

PROFESSOR: Dr Marcos Camada

ESTUDANTE: Oscar Borges

```
#include <stdio.h>
#include <stdlib.h>
#define tam 10

struct Pilha {
    char dados [tam];
    int qtd;
};

struct Pilha* criarPilha() {
    struct Pilha *pilha;
    pilha = (struct Pilha*)malloc(sizeof(struct Pilha));
    if(pilha != NULL) {
        pilha->qtd = 0;
    }
    return pilha;
}

void excluirPilha(struct Pilha *pilha) {
    free(pilha);
}

int tamanhoPilha(struct Pilha *pilha) {
    if(pilha == NULL) return -1;
    else return pilha->qtd;
}

int pilhaCheia(struct Pilha *pilha) {
    if(pilha == NULL) return -1;
    else return (pilha->qtd == tam);
}

int pilhaVazia(struct Pilha *pilha) {
    if(pilha == NULL) return -1;
    else return (pilha->qtd == 0);
}

int push(struct Pilha *pilha, char dado) {
    if(pilha == NULL) {
        printf("Pilha não existente\n");
    }
}
```

```

        return 0;
    };
    if(pilhaCheia(pilha)) {
        printf("Impossivel adicionar '%c'. Pilha Cheia!\n", dado);
        return 0;
    };
    pilha->dados[pilha->qtd] = dado;
    pilha->qtd++;
    printf("Dado '%c' inserido!\n", dado);
    return 1;
}

int pop(struct Pilha* pilha) {
    if(pilha == NULL || pilhaVazia(pilha)) {
        printf("Pilha não existe ou está vazia. Nada para ser
removido!\n");
        return 0;
    };
    printf("Dado '%c' removido!\n", pilha->dados[pilha->qtd-1]);
    pilha->qtd--;
    return 1;
}

void verTopo(struct Pilha* pilha) {
    if(pilhaVazia(pilha)) printf("Pilha vazia. Não existe topo!\n");
    else printf("Topo da pilha é o: %c\n", pilha->dados[pilha->qtd-1]);
}

int main(int argc, char const *argv[])
{
    struct Pilha *pilha = criarPilha();
    push(pilha, 'a');
    verTopo(pilha);
    push(pilha, 'b');
    verTopo(pilha);
    push(pilha, 'c');
    verTopo(pilha);
    push(pilha, 'd');
    verTopo(pilha);
    push(pilha, 'e');
    verTopo(pilha);
    push(pilha, 'f');
    verTopo(pilha);
}

```

[illegible]