Anhanguera

Aqui o seu esforço ganha força.

Aula 04 – Pilhas em C

Prof. Esp. Rodrigo Hentz

Pilhas em C

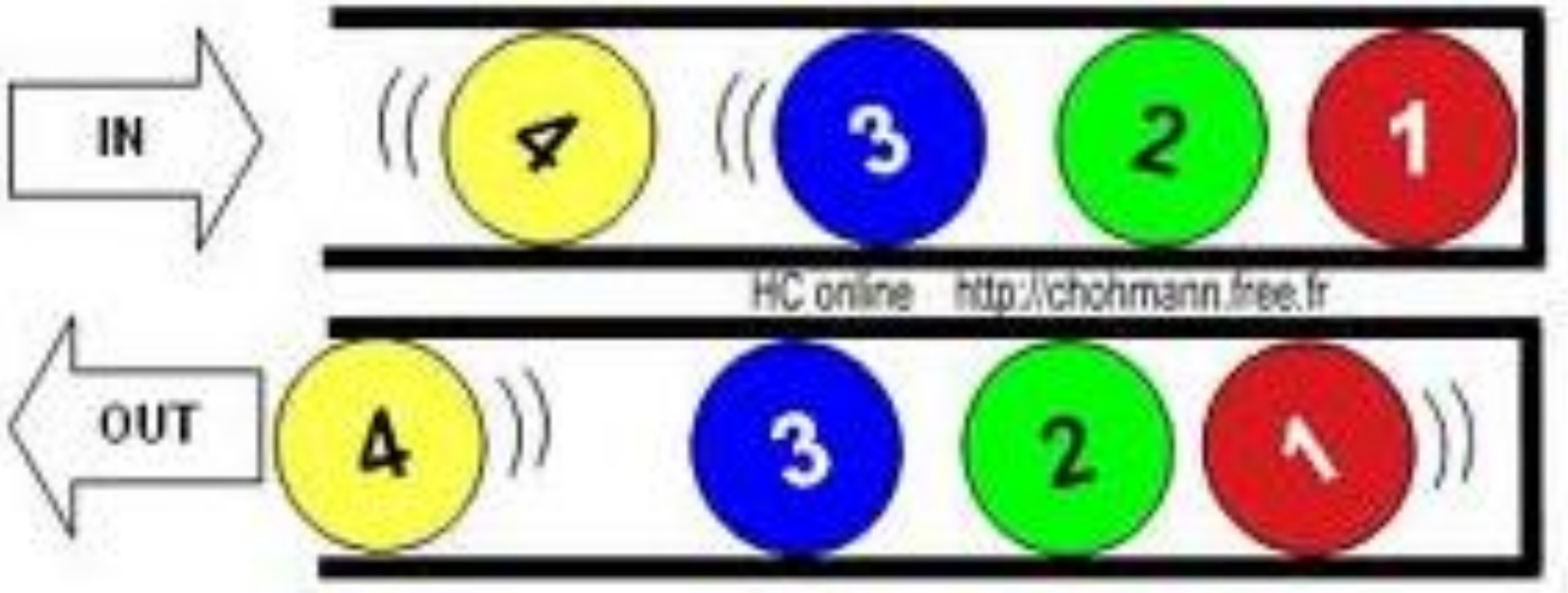
Anhanguera

Definição

- A pilha é um objeto dinâmico que possui um conjunto ordenado de itens
- Novos itens podem ser inseridos e excluídos apenas de seu “topo”.
- Implementa o conceito de LIFO (last-in, first-out) ou UEPS (último a entrar é o primeiro a sair).

