



**Atividade EXTRA - AV2 - Algoritmos e Estruturas de Dados - 2,0 pontos**

**1. QUESTÕES**

Com base no código apresentado (que apresenta uma Pilha de 20 elementos do tipo inteiro utilizando struct, bem como 3 funções: empilha, desempilha e imprimePilha) na página a seguir, responda:

1. Escreva o código que execute os passos apresentados abaixo:

Inserir o número 10 na pilha P1

Inserir o número 20 na Pilha P1

Inserir o número 30 na Pilha P1

Inserir o número 40 na Pilha P1

Remover.

Imprimir a Pilha P1

Criar uma pilha P2.

Inserir o número 55 na Pilha P2

Inserir o número 66 na Pilha P2

Imprimir a Pilha P2

2. Dado uma pilha que armazene números, escreva uma função que informe o maior, o menor e a média aritmética dos elementos da pilha.

```
void maior_menor_media(pilha *p1);
```

3. Faça uma função que receba uma pilha como argumento e retorne o valor armazenado em seu topo. A função deve remover também esse elemento.

```
int retorna_remove_topo(pilha *p1);
```

4. Desenvolva uma função que informe se uma pilha P1 tem mais elementos que uma pilha P2 ou não.

```
void mais_elementos(pilha *p1, pilha *p2);
```



```
#include <stdio.h>
#include <stdlib.h>
#define TAMANHO_PILHA 20
typedef struct{
    int vetor[TAMANHO_PILHA]; //tamanho da pilha
    int topo;
}Pilha;

//prototipo da função empilha
void empilha(int valor,Pilha *P1){
    //pilha->topo significa: ponteiro "pilha" apontando para CONTEÚDO de um item de uma struct
    if(P1->topo < TAMANHO_PILHA){ //verificando se a pilha não está cheia
        //daí pode empilhar
        P1->vetor[P1->topo]=valor;
        P1->topo++;
        printf("O valor %d foi adicionado \n",valor);
    }else{
        printf("Nao ha mais espaco na pilha, \n");
    }
}

void desempilha(Pilha *P1){
    if(P1->topo > 0){
        P1->topo--; //desempilha
        printf("Elemento retirado: %d. \n",P1->vetor[P1->topo]);
    }else{
        printf("A pilha está vazia. \n"); //pilha vazia
    }
}

void imprimePilha(Pilha *P1){
    int i;
    printf("\nSegue Impressao da Pilha: \n");
    for(i=((P1->topo)-1);i>=0;i--){ //valor inicial de i é a ultima posição da pilha e daí decrementa
        printf("\t %d \n",P1->vetor[i]);
    }
}

int main() {
    //DECLARA UMA PILHA
    Pilha P1;
    P1.topo=0; // o topo da pilha deve começar em zero
}
```