



UFRJ



INSTITUTO DE
COMPUTAÇÃO
UFRJ

Programação de Computadores II

Laboratório - Pilhas

Profa. Giseli Rabello Lopes

Instruções para resolução dos exercícios

- Sempre que possível, faça uso das funções já definidas referentes às operações básicas sobre a pilha, vistas em aula, para resolução dos exercícios a seguir.
- Caso seja necessário, pode-se realizar mudanças para a pilha armazenar outros tipos de dados.

Implementação da Pilha

```
#include <stdio.h>
#include <stdlib.h>

#define true 1
#define false 0
typedef int bool;

typedef int TIPOCHAVE;

typedef struct {
    TIPOCHAVE chave;
    //outros campos...
} REGISTRO;

typedef struct aux {
    REGISTRO reg;
    struct aux* prox;
} ELEMENTO;

typedef ELEMENTO* PONT;

typedef struct {
    PONT topo;
} PILHA;

void inicializarPilha(PILHA* p) {
    p->topo = NULL;
}

int tamanho(PILHA* p) {
    PONT end = p->topo;
    int tam = 0;
    while (end != NULL) {
        tam++;
        end = end->prox;
    }
    return tam;
}

bool estaVazia(PILHA* p) {
    if (p->topo == NULL)
        return true;
    return false;
}
```

```
void exibirPilha(PILHA* p) {
    PONT end = p->topo;
    printf("Pilha: \n ");
    while (end != NULL) {
        printf("%i ", end->reg.chave);
        end = end->prox;
    }
    printf("\n\n");
}

bool inserirElemPilha(PILHA* p, REGISTRO reg)
{
    PONT novo = (PONT) malloc(sizeof(ELEMENTO));
    novo->reg = reg;
    novo->prox = p->topo;
    p->topo = novo;
    return true;
}

bool excluirElemPilha(PILHA* p, REGISTRO* reg)
{
    if (p->topo == NULL) return false;
    *reg = p->topo->reg;
    PONT apagar = p->topo;
    p->topo = p->topo->prox;
    free(apagar);
    return true;
}

void reinicializarPilha(PILHA* p) {
    PONT apagar;
    PONT posicao = p->topo;
    while (posicao != NULL) {
        apagar = posicao;
        posicao = posicao->prox;
        free(apagar);
    }
    p->topo = NULL;
}
```

Exercício 1

- A conversão de números inteiros, na base 10, para outras bases numéricas se dá através de sucessivas divisões de um dado valor n pelo valor da base na qual se queira converter.

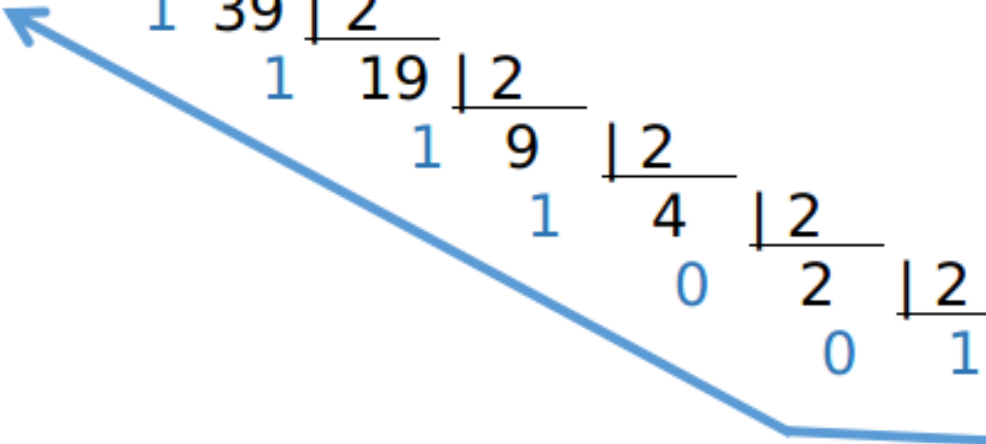
Exercício 1

- Exemplo:

$$159_{10} = X_2$$

$$\begin{array}{r} 159 \mid 2 \\ \hline 1 \quad 79 \mid 2 \\ \hline 1 \quad 39 \mid 2 \\ \hline 1 \quad 19 \mid 2 \\ \hline 1 \quad 9 \mid 2 \\ \hline 1 \quad 4 \mid 2 \\ \hline 0 \quad 2 \mid 2 \\ \hline 0 \quad 1 \end{array}$$

$X_2 = 10011111_2$



Exercício 1

- Faça uma função para obter a conversão numérica, utilizando uma pilha, de Decimal (base 10) para Binário (base 2).
- Teste sua função, criando um programa principal que a utilize.

