uso de uma estrutura de dado do tipo pilha.

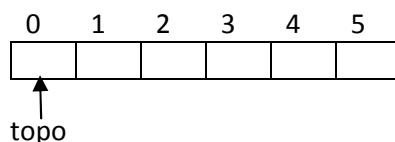
Supondo que  $A = 3$ ,  $B = 5$  e  $C = 2$  e deseja-se calcular a expressão posfixa  $A B C * +$ , quais são os passos a serem executados na calculadora com notação RPN? São eles:

- **EMPILHAR** o valor 3 na pilha, para isso digitar **<3>** **<ENTER>**



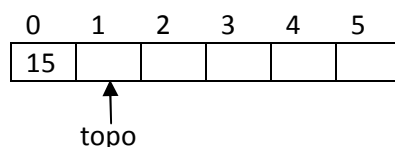
- **Inserir** o valor 5 no visor da calculadora, digitando **<5>** e, em seguida, a operação de multiplicação digitando **<x>**. O valor 5 (que está no visor da calculadora) é o valor de um **dado1** o outro será obtido desempilhando o valor que está na pilha, assim, implicitamente:

- **DESEMPILHAR** o valor do topo da pilha:



**dado2 = 3**

- **Multiplicação é feita:** **res = dado2 \* dado1 = 3 \* 5 = 15**
- **EMPILHAR** o valor 15 (res) na pilha:



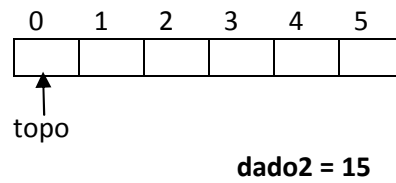
- **Inserir** o valor 2 no visor da calculadora, digitando **<2>** e, em seguida, a operação de adição digitando **<+>**. O valor 2 (que está no visor da calculadora) é o valor de um

---

<sup>1</sup><http://www.fazerfacil.com.br/calculadoras/hp12c-platinum.html>

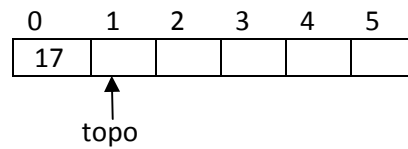
**dado1** o outro será obtido desempilhando o valor que está na pilha, assim, implicitamente:

- **DESEMPILHAR** o valor do topo da pilha:



- **Soma é feita:**  $\text{res} = \text{dado2} + \text{dado1} = 15 + 2 = 17$

- **EMPILHAR** o valor 17 na pilha:



Assim, o resultado final da expressão pode ser calcular usando a notação RPN.

**Experimente propor novas expressões!!!**

## Trabalho a ser Desenvolvido

Elabore um programa em C que implemente o fluxograma apresentado na página a seguir.

O fluxograma converte uma expressão matemática na notação infixa em notação posfixa. A expressão posfixa obtida realiza operação aritmética sobre 2 dados utilizando a notação RPN descrita anteriormente. O programa a ser desenvolvido pode apenas realizar as operações aritméticas básicas (+ - \* /) simulando uma calculadora que usa notação RPN.

Para tanto, são necessárias algumas funções especiais da linguagem C\*, como por exemplo:

- isdigit(): verifica se um caracter é um símbolo numérico (0,1,..9).

Essa função e outras que analisam caracteres podem ser encontradas na referência:

\* No portal=>biblioteca virtual=> livro: "C: como programar", 6ª Ed; Deitel, P.; Deitel, H. =>capítulo 8

Para simplificar o programa:

- cada número da expressão poderá ter apenas 1 dígito (0 a 9).
- não existe espaço em branco entre os dígitos e operações na string que será lida do teclado contendo a expressão infixa a ser transformada.
- Não será entrada parênteses na expressão infixa.

Por exemplo:

- a expressão infixa  $3+2$  equivale a posfixa  $23+$ .
- a expressão infixa  $3+2/4*3$  equivale a posfixa  $324/3*+$ .
- a expressão infixa  $6/3-4*3$  equivale a posfixa  $63/43*-$ .

**Data Final de Entrega:** 10 de abril de 2019

**Grupo:** máximo 4 pessoas.

**Entrega:** O arquivo ".c" deve **IMPRESSO** e entregue em sala de aula até o dia 10.

**NÃO** será aceito entrega em dia posterior.

- A entrega do arquivo ".c" terá peso 2 da nota NAC 2 (parte1)
  - O cabeçalho deve ter a especificação da turma e dos componentes do grupo, que devem estar identificados completamente **nome e RM**.
  - Os trabalhos serão recolhidos minutos antes do início do exercício descrito a seguir.
- No dia 10 de abril será realizado um exercício individual em sala:
  - peso 8 da nota NAC 2 (parte1)
  - serão avaliados conhecimentos utilizados na elaboração do trabalho entregue
  - exercício terá duração máxima de 40 minutos no final da aula.

**Observação:** NAC2 será feita em duas partes de pesos iguais. A nota final da NAC2 será lançada no final do semestre e terá peso 10.

