

```

#include<stdio.h>
#include<stdlib.h> //malloc e free
#include<stdbool.h>

typedef int Eduardo;

typedef struct no{
    int dado;
    struct no * prox;
}tno;

typedef tno * tpilha;
//typedef struct no * tpilha;

void criar(tpilha * pp){
    * pp = NULL;
}

bool push(tpilha * pp, Eduardo valor){
    tpilha novo;
    novo = (tno *) malloc(sizeof(tno));
    if(novo == NULL)
        return false;
    else{
        novo->dado = valor;
        novo->prox = * pp;
        *pp = novo;
        return true;
    }
}

bool pop(tpilha * pp, Eduardo * ppegaValor){
    tpilha aux;
    if(*pp == NULL)
        return false;
    else{
        aux = *pp;
        *ppegaValor = aux->dado; // *ppegaValor = (*pp)->dado
        *pp = aux->prox; // *pp = (*pp)->prox;
        free(aux);
        return true;
    }
}

int main(){

    tpilha p;          //p é uma variável do tipo ponteiro para
    acessar os dados de uma struct no
    Eduardo pegaValor, valor, op; //variavel do tipo inteiro

    criar(&p);

    do{
        printf("Entre com a opcao 1-Push e 2-Pop 3-Inverter a
        Pilha (-1 Sair)!");
        scanf("%d", &op);
    }
}

```

```

switch(op){
    case 1: do{
        printf("Informe o
valor para Push (-1 Sair) : ");
        scanf("%d", &valor);
        if(valor ==
-1)
            printf("Saindo...\n");

        if(push(&p, valor))
            printf("Pilha[%d] \n", p->dado);

        printf("Erro: Memoria Insuficiente!\n");
        valor = -1;
    }

    case 2: do{
        printf("Entre
com o valor 1 para Pop ou (-1 Sair) : ");
        scanf("%d",
&valor);
        if(valor == 1){
            if(pop(&p, &pegaValor))
                printf("Lido valor do topo da pilha: %d \n", pegaValor);
            else
                printf("Pilha Vazia! \n");

            printf("Valor invalido! \n");
        }
        else{
            }
        }
        while(valor != -1);
        break;
    }
}
while(op != -1);
return 0;
}

```