Exercício 2

- Faça um programa que calcule o resultado de uma expressão em notação pós-fixada, recebida através de um arquivo texto, utilizando uma pilha.
- Considere que os operandos e resultados das operações realizadas serão todos inteiros.

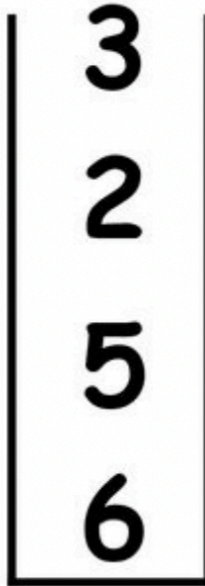
Exercício 2 - Exemplo

6 5 2 3 + 8 * - 3 + * = ?

↑ ↑ ↑ ↑ ↑

input: 6523

Push : 6, 5, 2, 3



S

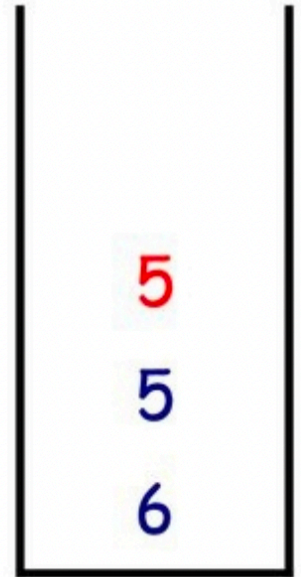
input: +

o2 <- pop#1

o1 <- pop#2

push +

5



S

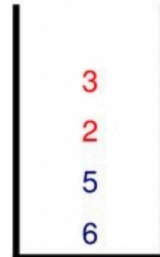
What 's next ?

Exercício 2 - Exemplo

6 5 2 3 + 8 * - 3 + * = ?

input: 6523

Push : 6 5 2 3

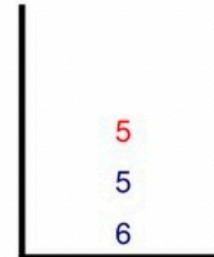


input: +

pop#1 → 3

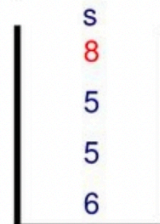
pop#2 → 2

push 2+3



input: 8

push 8

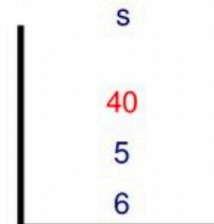


input: *

pop#1 → 8

pop#2 → 5

push 5*8

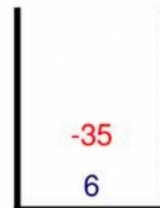


input: -

pop#1 → 40

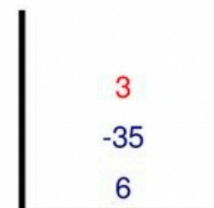
pop#2 → 5

push 5-40



input: 3

push 3

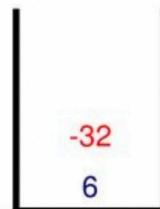


input: +

pop#1 → 3

pop#2 → -35

push -35+3



input: *

pop#1 → -32

pop#2 → 6

push 6 * -32



Exercício 2

- Seu programa deverá exibir quais operações estão sendo realizadas e o resultado final.
- Exemplo:

Arquivo de Entrada:

6 5 2 3 + 8 * - 3 + *

Saída:

2 + 3 = 5

5 * 8 = 40

5 - 40 = -35

-35 + 3 = -32

6 * -32 = -192

Resultado: -192

Exercício 3

- Escreva uma função que verifique se uma palavra é um palíndromo (se pode ler, indiferentemente, da esquerda para a direita ou vice-versa).
- Utilize uma pilha que armazene caracteres em sua solução.
- Faça um programa principal que use a função.
- Exemplos de palíndromos:
 - `ala; ama; anilina; arara; ata; ele; esse; mamam; matam; metem; mirim; oco; osso; ovo; radar; raiar; ralar; reger; reler; reter; rever; reviver; rir; rodador; sacas; saias; salas; socos; sós.`

Exercício 4

- Escreva uma função para verificar se uma expressão matemática tem os parênteses, colchetes e chaves balanceados e agrupados de forma correta.
- Exemplos de expressões **incorretas**:
 - $((A+B) \text{ ou } A+B (\text{ ou } (A+B - [C+D)$
 - $) A+B (-C \text{ ou } (A+B)) - (C+D \text{ ou }] A+B (-C$
 - $[A+B (C-D)] \text{ ou } (A+B * \{D/E\}]$
- Utilize uma pilha que armazene caracteres em sua solução.
- Faça um programa principal que use a função.