PILHAS Como podemos definir uma Estrutura de Dados em Pilha? Como podemos exemplificar o funcionamento de uma Estrutura em Pilha? Que Operações nós podemos fazer numa Estrutura do tipo Pilha? - reservado para a questão acima -- reservado para a questão acima -Como podemos criar uma variável do tipo Pilha? Como podemos fazer para atribuir valores a uma Pilha? - reservado para a questão acima pilha. - reservado para a questão acima -- reservado para a questão acima -

PILHAS

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).

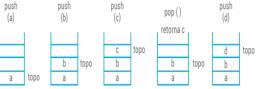
Para se entender o funcionamento de uma estrutura de pilha, pode-se fazer uma analogia com uma pilha de pratos. Se quiser adicionar um prato na pilha, o mesmo deve ser colocado no topo. Para pegar um prato da pilha, deve-se retirar o prato do topo. Assim, tem-se que retirar o prato do topo para ter acesso 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.

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:

push (p, x): procedimento que insere (empilha) x no topo de p. pop (p): função que remove (desempilha) o elemento do topo de p, devolvendo o valor do topo.

A figura a seguir ilustra o funcionamento conceitual das operações push e pop em uma pilha:



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 dissemos antes, precisamos criar uma função onde é possível atribuir valores á Pilha á 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 inicio 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 nilha

const int MAX[30]

```
void push (int pilha [MAX], int x)
{
  if (topo == MAX)
    printf ("\n Pilha Cheia");
  else{
    pilha [topo] = x;
    topo++;
  }
}
```

PILHAS Como podemos fazer para remover valores a uma Pilha? - reservado para a questão acima -- reservado para a questão acima -Como podemos exibir todos os valores de uma Pilha? - reservado para a questão acima -Como podemos aplicar o uso de uma Estrutura em Pilha? Ilustre... - reservado para a questão acima -- reservado para a questão acima -

PILHAS

Para remover (desempilhar) elementos do topo da pilha, deve ser utilizada a função "pop". Além de desempilhar o elemento, a função também retorna o valor que foi retirado da pilha. Vale lembrar que a remoção na pilha é feita por meio da variável topo, sendo que, a cada elemento retirado, é feito um decremento nesta variável. Abaixo é mostrada a implementação da função "pop".

```
int pop (int pilha [MAX])
{
    int x;
    if (topo >= 1)
    {
        x = pilha [topo - 1];
        topo--;
    }
    else printf ("\n Pilha vazia");
    return (x);
}
```

A exibição dos elementos da pilha deve ser feita por meio da utilização do **laço "para"**, onde o fim do laço é definido pela variável topo. Abaixo é mostrada a implementação do procedimento para exibir os elementos da pilha.

```
void exibe (int pilha [MAX])
{
   int i;

   if (topo >= 1){
      for (i=0; i<topo; i++)
        printf ("\n Pilha [%d] = %d",i,pilha [i]);
   }
   else printf ("\n Pilha vazia");
}</pre>
```

As estruturas de pilha são comumente usadas em algoritmos de gerenciamento de memória (alocação de variáveis na memória e retorno de procedimentos), compiladores e sistemas operacionais.

Por exemplo podemos usar aplicação da pilha no funcionamento de uma calculadora com expressões pós-fixadas, que poderia calcular a seguinte expressão: (1-2)*(4+5) que na mesma calculadora poderia ser digitado 1 2 – 4 5 + *. Por que? Podemos entender isso melhor analizando o passo a passo desse calculo numa estrutura em Pilha...

Passo 1	Passo 2	Passo 3	Passo 4	Passo 5	Passo 6	Passo 7
12-45+*	12-45+	12-45+*	12-45+*	12-45+*	12-45+*	12-45+*
Push(1)	Push(2)	n1 = pop() n2 = pop() push(n1-n2)	Push(4)	Push(5)	n1 = pop() n2 = pop() push(n1+n2)	1 1 17
	0 . to		A . tono	5 ←topo		
1 ←topo	2 ← top	-1 ←topo	4 ← topo -1	-1	9 ← topo -1	-9 ←topo