

UNIFEI	Universidade Federal de Itajubá Instituto de Engenharia de Sistemas e Tecnologias da Informação-IESTI
	2º Laboratório de Estruturas de Dados Profª. Thatyana de Faria Piola Seraphim

Esta atividade de laboratório da disciplina de ECOP12 deverá ser postado como tarefa no SIGAA até o prazo definido no SIGAA. O aluno que não postar a atividade, ficará com falta no laboratório.

Essa mesma atividade deverá ser postada também para a disciplina de ECOPO2 para os alunos que cursam a disciplina teórica. O aluno que não postar a atividade ficará com falta na teórica.

Crie um arquivo chamado **ecop12-labo2.c**, que será utilizado para implementar as funções para o laboratório de lista estática encadeada. Esse arquivo que deverá ser postado no SIGAA para ambas as disciplinas ECOPO2 (para quem está matriculado) e ECOP12 (para quem está matriculado).

1) **(0,5 ponto)** Defina o tipo “bool” com os valores “false” e “true”.

```
typedef enum{false, true} bool;
```

2) **(0,5 ponto)** Defina uma estrutura chamada “noAluno”, que será usada para representar os dados de um determinado aluno. Esta estrutura armazenará o nome do tipo char com 20 caracteres, o número de matrícula do tipo inteiro, o IMC do tipo float, e o índice do próximo elemento da lista representado por um número inteiro.

3) **(0,5 ponto)** Crie um vetor contendo 30 elementos do tipo “noAluno”, que armazenará as informações de cada aluno a ser inserido na lista estática encadeada.

4) **(0,5 ponto)** Inicialize as variáveis `prim` com o valor -1, que indica a primeira posição da lista e `dispo` com o valor 0, indicando a posição disponível na lista.

5) **(1 ponto)** Implemente uma função chamada inicializaLista, que não retorna nada e realiza a inicialização de cada índice dos elementos do vetor.

6) **(2 pontos)** Implemente uma função chamada insereAluno, que recebe como parâmetro um elemento do tipo `noAluno`. Essa função deverá retornar verdadeiro se o elemento foi inserido com sucesso na lista e falso caso contrário. Os elementos deverão ser inseridos sempre na primeira posição que estiver disponível na lista estática encadeada. Lembrando que a cada elemento inserido é necessário atualizar o índice do elemento e também a variável `dispo` que indica a próxima posição disponível.

7) **(1,5 pontos)** Implemente uma função chamada imprimeLista que não retorna e imprime a lista com as informações armazenadas. A função deverá imprimir os elementos de forma ordenada pelo índice. Deverá ser impresso na tela as seguintes informações: nome do aluno, matrícula, IMC e o índice. Lembre-se, o primeiro elemento está na posição “`prim`” e deverá ser utilizado o “`prox`” para percorrer a lista de forma ordenada.

8) **(1,5 pontos)** Implemente uma função chamada buscaAluno que deverá retornar a posição do aluno no vetor, ou seja, deverá retornar qual é o índice do vetor. Essa função receberá como parâmetro o número de matrícula do aluno. Pode ser usado o algoritmo de busca sequencial para percorrer a lista.

9) **(2 pontos)** No programa principal faça:

- Faça a inicialização dos índices da lista chamando a função inicializaLista.
- Faça um loop para realizar a leitura dos 30 alunos pelo teclado. A cada leitura verifique se o número de matrícula não está inserido na lista, caso ele não se encontre, então utilize a função insereAluno para inserir um novo aluno na lista.
- Após todas as inserções, faça uma chamada para a função imprimeLista, para imprimir todos os elementos armazenados na lista.
- Faça a leitura pelo teclado de um número de matrícula que deverá ser passado como parâmetro para a função buscaAluno.