

- Data de entrega: Até 12 de março às 17h .

- Procedimento para a entrega:.

1. Submissão: via **RunCodes**.
2. Os nomes dos arquivos e das funções devem ser especificados considerando boas práticas de programação.
3. Funções auxiliares, complementares aquelas definidas, podem ser especificadas e implementadas, se necessário.
4. A solução deve ser devidamente modularizada e separar a especificação da implementação em arquivos *.h* e *.c* sempre que cabível.
5. Os arquivos a serem entregues, incluindo aquele que contém *main()*, devem ser compactados (*.zip*), sendo o arquivo resultante submetido via **RunCodes**.
6. Dentre os arquivos submetidos, deve existir um intitulado *compilcao.txt*, contendo os comandos especificados no *prompt/console* para compilar e executar seu programa.
7. Caracteres como acento, cedilha e afins não devem ser utilizados para especificar nomes de arquivos ou comentários no código.

- Bom trabalho!

## Questão 01

Implemente um **Tipo Abstrado de Dado (TAD) Lista** utilizando **vetor** e utilizando **ponteiros - lista encadeada**. Seu TAD deve contemplar a seguinte especificação.

```
1 #ifndef lista_h
2 #define lista_h
3     typedef struct lista Lista;
4     Lista* lst_cria();
5     void lst_imprime(Lista*);
6     int lst_pertence(Lista*, int);
7     void lst_insere_ordenado(Lista*, int);
8     void lst_retira(Lista*, int);
9     void lst_liberar(Lista*);
10 #endif /* lista_h */
11
```

Dica, na implementação utilizando vetor, para facilitar, defina o tamanho máximo do vetor, como segue:

```
1 | define MAXTAM 1000 /*vetor estatico*/
```

Sua solução deverá conter, ao menos, quatro arquivos: *lista.h*, *lista.c*, *lista\_vetor.c* e *principal.c*. Teste sua implementação utilizando tanto a implementação via vetor quanto usando a lista encadeada.

## Entrada

A entrada é composta por códigos, descritos a seguir, que identificam as operações a serem realizadas e os seus dados complementares, quando cabível. Para facilitar, neste momento, considere que somente uma variável do tipo *lista* será criada e manipulada. **Dica:** antes de executar as operações a seguir, a variável do tipo *Lista\** deve estar devidamente alocada.

- 2 para inserir, de forma ordenada, um *elemento* na lista; seguido pelo *elemento*.
- 3 para retirar um elemento da lista; seguido pelo *item* a ser removido.
- 4 para verificar se *um* elemento pertence à lista, seguido pelo *elemento* em questão.

- 5 para imprimir todos os elementos da lista.
- 6 para liberar a memória alocada para a lista.
- 1 para encerrar o programa.

## Saída

A saídas serão em conformidade com as operações que manipulam um diretório especificadas na entrada.

## Exemplo de Entrada e Saída

A seguir, um exemplo de cenário de execução do seu programa:

Entrada	Saída
2	1 4 5 6
5	5 6
2	
4	
2	
6	
2	
1	
5	
3	
4	
3	
1	
5	
1	
-1	

## Questão 02

Num arquivo *.pdf*, defina, utilizando notação assintótica a complexidade e tempo de todas as funções implementadas tanto usando vetor quanto usando encadeamento, tal como segue. Faça um tabela comparativa para evidenciar a complexidade considerando essas duas formas de implementação do TAD Lista. Na tabela, aponte também a complexidade de tempo para o melhor caso e para o pior caso de cada função.