

```

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

struct Pilha{

    int topo; /* posição elemento topo */
    int capa;
    float *pElem;

};

void criarpilha( struct Pilha *p, int c){

    p->topo = -1;
    p->capa = c;
    p->pElem = (float*) malloc (c*sizeof(float));

}

int estavazia ( struct Pilha *p){

    if(p-> topo ==-1)

        return 1; //true

    else

        return 0; //false

}

int estacheia (struct Pilha *p){

    if (p->topo == p->capa -1)

        return 1;

    else

        return 0;

}

void empilhar ( struct Pilha *p, float v){

    p->topo ++;
    p->pElem [p->topo] = v;

}

float desempilhar (struct Pilha *p){

    float aux = p->pElem [p->topo];
    p->topo--;
    return aux;

}

float retornatopo ( struct Pilha *p){

    return p->pElem [p->topo];

}

int main(){

    struct Pilha minhapilha;
    int capacidade, op;
    float valor;

    printf( "\nCapacidade da pilha?" );
    scanf( "%d", &capacidade );

    criarpilha (&minhapilha, capacidade);

    while(1){/* loop infinito */

        printf("\n1- empilhar (push)\n");
        printf("2- desempilhar (POP)\n");
        printf("3- Mostrar o topo \n");
        printf("4- sair\n");

        printf("\nopcao?");
        scanf("%d", &op);

        switch (op){

```

```

case 1: //push

    if( estacheia( &minhapilha ) == 1)

        printf("\nPILHA CHEIA!\n");

    else{

        printf("\nVALOR?");
        scanf("%f", &valor);
        empilhar (&minhapilha, valor);

    }
    break;

case 2: //pop

    if ( estavazia(&minhapilha) == 1 )

        printf( "\nPILHA VAZIA! \n");

    else{

        valor = desempilhar (&minhapilha);
        printf ( "\n%.1f DESEMPILHADO!\n", valor );

    }
    break;

case 3: //mostrar o topo

    if ( estavazia (&minhapilha) == 1)

        printf( "\nPILHA VAZIA!\n" );

    else{

        valor = retornatopo (&minhapilha);
        printf ( "\nTOPO: %.1f\n", valor );

    }
    break;

case 4:

    exit(0);

default: printf("\nOPCAO INVALIDA! \n");

}

}

}

```