Aluno: Erivaldo Barros (Neto)

Crie um vetor com ponteiros utilizando alocação dinâmica na linguagem C, que:

- use a função realloc;

- use a função sizeof;

- que tenha tamanho 22 de vetor;

- depois libere o bloco utilizando a função free.

*Código abaixo está escrito na linguagem* ***C.***

**#include <stdio.h>** *//biblioteca para entrada e saída de dados*

**#include <stdlib.h>** *//biblioteca para as funções de alocação dinâmicas*

**#include <locale.h>** *//biblioteca para utilizar caracteres especiais nas strings.*

**int main() {**

**setlocale(LC\_ALL, "Portuguese");** *//definindo o tipo de idioma utilizado nas strings.*

**int i, \*ptr, \*ptr2;** *//Criando ponteiro do tipo inteiro \*ptr.*

**ptr = (int \*) malloc(5 \* sizeof(int));** *//alocando 5 posições na memória.*

**if (ptr == NULL){**

**printf(“Sem memória!\n”);**

**exit(1);***//termina o programa.*

**}**

**printf("\*---Valores referente ao ponteiro de 5 posições---\*\n\n");**

**for (i=0; i<5; i++) {**

**ptr[i] = i\*2;**

**printf("Ponteiro na posição [%d] com valor de: %d\n",i,ptr[i]);**

**}**

**ptr2 = (int \*) realloc(ptr,22\*sizeof(int));** *//realocando 22 posições na memória.*

**printf("\n\*---Valores referente ao ponteiro de 22 posições---\*\n\n");**

*//a variavel ptr2 serviu apenas como uma forma de segurança ou bkp do ptr, pois se eu tivesse colocado ptr para realloc também, as informações poderiam ser perdidas caso ele desse NULL. Então essa leve verificação pelo menos garante que ptr fique intacto caso o ptr2 seja NULL, se não, ele apenas reatribui o valor do ptr2 para o ptr e o programa prossegue normalmente.*

**if (ptr2 != NULL){**

**ptr = ptr2;**

**}**

**for (i=0; i<22; i++) {**

**ptr2[i] = i\*2;**

**printf("Ponteiro na posição [%d] com valor de: %d\n",i,ptr2[i]);**

**}**

*//obs: o ptr[i]\*2 é apenas ilustrativo para demonstrar um valor qualquer dentro da posição do vetor descrito anterior a ele.*

**free(ptr);** *//liberando o bloco.*

**return 0;**

**}**