#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

// struct \*NoPilha ->PtroNoPilha

typedef struct NoPilha \*PtrNoPilha;

typedef struct // uso para poder usar PtrNoPilha proximo;

{

int x;

PtrNoPilha proximo;

// struct NoPilha\* Proximo (eq a de cima mas precisamos do outro typedef)

} NoPilha;

typedef struct

{

/\* topo e contador \*/

int qnde; // quantidade de elementos

PtrNoPilha topo; // aponta para um no de pilha

} PilhaDinamica;

//----------INICIALIZAR--------------

void iniciapilhadinamica(PilhaDinamica \*pilha)

{

pilha->qnde = 0;

pilha->topo = NULL;

}

//----------DESTRUIR--------------

void destruir(PilhaDinamica \*pilha)

{

free(pilha);

}

//----------INSERÇÃO (PUSH)--------------

void insercao(PilhaDinamica \*pilha, int novo\_nro)

{

/\*considerar se é primeira inseção ou não

1 topo -> NULL

2 topo \_> algum outro{

criar novo n´

copiar valor no novo no

atualizar referencias

-novo->proximo = (topo)

-topo->novo

qde++

}

\*/

PtrNoPilha novo;

novo = malloc(sizeof(PtrNoPilha));

novo->valor = valor; //copia o valor p dentro do bloco

//passo 4

novo->proximo = pilha->topo; // Novo-proximo -> topo

pilha->topo = novo;

pilha->qnde++;

}

//----------REMOVER (POP)--------------

int remover (PilhaDinamica \*pilha){

int ret ;

// ver se ta vazia

if(estavazia(pilha)){

printf("TA VAZIA");

}else{

//salvar valor

ret = pilha->topo->x;

//criar ptr aux

PtrNoPilha aux;

//att topo

//desaloc memo

free(aux);

//qnde --

pilha->qnde--;

}

//ret

return ret;

}

//----------PROCURAR (TOP)--------------

//----------ESTA VAZIA--------------

bool estavazia(PilhaDinamica \*pilha)

{

return (pilha->topo == NULL);

}

//----------TAMANHO--------------

int tamanhopilha(PilhaDinamica \*pilha)

{

return (pilha->qnde);

}

//----------imprimir pilha--------------

void imprimir(PilhaDinamica \*pilha)

{

PtrNoPilha var = pilha->topo;

printf("[");

/\*

enquanto var !=null

imprimir valor do nó atual

var recebe o proximo

\*/

while (var != NULL)

{

printf("%d", var->x);

var = var->proximo;

}

printf("]");

}

int main(int argc, char const \*argv[])

{

PilhaDinamica p;

iniciapilhadinamica(&p);

if (estavazia(&p))

{

printf("Vazia");

}

imprimir(&p);

return 0;

}