

## 5. PILHA em C - Estruturas de Dados

---

### 5.1 Conceito

---

- **Pilha** é uma estrutura de dados que admite *remoção de elementos e inserção de novos objetos*;
- Sempre que houver uma remoção, o elemento removido é o que está na estrutura a menos tempo;
- **LIFO** ("Last in, First out"), ou seja, o último a entrar será o primeiro a sair;
- Os elementos vão sendo inseridos um a um com o comando PUSH, e retirados com o comando POP, assim são empilhados e desempilhados sempre pelo topo da pilha;
- Pode-se fazer uma comparação com uma pilha de pratos, em que, se quiser adicionar um prato terá que ser no topo, e para pegar um prato é retirado do topo. Dessa forma é preciso retirar o prato do topo se quiser ter acesso ao próximo prato.

### 5.2 Exemplos de Aplicação

---

- Softwares aplicativos;
- Editores de texto;
- Editores de Planilhas;
- Função "CTRL+Z", o voltar;
- Funções recursivas em compiladores;
- Mecanismo de desfazer/ refazer do bloco de notas;
- Navegação entre páginas Web;
- Análise de expressões aritméticas;
- Entre outros.

### 5.3 Estrutura da FILA

---

TAD (Tipo Abstrato de Dados) da PILHA:

```
#define MAXPILA 10

struct TpPilha{
    int topo;
    char pilha[MAXPILA];
};
```

Obs.: Nesse exemplo é criado uma pilha de char(caracteres).

## 5.4 Operações Associadas

---

- void inicializa (TpPilha &p);
- void insere (TpPilha &p, char elemento);
- char retira (TpPilha &p);
- char elementoTopo (TpPilha p);
- int cheia (int topo);
- int vazia (int topo);
- int exhibe (TpPilha p);

## 5.5 Implementação em C

---

```
#define MAXPILA 10

struct TpPilha{
    int topo;
    char pilha[MAXPILA];
};

void inicializa(TpPilha &p){
    p.topo = -1;
}

void insere(TpPilha &p, char elemento){
    p.pilha[++p.topo] = elemento;
}

char retira(TpPilha &p){
    return p.pilha[p.topo--];
}

char elementoTopo(TpPilha p){
    return p.pilha[p.topo];
}

int cheia(int topo){
    return topo == MAXPILA-1;
}

int vazia(int topo){
    return topo == -1;
}

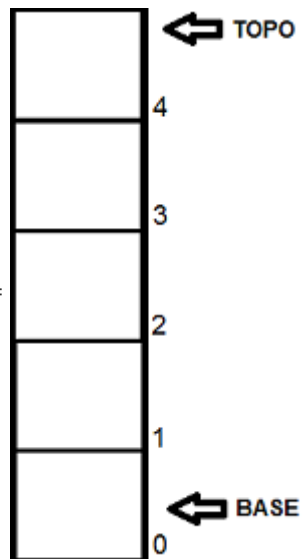
int exhibe(TpPilha p){
    while( !vazia(p.topo) )
        printf("\n%c",retira(p));
}
```

## 5.6 Funcionamento

---

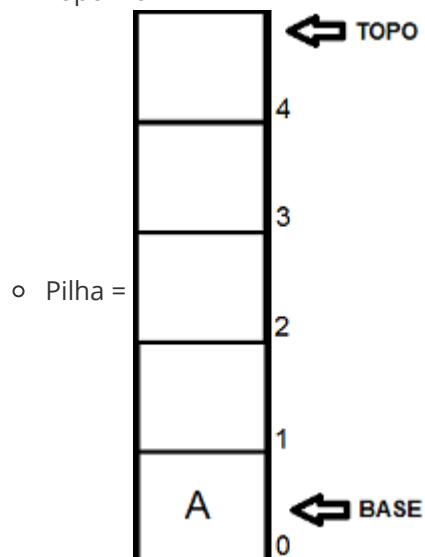
1. Inicializar:

- Topo = -1



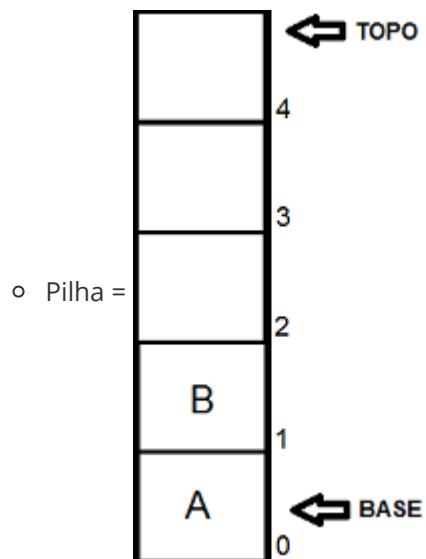
2. Inserir 'A':

- Topo = 0

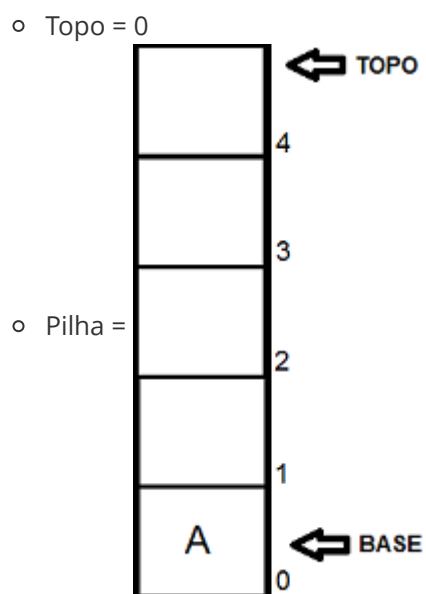


3. Inserir 'B':

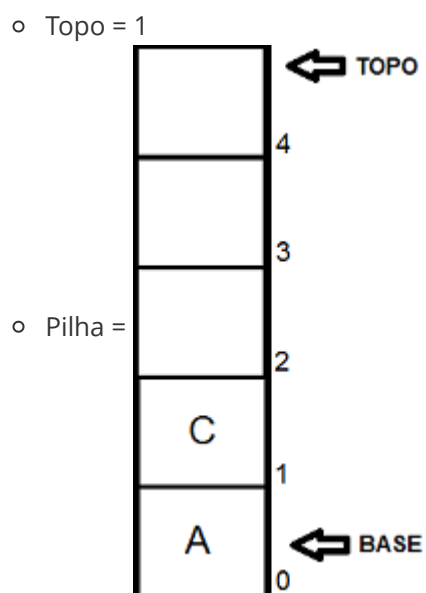
- Topo = 1



4. Retirar:



5. Inserir 'C':



6. Retirar:

○ Topo = 0

o Pilha =

