

LISTA DE EXERCÍCIOS PONTEIROS E ALOCAÇÃO DINÂMICA

- 1) Crie uma função que receba como parâmetros dois vetores de inteiros: x1 e x2 e as suas respectivas quantidades de elementos: n1 e n2. A função deverá retornar um ponteiro para um terceiro vetor, x3, alocado dinamicamente, contendo a união de x1 e x2 e usar o ponteiro qtd para retornar o tamanho de x3.

Sendo x1 = {1, 3, 5, 6, 7} e x2 = {1, 3, 4, 6, 8}, x3 irá conter {1, 3, 4, 5, 6, 7, 8}.

Assinatura da função:

```
int* uniao(int *x1, int *x2, int n1, int n2, int* qtd);
```

- 2) Crie uma função que receba como parâmetros dois vetores de inteiros: x1 e x2 e as suas respectivas quantidades de elementos: n1 e n2. A função deverá retornar um ponteiro para um terceiro vetor, x3, alocado dinamicamente, contendo a intersecção de x1 e x2 e usar o ponteiro qtd para retornar o tamanho de x3.

Sendo x1 = {1, 3, 5, 6, 7} e x2 = {1, 3, 4, 6, 8}, x3 irá conter {1, 3, 6}.

Assinatura da função:

```
int* interseccao(int *x1, int *x2, int n1, int n2, int *qtd);
```

- 3) Faça um programa que leia a quantidade de elementos n e crie dinamicamente um vetor de n elementos e passe esse vetor para uma função que irá ler os elementos desse vetor. Depois, no programa principal, o vetor preenchido deve ser impresso.

- 4) Analise o código abaixo e determine o valor das variáveis ao final da execução.

```
int i=34, j=0;
int *p;
p = &i;
(*p)++;
j = *p + 33;
printf("%d %d", j, *p);
```

- 5) Analise o código abaixo e determine o valor das variáveis ao final da execução.

```
int i=7, j=5;
int *p;
int **q;
p = &i;
q = &p;
c = **q + j;
```

- 6) Analise o código abaixo e determine o valor das variáveis ao final da execução.

```
int a=5, b=12, c=5;
int *p;
int *q;
p = &a;
q = &b;
```

```
(*p)++;  
c = *p + *q;
```

7) O que faz o código abaixo?

```
#define Y 2  
#define X 2  
  
int main(int argc, char *argv[]){  
    int **A;  
    int i,j;  
  
    A = (int**)malloc(Y * sizeof(int*));  
    for(i = 0; i < Y; i++)  
        A[i] = (int*)malloc(X * sizeof(int));  
    for(i = 0; i < Y; i++)  
        for(j = 0; j < X; j++)  
            A[i][j]=i+j;  
    for(i = 0; i < Y; i++)  
        for(j = 0; j < X; j++)  
            printf("%d \t",A[i][j]);  
    for(i = 0; i < Y; i++)  
        free(A[i]);  
    free(A);  
    return 0;  
}
```