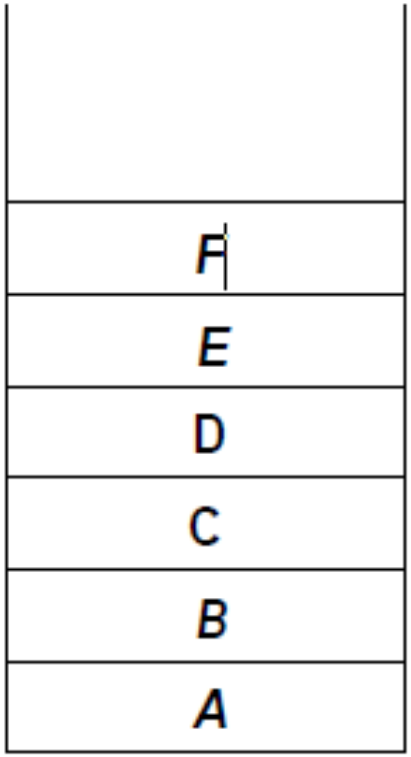
Pilhas em C

LIFO



Pilhas em C

Pilha



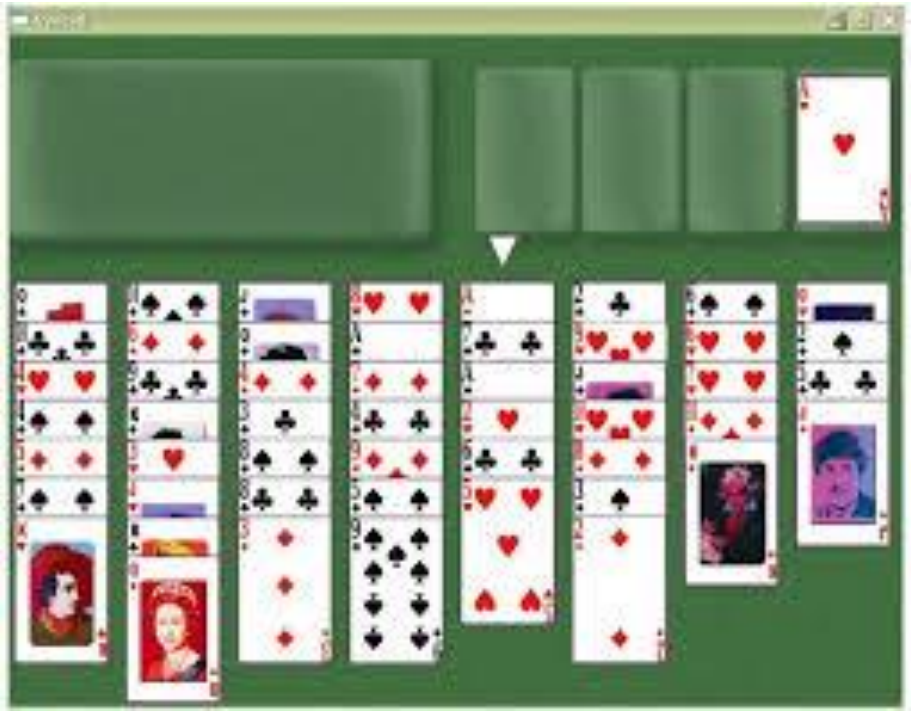
Pilhas em C

Aplicação de uso

```
(Inactive D:\AULAS\ESTRUT~1\PILHA\PILHA3.EXE)

Forneca uma expressao matematica para a analise de separadores:[2+3*{1-2*(2+3)}]

expressao valida!
```

