**UDESC – Universidade do Estado de Santa Catarina**
**Curso: Tecnologia em Análise e Desenvolvimento de Sistemas**
**Disciplina: Linguagem de Programação**
**Professora: Luciana Rita Guedes**

# EXERCÍCIOS DE FIXAÇÃO Nº 2.4
## Realocação Dinâmica de Memória

1) Faça um programa que leia uma certa quantidade de inteiros que são armazenados num vetor **v**. A quantidade NÃO será indicada previamente pelo usuário e o programa deve alocar/realocar espaço para **v**, sempre que necessário. À medida que os valores são digitados e armazenador no vetor **v**, o programa também deve armazenar os valores positivos em um vetor **vp** e o valores negativos no vetor **vn**. Da mesma forma como ocorre com o vetor **v**, os dois outros vetores (**vp** e **vn**) devem ser alocados dinamicamente. Os vetores **vp** e **vn** não devem conter zeros. A cada digitação de um novo valor, o programa deve perguntar se o usuário deseja digitar outro número. A digitação encerra quando a resposta do usuário é negativa. Ao final, imprima os três vetores.

2) Considere um vetor **v** contendo **n** valores inteiros. Considere também uma chave de busca **k** (um valor inteiro) que se deseja procurar dentro do vetor **v**. Considere ainda que é possível encontrar de 0 a **n** ocorrências da chave **k** no vetor **v**.
Crie uma função que retorna um vetor (alocado dinamicamente) com os índices em que a chave se encontra (que sempre deve terminar com -1). A função deve realocar memória sempre que encontrar um novo valor no vetor para armazenar o novo índice do valor encontrado.
Sugere-se que o protótipo da função seja como a seguir:

```
int *busca_seq( int v[], int n, int  k );
```

- Exemplo de entrada: v = {3, 6, 7, -1, 3, 12, 9, 8, 3, 17}  chave k = 3
- Saída: vetor resultante = {0, 4, 8, -1}

3) Escreva uma função que realiza a ***união*** entre dois conjuntos de inteiros contidos nos vetores ***v1*** e ***v2***. A função recebe os vetores e suas respectivas capacidades (***n1*** e ***n2***) como parâmetros de entrada e retorna o endereço do vetor alocado (contendo a união entre ***v1*** e ***v2***). Além disso, há um parâmetro passado por referência (ponteiro ***p3***), que serve para "retornar" a capacidade do vetor gerado. Faça o programa principal invocando a função (a estrutura do programa é semelhante ao exemplo dado em aula – *intersecção*). Protótipo da função:

```
int *uniao( int *v1, int n1, int *v2, int n2, int *p3 );
```