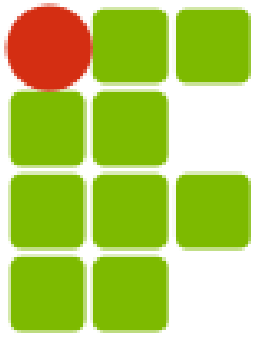


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

Estrutura de Dados

Ricardo José Cabeça de Souza
www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

Parte 10

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

- **PILHAS**
  - A ideia fundamental da pilha é que todo o acesso a seus elementos é feito através do seu topo
  - É uma estrutura sujeita à seguinte regra de operação: sempre que houver uma remoção, o elemento removido é o que está na estrutura há menos tempo
  - O primeiro que sai é o último que entrou (a sigla **LIFO – *last in, first out*** – é usada para descrever esta estratégia)

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

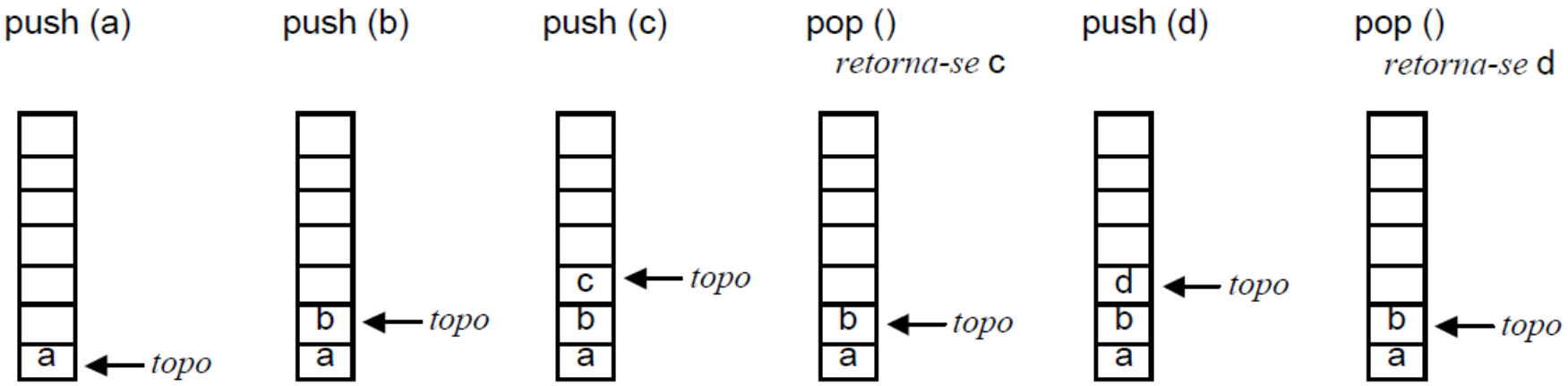
- **PILHAS**
  - Existem duas operações básicas que devem ser implementadas numa estrutura de pilha:
  - Operação para **empilhar** um novo elemento, inserindo-o no topo
  - Operação para **desempilhar** um elemento, removendo-o do topo
  - É comum nos referirmos a essas duas operações pelos termos em inglês ***push*** (**empilhar**) e ***pop*** (**desempilhar**)

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARÁ

PILHAS

- **PILHAS**



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

- **PILHAS**
  - Implementações de pilha
    - Usando vetores
    - Usando lista encadeada
  - Operações:
    - criar uma estrutura de pilha
    - inserir um elemento no topo (*push*)
    - remover o elemento do topo (*pop*)
    - verificar se a pilha está vazia
    - liberar a estrutura de pilha

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

- **PILHAS**
  - Implementação de pilha com vetor
  - Devemos ter um vetor (**vet**) para armazenar os elementos da pilha
  - Se temos **n** elementos armazenados na pilha, o elemento **vet[n-1]** representa o elemento do topo
  - A estrutura que representa o tipo pilha deve ser composta pelo **vetor** e pelo **número de elementos armazenados**

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARÁ

PILHAS

- **PILHAS**
  - A pilha está *vazia* se **t** vale **0** e *cheia* se **t** vale **n**

pilha[0..t-1] .


```
#define MAX  50

struct pilha {
   int n;
   float vet[MAX];
};
```

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

- **PILHAS**
  - Criação de pilha vazia

```c
/*Estrutura Pilha*/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define MAX 50
/*Definição da Estrutura Pilha*/
struct pilha{
   int n;
   float VET[MAX];
};
/*Função para criar a pilha vazia*/
pilha *cria (void)
{
pilha *p = (pilha*) malloc(sizeof(pilha));
p->n = 0; /* inicializa com zero elementos */
printf("\nPilha criada com sucesso!");
return p;
}
main()
{
pilha *ini;
ini = cria();
getch();
}
```