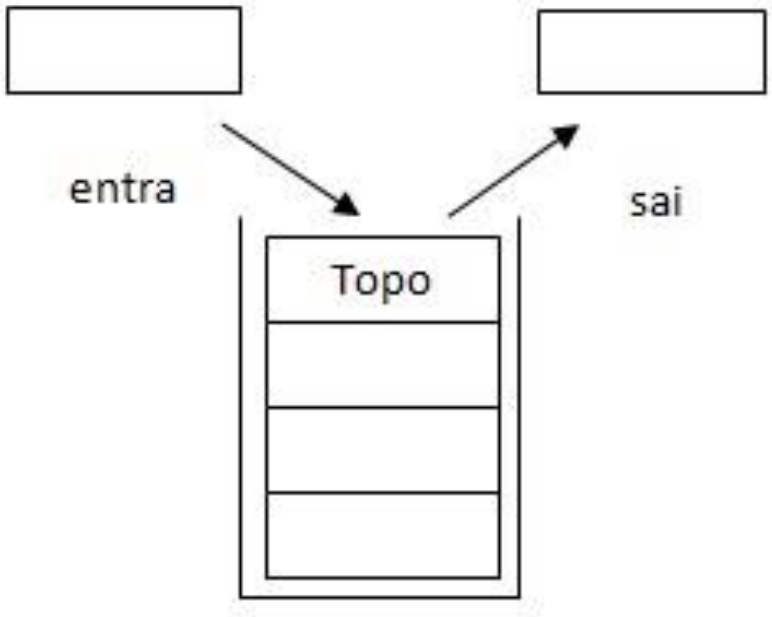


Pilhas em C

Operações

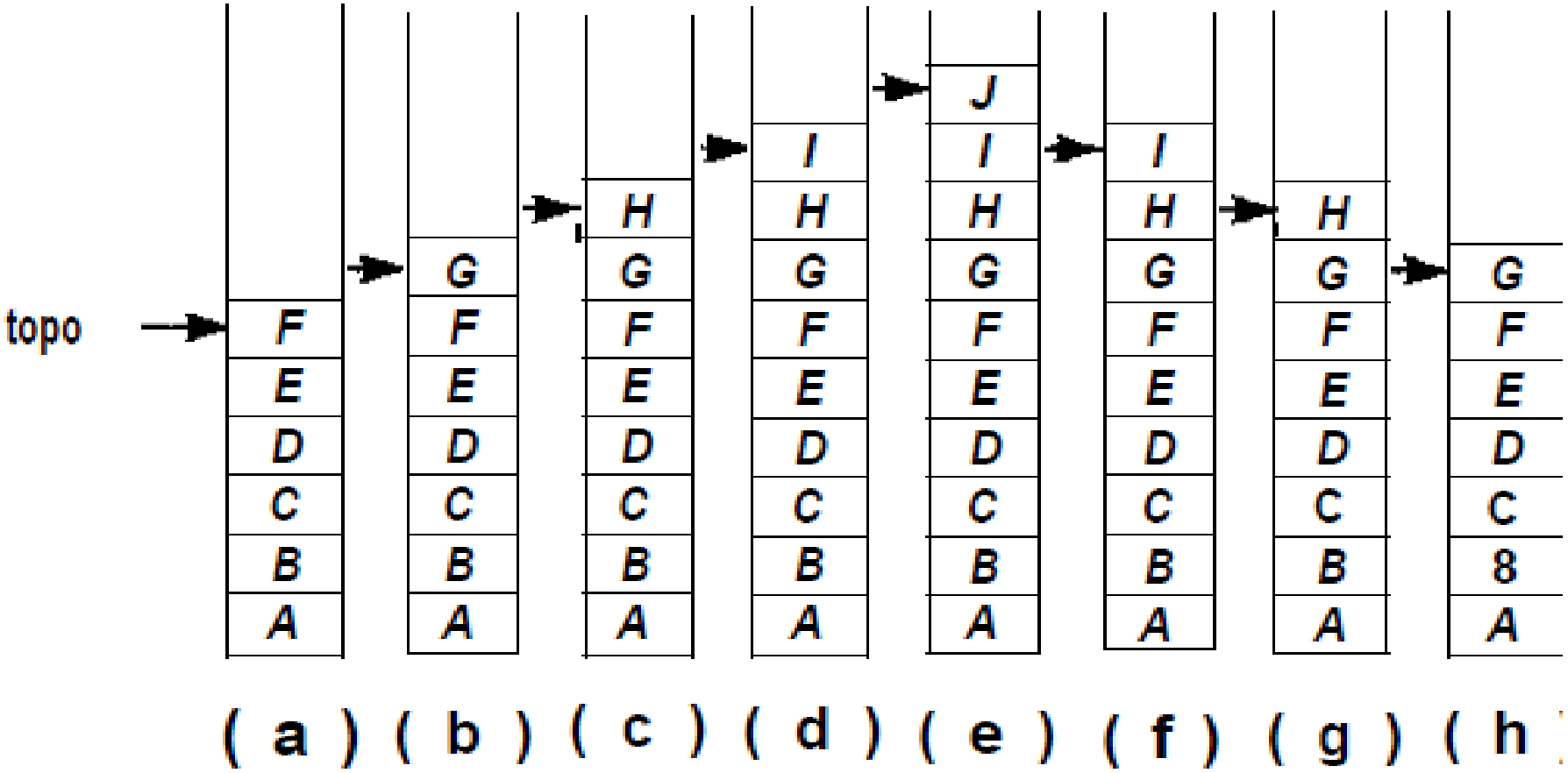
Duas operações principais são realizadas em uma pilha:

