

RESOLUÇÃO DA LISTA 5

1) Escreva um programa que contenha duas variáveis inteiras. Compare seus endereços e exiba o maior endereço.

```
#include<stdio.h>

#include<stdlib.h>

int main() {

printf("\tLISTA 5 - Questao 1 \n\n");

    int a,b;

    int *p,*q;

    p=&a;

    q=&b;

    printf("\nO endereco de A eh %x",p);

    printf("\nO endereco de B eh %x",q);

    if(p>q)

        printf("\n\nO maior endereco eh de A: %x",p);

    else printf("\n\nO maior endereco eh de B: %x",q);

    return 0;

}
```

2) Crie um programa que contenha um array de float contendo 10 elementos. Imprima o endereço de cada posição desse array.

```
#include<stdio.h>

#include<stdlib.h>

int main() {

printf("\tLISTA 5 - Questao 2 \n\n");

    float a[10];

    int i;
```

}

endereço de cada posição dessa matriz.

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

elementos com um ponteiro.

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
printf("\tLISTA 5 - Questao 4 \n\n");
```

```
int v[5]={1,2,3,4,5};
```

```
int *p;
```

```

int i;

for(i=4;i>=0;i--){

    p=&v[i];

    printf("\nVetor[%d]:%d ender:%x",i,v[i],p);

}

return 0;

}

```

5) Crie um programa que:

- (a) Aloque dinamicamente um array de 5 números inteiros;
- (b) Peça para o usuário digitar os 5 números no espaço alocado;
- (c) Mostre na tela os 5 números;
- (d) Libere a memória alocada.

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```

int main() {

printf("\tLISTA 5 - Questao 5 \n\n");

int *p= (int*)malloc(5*sizeof(int));

int i;

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

    printf("\nAdicione um numero inteiro: ");

    scanf("%d",&p[i]);

}

getchar();

printf("\nNumeros inseridos - Aperte enter para exibir ");

getchar();

printf("\n NUMEROS INSERIDOS \n");

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

    printf("\nVetor[%d]: %d",i,p[i]);

printf("\n\nAperte enter para limpar a memoria ");

getchar();

p=(int*) realloc(NULL,0);

printf("\nMemoria Liberada\n");

```

```

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

printf("\nVetor[%d]: %d",i,p[i]);


        return 0;

}

```

6) Faça um programa que leia uma quantidade qualquer de números armazenando-os na memória e pare a leitura quando o usuário entrar com um número negativo. Em seguida, imprima o vetor lido. Use a função REALLOC.

```

#include<stdio.h>

#include<stdlib.h>


int main() {

printf("\tLISTA 5 - Questao 6 \n\n");

int *p;

int j,i=1,opcao;

p=(int*) malloc(10);

p=(int*) realloc(NULL,0);

while(opcao>=0){

    p=(int*) realloc(p,i);

    printf("Entre com um numero inteiro: ");

    scanf("%d",&p[i]);

    opcao=p[i];

    i++;

}

printf("\n*** VETOR ***\n");

for(j=1;j<i-1;j++)

    printf("\nPosicao [%d]: %d",j,p[j]);

printf("\n");

return 0;

}

```

7) Crie um programa que declare uma estrutura (registro) para o cadastro de alunos.

- Deverão ser armazenados, para cada aluno: matrícula, sobrenome (apenas um) e ano do nascimento.

- Ao início do programa, o usuário deverá informar o número de alunos que serão armazenados.

- O programa deverá alocar dinamicamente a quantidade necessária de memória para armazenar os registros dos alunos.

- O programa deverá pedir ao usuário que entre com as informações dos alunos.

- Ao final, mostrar os dados armazenados e liberar a memória alocada.

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
struct aluno{
```

```
    int mat;
```

```
    char sob[10];
```

```
    int ano;
```

```
};
```

```
int main() {
```

```
printf("\tLISTA 5 - Questao 7 \n\n");
```

```
int num,i,j;
```

```
printf("Quantos alunos serao cadastrados?: ");
```

```
scanf("%d",&num);
```

```
struct aluno *p=(struct aluno*) malloc(num*sizeof(struct aluno));
```

```
for(i=0;i<num;i++){
```

```
    printf("\nEntre com a matricula: ");
```

```
    scanf("%d",&p[i].mat);
```

```
    printf("Entre com o sobrenome: ");
```

```
    scanf("%s",&p[i].sob);
```

```
    printf("Entre com o ano de nascimento: ");
```

```
    scanf("%d",&p[i].ano);
```

```
}
```

```
getchar();
```

```
printf("\n");
```

```
printf("\n\t*** EXIBIR ALUNOS ***");

printf("\n");

for(j=0;j<num;j++)

    printf("\nMatricula: %d | Sobrenome: %s | Ano de Nascimento: %d\n",p[j].mat,p[j].sob,p[j].ano);

printf("\nApertar enter para limpar a memoria ");

getchar();

p=(struct aluno*) realloc(NULL,0);

printf("\n\n\t*** Memoria Liberada ***\n");

for(j=0;j<num;j++)

    printf("\nMatricula: %d | Sobrenome: %s | Ano de Nascimento: %d\n",p[j].mat,p[j].sob,p[j].ano);


    return 0;

}
```