

```

/*
Exercício 1

Escreva uma função em C que permita ler um conjunto de valores inteiros para
um vector utilizando apontadores. Não se conhecendo à partida quantos valores
inteiros serão lidos, considere a alocação dinâmica de memória. Escreva uma
outra função em C que permite procurar no vector por um determinado valor.
Teste a duas funções.

*/

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

#define BLOCO_INI 10
#define BLOCO_ACR 5

enum BOOL { FALSE=0, TRUE };

int * lerInt( int * sz )
{
    int *pvector, num = BLOCO_INI, count = 0, ival;

    /* reservar memoria */
    if( (pvector = (int *)malloc( sizeof(int) * num )) == NULL ){
        printf("Memoria insuficiente!\n");
        return NULL;
    }

    do{
        printf("Valor (-1 terminar):");scanf("%d", &ival );
        if ( ival != -1 ) {
            *( pvector + count++ ) = ival;

            /* realocar memoria */
            if ( count == num ) {
                num += BLOCO_ACR;

                if( (pvector = (int *)realloc( pvector, sizeof(int) * num )) == NULL
){
                    printf("Memoria insuficiente!\n");
                    return NULL;
                }
            }
        }
    } while ( ival != -1 );

    *sz=count;

    return pvector;
}

enum BOOL pesqInt( int * pvect , int sz , int valor)
{
    enum BOOL ret=FALSE;

    int i;

    for(i=0;i<sz;i++)
        if( pvect[i] == valor ) {
            ret=TRUE;
            break;
        }

    return ret;
}

```