

UNIFEI	<p style="text-align: center;">Universidade Federal de Itajubá</p> <p style="text-align: center;">Instituto de Engenharia de Sistemas e Tecnologias da Informação-IESTI</p>
1º Trabalho de ECOPO2 – Estrutura de Dados Profa. Thatyana de Faria Piola Seraphim	

Dado os exercícios a seguir, implemente cada um dos exercícios usando a Linguagem de Programação C. Os exercícios podem ser entregues em arquivos separados, ou em apenas um arquivo com a extensão .c. A atividade deve ser postada direto no SIGAA.

1) (2 pontos) Faça um **programa** que inicializa uma lista estática encadeada com 10 palavras contendo até 30 caracteres. Após a inserção das palavras, o programa deverá percorrer a lista e imprimir na tela as palavras que foram inseridas. O programa deve garantir que não sejam inseridas palavras repetidas. Se necessário usar outras funções auxiliares, todas devem estar implementadas. Todas as funções usadas na função `insereEstEnc` devem ser apresentadas (implementadas). São dados:

```
#define MAX 10
typedef struct {
    char words[30];
    int prox;
}noListaEncadeada;
noListaEncadeada lista[MAX];
int prim=-1;
int dispo=0;
```

2) (1 ponto) Faça a simulação de uma **lista estática encadeada** que armazenada os valores inteiros em um vetor de inteiros de 10 posições. Inicialmente serão inseridos os seguintes valores: 7, 4, 2, 9, 1, 5, 0, 8, 3, 6. Em seguida serão removidos os valores: 4, 7, 0, 9. Finalmente é inserido: 22 e 13. Mostre como fica esse vetor após essa sequência de inserções e remoções. O exercício pode ser feito na folha mesmo.

0	1	2	3	4	5	6	7	8	9

0	1	2	3	4	5	6	7	8	9

0	1	2	3	4	5	6	7	8	9

3) Dado a estrutura de um nó de uma **lista dinâmica simplesmente encadeada** e a variável global chamada **prim** que guarda o início da lista dinâmica simplesmente encadeada:

```
typedef struct no{
    int idade, matrícula;
    char nome[30];
    struct no *prox;
}noLista;
noLista *prim;
```

a) (2 pontos) Implemente uma função chamada **insereDados** que deverá receber como parâmetro uma variável do tipo `noLista`. As informações contidas na variável do tipo `noLista` serão inseridas em uma lista dinâmica simplesmente encadeada. A lista dinâmica deverá ser mantida em ordem alfabética através da matrícula dos alunos. A função deverá retornar verdadeiro, caso os dados sejam inseridos na lista, ou falso, caso contrário.

b) (1,5 pontos) Implemente uma função chamada **desvioPadrao** que calcula e retorna o desvio padrão das idades dos alunos de uma lista dinâmica simplesmente encadeada. O desvio padrão é definido pela raiz quadrada da variância. A variância é definida pela fórmula
$$vr = \frac{1}{n} \sum_{i=1}^n (x_i - md)^2$$
, onde md é a média do conjunto e x_i é cada elemento. Pode ser usada as funções da biblioteca `math.h` `pow()` e a função `sqrt()`.

c) (1,5 pontos) Faça uma função chamada **buscaNome** que recebe como parâmetro o número de matrícula de um aluno que deverá ser encontrado na lista. Essa função deverá retornar se o número existe ou não na lista.

d) (2 pontos) Faça uma função **main** que deverá realizar a leitura pelo teclado das informações de 10 alunos. Deverá ser verificado se a matrícula do aluno já existe na lista. Caso já existe, deverá ser realizado a leitura da matrícula novamente. O programa deverá realizar a chamada para a função **desvioPadrao**, que deverá mostrar na tela o resultado calculado. Não será permitido matrículas repetidas na lista. Não esqueça de liberar toda a memória utilizada pelo programa.