- o empilhamento (ou push - entrada)
- e o desempilhamento (ou pop - saída)



Pilhas em C

Exemplo



Pilhas em C

Operações auxiliares para a estrutura de pilha:

- empty(pilha) - Retorna se a pilha está vazia
- stacktop(pilha) - Retorna o item superior da pilha
- inicialize(pilha) - Inicializa uma pilha vazia
- full(pilha) - Retorna se a pilha está cheia

Pilhas em C

Implementação

- Pode ser realizada com vetores ou através de ponteiros.

```
typedef struct {
        int topo;
        int dados [TAM];
} sPilha;

sPilha varPilha;
```

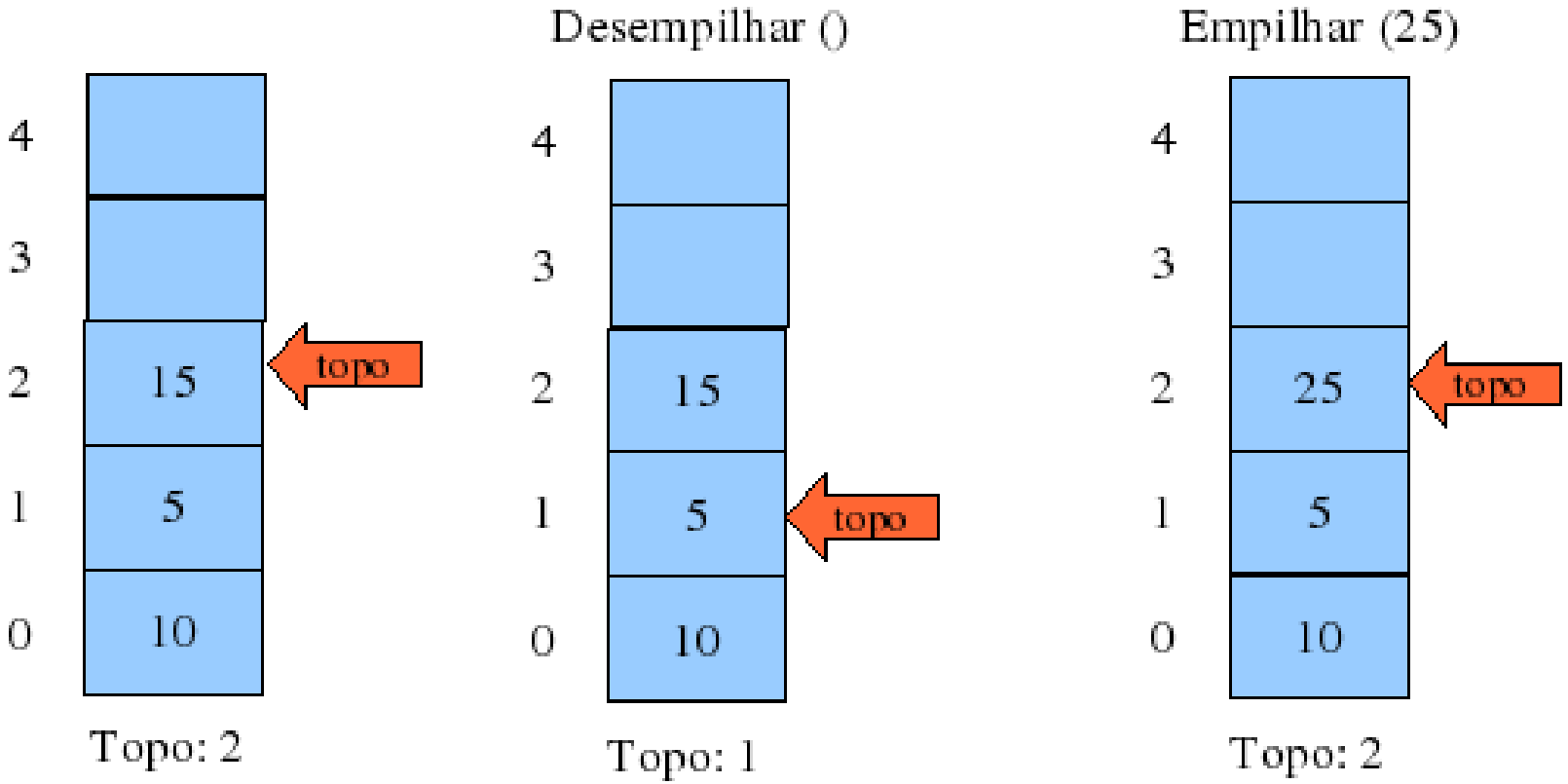
Pilhas em C

Anhanguera

Cuidados

- Quando implementamos pilhas utilizando vetores deve-se ter atenção quando:
  - A pilha já estiver lotada e for solicitado a inclusão de um novo item
  - A pilha estiver vazia e for solicitada uma exclusão
- Assim, temos de ter atenção ao índice que deve controlar a posição da última inserção.

Pilhas em C


4
3
2
15
topo
1
5
0
10
Topo: 2
Desempilhar ()
4
3
2
15
1
5
topo
0
10
Topo: 1
Empilhar (25)
4
3
2
25
topo
1
5
0
10
Topo: 2

Pilhas em C

Anhanguera

Implementação em C utilizando vetores

```c
#define TAM 10

typedef struct
{
    int topo;
    int valores[TAM];
} sPilha;

void inicialize(sPilha* pilha)
{
    pilha->topo = -1;
    printf("\nPilha inicializada");
}

int full(sPilha* pilha)
{
    return pilha->topo == (TAM - 1);
}
```

Pilhas em C

```
int empty(sPilha* pilha)
{
    return  pilha->topo == -1;
}

void push(sPilha* pilha, int valor)
{
    if (full(pilha))
    {
        printf ("\nPilha cheia");
    }
    else
    {
        pilha->topo = pilha->topo + 1;
        pilha->valores[pilha->topo] = valor;
    }
}
```

Pilhas em C

```
int pop(sPilha* pilha)
{
    int num;
    if (empty(pilha))
    {
