

```
// pilha.h - implementacao dinamica sequencial de pilha

#include <stdlib.h>

typedef int Itemp;

typedef struct pilha {
    int    max;
    int    topo;
    Itemp *item;
} *Pilha;

Pilha pilha(int m) {
    Pilha P = malloc(sizeof(struct pilha));
    P->max = m;
    P->topo = -1;
    P->item = malloc(m*sizeof(Itemp));
    return P;
}

int vaziap(Pilha P) {
    if( P->topo == -1 ) return 1;
    else return 0;
}

int cheiap(Pilha P) {
    if( P->topo == P->max-1 ) return 1;
    else return 0;
}

void empilha(Itemp x, Pilha P) {
    if( cheiap(P) ) { puts("pilha cheia!"); abort(); }
    P->topo++;
    P->item[P->topo] = x;
}

Itemp desempilha(Pilha P) {
    if( vaziap(P) ) { puts("pilha vazia!"); abort(); }
    Itemp x = P->item[P->topo];
    P->topo--;
    return x;
}

Itemp topo(Pilha P) {
    if( vaziap(P) ) { puts("pilha vazia!"); abort(); }
    return P->item[P->topo];
}

void destroip(Pilha *Q) {
    free((*Q)->item);
    free(*Q);
    *Q = NULL;
}
```