



1. INFORMAÇÕES SOBRE A ATIVIDADE

| | | |
|--|-----------------------------|----------------|
| Curso: | Turma: | Turno: Noturno |
| Disciplina: Algoritmos e Estruturas de Dados | Data: Até dia 15/07/2023 | Prova: AV3 |
| Aluno(a): | | |

AV3 – ALGORITMOS E ESTRUTURAS DE DADOS

As respostas devem ser colocadas em um único Arquivo e enviadas via SIGAA na aba da Tarefa AV3.

2. QUESTÕES

1. Com base no código apresentado (que utiliza Pilha de 20 elementos do tipo inteiro utilizando struct, bem como 3 funções: empilha, desempilha e imprimePilha) na página a seguir, responda: Adicione um menu para que o usuário possa escolher se quer adicionar um elemento à pilha, ou remover um elemento, etc, com base nas instruções abaixo:

- Se pressionar 'a', o programa solicita um número a ser adicionado à pilha
- Se pressionar 'r', o programa remove um elemento da pilha
- Se pressionar 's', o programa exibe os elementos já presentes na pilha
- Se pressionar 't', o programa exibe o número de elementos
- Se pressionar 'x', encerra o programa

```
Escolha uma opção:  
a: adiciona elemento  
r: remove elemento  
s: exibe elementos  
...
```



```
#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
}
```



2. Faça um programa que implemente uma fila de 20 elementos do tipo inteiro utilizando struct. Para isso, siga as instruções abaixo:

- **Faça um programa que implemente uma fila de 20 elementos do tipo inteiro utilizando struct**
 - **Implemente um método que insira na fila um novo inteiro**
 - O método deve enviar uma mensagem se a fila estiver cheia
 - void adiciona(int valor, Fila *fila);
 - **Implemente um método que remova o primeiro elemento da fila**
 - O método deve exibir uma mensagem se a fila estiver vazia
 - void retira(Fila *fila);
 - **Implemente um método que retorne 1 se a fila está cheia e 0 se não**
 - int isCheia(Fila *fila);
 - **Implemente um método que retorne 1 se a fila está vazia e 0 se não**
 - int isVazia(Fila *fila);

3. Dado uma fila que armazene números, escreva 3 funções que informem o maior, o menor e a média aritmética dos elementos da pilha.

- a) void maior(fila *fila);
- b) void menor(fila *fila);
- c) void media(fila *fila);

4. Insira os seguintes valores em uma árvore binária, coloque os fatores de balanceamento e diga se é ou não uma AVL e qual nó está desbalanceado:

- a) [30,15, 50, 5,10, 20]
- b) [80, 40, 100, 120, 90, 30]
- c) [10, 50, 4, 90, 20, 8]

5. Insiram os seguintes valores e depois rotacione para a direita a partir da raiz:

- a) [40,30, 20]
- b) [40, 30, 20, 35]
- c) [40, 50, 30, 20, 35]



6. Insiram os seguintes valores e depois rotacione dupla a direita a partir da raiz:

- a) [40, 20, 30]
- b) [40, 20, 30, 50]
- c) [40, 20, 30, 10, 50, 80]

7. Faça uma AVL que apresente os dados (obs: Você pode utilizar o link dado):

- a) [DD, MM, AA, 15, 22, 01, 45, 68, 87, 92, 15]
- b) [DD, MM, AA, 02, 13, 15, 48, 72, 78, 29, 51]
- c) [DD, MM, AA, 51, 44, 08, 54, 86, 55, 38, 92]

Apresentem o passo a passo da AVL ao incluir, excluir e rotacionar nós (pode ser através de prints da tela, evidenciando as rotações realizadas no processo. OBS: AA: Dia do seu Nascimento; MM: mês do seu nascimento; AA: Ano do seu nascimento com 2 dígitos.

LINK: <https://visualgo.net/en/bst>