        printf ("\nPilha esta vazia.");
        num = pilha->topo;
    }
    else
    {
        num = pilha->valores[pilha->topo];
        pilha->topo = pilha->topo - 1;
    }
    return num;
}
```

Pilhas em C

```
int stacktop(sPilha* pilha)
{
    int num;
    if (empty(pilha))
    {
        printf ("\nPilha esta vazia.");
        num = pilha->topo;
    }
    else
    {
        num = pilha->valores[pilha->topo];
    }
    return num;
}
```

Pilhas em C

```
void print(sPilha* pilha)
{
    int i;
    if (empty(pilha)) printf ("\nPilha esta vazia.");
    else
    {
        for(i = pilha->topo; i >= 0; i--)
        {
            printf("\nPilha posicao %d com valor %d .",
                    i,
                    pilha->valores[i]
            );
        }
    }
}
```

Pilhas em C

```
int main(int argc, char *argv[]) {
    sPilha pilha; int opcao, num;
    do
    {
        printf("\n");
        printf("1 - Inicializa\n");
        printf("2 - POP\n");
        printf("3 - PUSH\n");
        printf("4 - STACKTOP\n");
        printf("5 - PRINT\n");
        printf("0 - SAIR\n");
        printf("\nEntre com a opcao: "); scanf("%d", &opcao);
        switch (opcao)
        {
            case 1: inicialize(&pilha);
                break;
            case 2: printf("\nNumero desempilhado %d", pop(&pilha));
                break;
            case 3:
                printf ("\nEntre com o numero para empilhar: ");
                scanf ("%d", &num);
                push(&pilha, num);
                break;
```

Pilhas em C

```
        case 4:
            printf("\nNumero no topo %d", stacktop(&pilha));
            break;
        case 5:
            print(&pilha);
            break;
    }
    fflush(stdin);
} while (opcao != 0);
return 0;
```

Finalização