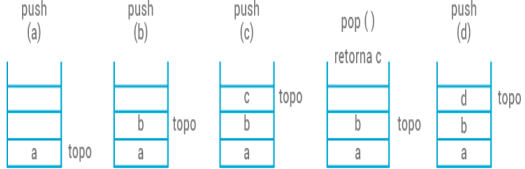


PILHAS	PILHAS
Como podemos definir uma <b>Estrutura de Dados em Pilha</b> ?	Uma pilha é uma lista linear em que apenas as operações de acesso, inserção e remoção são possíveis. A ideia fundamental da pilha é que todo o acesso aos seus elementos é feito por meio do seu topo. Assim, quando um elemento novo é introduzido na pilha, passa a ser o elemento do topo, e o único elemento que pode ser removido da pilha é o do topo. Isto faz com que os elementos da pilha sejam retirados na ordem inversa à ordem em que foram introduzidos: o primeiro que sai é o último que entrou (a sigla LIFO – last in, first out – é usada para descrever esta estratégia).
Como podemos <b>exemplificar</b> o funcionamento de uma Estrutura em Pilha?	Para se entender o funcionamento de uma estrutura de pilha, pode-se fazer uma analogia com uma pilha de pratos. Se quiser adicionar um prato na pilha, o mesmo deve ser colocado no topo. Para pegar um prato da pilha, deve-se retirar o prato do topo. Assim, tem-se que retirar o prato do topo para ter acesso ao próximo prato. A estrutura de pilha funciona de maneira análoga. Cada novo elemento é inserido no topo e só tem-se acesso ao elemento do topo da pilha.
Que <b>Operações</b> nós podemos fazer numa Estrutura do tipo Pilha?	Existem duas operações básicas que devem ser implementadas numa estrutura de pilha: a operação para empilhar um novo elemento, inserindo-o no topo, e a operação para desempilhar um elemento, removendo-o do topo. É comum se referir a essas duas operações pelos termos em inglês push (empilhar) e pop (desempilhar). Vamos usar como exemplo uma variável "p" que é do tipo pilha e uma variável "x" um elemento qualquer que vamos empilhar e depois desempilhar. Então, teremos:  <b>push (p, x): procedimento que insere (empilha) x no topo de p.</b> <b>pop (p): função que remove (desempilha) o elemento do topo de p, devolvendo o valor do topo.</b>
- reservado para a questão acima -	A figura a seguir ilustra o funcionamento conceitual das operações push e pop em uma pilha:
- reservado para a questão acima -	
Como podemos <b>criar uma variável do tipo Pilha</b> ?	Podemos criar uma variável do tipo Pilha usando um vetor, pois estaremos criando um elemento que acumula valores, assim como uma pilha faz. Agora, para que ele funcione como Pilha de fato, temos que criar uma função onde só será possível atribuir valores aos últimos lugares do vetor e retirar valores sempre do último valor atribuído para os primeiros valores atribuídos.
Como podemos fazer para <b>atribuir</b> valores a uma Pilha?	Como dissemos antes, precisamos criar uma função onde é possível atribuir valores à Pilha a partir das últimas colocações. Para isso é necessário sabermos a quantidade máxima de elementos que podem ser armazenados na pilha, ou seja, neste caso, a estrutura pilha tem seu limite conhecido. Dado um vetor conhecido de números inteiros para armazenar os elementos da pilha, os primeiros elementos ocupam as primeiras posições do vetor (n posições ocupadas). Temos, então, n elementos armazenados na pilha, logo o elemento n-1 representa o elemento do topo. O tamanho do vetor é determinado por meio da constante MAX, a qual deve ser definida no início do programa. Abaixo é mostrado o procedimento “push” para inserir um elemento no topo da pilha. O procedimento recebe como parâmetros o vetor “pilha” e o novo elemento x que será inserido. Para a implementação desse procedimento, é preciso considerar que a variável topo é global e, inicialmente, seu valor é zero. Além disso, ela está sendo utilizada para controlar a posição no vetor onde será inserido o novo elemento na pilha.
- reservado para a questão acima -	<pre>const int MAX[30]  void push (int pilha [MAX], int x) {     if (topo == MAX)         printf ("\n Pilha Cheia");     else{         pilha [topo] = x;         topo++;     } }</pre>
- reservado para a questão acima -	

