

Exercícios

1. Implemente um programa, para alocar memória para um vetor. O número de posições do vetor será indicado via teclado, assim como o conteúdo das posições. Após a atribuição de valores às posições do vetor, o programa deve imprimir(na tela) as posições e seus valores.
2. Altere o programa para armazenar uma estrutura em cada posição do vetor.

```
typedef struct{  
    char nome[50];  
    char endereco[100];  
    int matricula;  
} estudante;
```

3. Altere o programa do exercício 1, de forma que receba números inteiros do usuário indefinidamente. O programa finaliza quando o usuário entrar com o número 0.
 - a) Aloque, inicialmente, memória para 5 inteiros;
 - b) Caso o usuário entrar com mais inteiros, faça a realocação, alocando espaço para mais 5 inteiros e assim sucessivamente;
4. Crie uma função que receba como parâmetros dois vetores de inteiros, v1 e v2, e as suas respectivas quantidades de elementos, n1 e n2. A função deverá retornar um ponteiro para um terceiro vetor, v3, com capacidade para (n1 + n2) elementos, alocado dinamicamente, contendo a união de v1 e v2.

Por exemplo, se $v1 = \{11, 13, 45, 7\}$ e $v2 = \{24, 4, 16, 81, 10, 12\}$, v3 irá conter $\{11, 13, 45, 7, 24, 4, 16, 81, 10, 12\}$.

O cabeçalho dessa função deverá ser o seguinte:

```
int* uniao(int *v1, int n1, int *v2, int n2);
```

Em seguida, crie a função principal do programa para chamar a função união passando dois vetores informados pelo usuário (ou declarados estaticamente). Em seguida, o programa deve exibir na tela os elementos do vetor resultante. Não esqueça de liberar a memória alocada dinamicamente.