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

- **PILHAS**
  - Inserir um elemento na pilha

```
/*Função push: insere um elemento na pilha*/
void push (pilha *p, float v)
{
if (p->n == MAX) { /* capacidade esgotada */
printf("Capacidade da pilha estourou.\n");
exit(1); /* aborta programa */
}
/* insere elemento na próxima posição livre */
p->VET[p->n] = v;
p->n++;
}
```

```
/*Função Principal*/
main()
{
int cont=0;
pilha *ini;
float valor;
ini = cria();
if (vazia(ini) == 0)
 printf("\nPilha vazia!");
 else
 printf("\nPilha nao esta vazia!");

do{
printf("\nDigite um valor:");
scanf("%f",&valor);
push(ini,valor);
cont++;
}while(cont<3);
printf("\n\n");
mostra(ini);
printf("\n\n");
printf("\nElemento retirado: %.2f",pop(ini));
printf("\n\n");
mostra(ini);
getch();
}
```

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

- **PILHAS**
  - Verifica se a pilha está vazia

```
/*Função vazia: verifica se a pilha está vazia*/
int vazia (pilha *p)
{
if (p->n == 0)
 return 0;
 else
 return 1;
}
```

```
/*Função Principal*/
main()
{
int cont=0;
pilha *ini;
float valor;
ini = cria();
if (vazia(ini) == 0)
 printf("\nPilha vazia!");
 else
 printf("\nPilha nao esta vazia!");

do{
printf("\nDigite um valor:");
scanf("%f",&valor);
push(ini,valor);
cont++;
}while(cont<3);
printf("\n\n");
mostra(ini);
printf("\n\n");
printf("\nElemento retirado: %.2f",pop(ini));
printf("\n\n");
mostra(ini);
getch();
}
```

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARÁ

PILHAS

- **PILHAS**
  - Retira o elemento do topo da pilha

```
/*Função pop: retira elemento do topo*/
float pop (pilha *p)
{
float v;
if (vazia(p)==0) {
printf("Pilha vazia.\n");
exit(1); /* aborta programa */
}
/* retira elemento do topo */
v = p->VET[p->n-1];
p->n--;
return v;
}
```

```
/*Função Principal*/
main()
{
int cont=0;
pilha *ini;
float valor;
ini = cria();
if (vazia(ini) == 0)
 printf("\nPilha vazia!");
 else
 printf("\nPilha nao esta vazia!");

do{
printf("\nDigite um valor:");
scanf("%f",&valor);
push(ini,valor);
cont++;
}while(cont<3);
printf("\n\n");
mostra(ini);
printf("\n\n");
printf("\nElemento retirado: %.2f",pop(ini));
printf("\n\n");
mostra(ini);
getch();
}
```

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

- **PILHAS**
  - Exibe os elementos da pilha

```
/*Função mostra: exibe os elementos da Pilha*/
void mostra(pilha *p)
{
int i;
if(vazia(p)!=0)
 for(i=0;i<p->n;i++)
   printf(" %.2f",p->VET[i]);
}
```

```
/*Função Principal*/
main()
{
int cont=0;
pilha *ini;
float valor;
ini = cria();
if (vazia(ini) == 0)
 printf("\nPilha vazia!");
 else
 printf("\nPilha nao esta vazia!");

do{
printf("\nDigite um valor:");
scanf("%f",&valor);
push(ini,valor);
cont++;
}while(cont<3);
printf("\n\n");
mostra(ini);
printf("\n\n");
printf("\nElemento retirado: %.2f",pop(ini));
printf("\n\n");
mostra(ini);
getch();
}
```

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARÁ

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

PILHAS

- **PILHAS**

```c
/*Função libera: liberar memória alocada pela pilha*/
void libera (pilha *p)
{
free(p);
}
```

```c
/*Função Principal*/
main()
{
int cont=0;
pilha *ini;
float valor;
ini = cria();
if (vazia(ini) == 0)
 printf("\nPilha vazia!");
 else
 printf("\nPilha nao esta vazia!");

do{
printf("\nDigite um valor:");
scanf("%f",&valor);
push(ini,valor);
cont++;
}while(cont<3);
printf("\n\n");
mostra(ini);
printf("\n\n");
printf("\nElemento retirado: %.2f",pop(ini));
printf("\n\n");
mostra(ini);
libera(ini)
getch();
}
```

www.ricardojcsouza.com.br
ricardo.souza@ifpa.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARÁ

Estrutura de Dados

- **REFERÊNCIAS**
- Feofiloff, Paulo. **Projeto de Algoritmos em C**. Disponível em http://www.ime.usp.br/~pf/algoritmos/aulas/lista.html acesso em 12/07/2011.
- Tenenbaum, Aaron M. Langsam, Yedidyah, Augenstein, Moshe J. **Estruturas de dados usando C**. São Paulo : MAKRON *Books,* 1995.
- Veloso, Paulo. et. al. **Estrutura de dados**. Rio de Janeiro: Campus, 1986.
- Moraes, Celso Roberto. **Estrutura de dados e algoritmos**. 2. ed. São Paulo: Futura, 2003.
- Celes, W. Rangel, J. L. **Curso de Estrutura de Dados**. PUC-Rio, 2002.
- W. Celes, R. Cerqueira, J.L. Rangel. **Introdução a Estruturas de Dados - com técnicas de programação em C**. Rio de Janeiro: Campus, 2004.