

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
**Faculdade da Computação**  
**Trabalho Individual de AED1 – Valor 12 Pontos**  
**Profa. Gina Maira B. de Oliveira**

- Deve ser entregue para a profa. em mídia (DVD) até **14/05/2018** na sala de aula.
  - A apresentação individual dos códigos será agendada posteriormente.
  - Os códigos deverão ser implementados somente em Linguagem C, sendo necessária a utilização das estruturas de dados conforme discutidas em sala.
- 1) Implementar o TAD **lista não ordenada** usando alocação **estática/seqüencial**. A TAD deve conter todas as operações vistas em sala e laboratório: **cria\_lista**, **lista\_vazia**, **lista\_cheia**, **insere\_elem**, **remove\_elem**, **imprime\_lista** e **remove\_todos**, além de incorporar as operações a seguir:
- **Inserir no início:** inserir o elemento no início da lista (obs: a **insere\_elem** vista em sala insere no final da lista, para essa forma de implementação).
  - **Remover ímpares:** remove todos os elementos ímpares da lista.
  - **Menor:** retorna o menor elemento da lista
  - **Tamanho:** retorna o número de elementos da lista
  - **Concatenar:** recebe duas listas (L1 e L2) e retorna uma nova lista L3 com os elementos de L1 seguidos dos elementos de L2.
- 2) Implementar o TAD **lista ordenada** usando alocação **estática/seqüencial**. A TAD deve conter todas as operações vistas em sala e laboratório: **cria\_lista**, **lista\_vazia**, **lista\_cheia**, **insere\_ord**, **remove\_ord**, **imprime\_lista** e **intercala**, além de incorporar as operações a seguir:
- **Remover pares:** remove todos os elementos pares da lista.
  - **Maior:** retorna o maior elemento da lista
  - **Tamanho:** retorna o número de elementos da lista
  - **Iguais:** recebe duas listas ordenadas e verifica se elas são iguais
- 3) Implementar o TAD **lista não ordenada** usando alocação **dinâmica/encadeada simples (SEM cabeçalho)**. A TAD deve conter todas as operações vistas em sala e laboratório: **cria\_lista**, **lista\_vazia**, **lista\_cheia**, **insere\_elem**, **remove\_elem**, **imprime\_lista** e **remove\_todos**, além de incorporar as operações a seguir:
- **Inserir no final:** inserir o elemento no final da lista (obs: a **insere\_elem** vista em sala insere no início da lista, para essa forma de implementação).
  - **Remover pares:** remove todos os elementos ímpares da lista.
  - **Maior:** retorna o menor elemento da lista
  - **Tamanho:** retorna o número de elementos da lista
  - **Concatenar:** recebe duas listas (L1 e L2) e retorna uma nova lista L3 com os elementos de L1 seguidos dos elementos de L2.
- 4) Implementar o TAD **lista ordenada** usando alocação **dinâmica/encadeada simples (SEM cabeçalho)**. A TAD deve conter todas as operações vistas em sala e

laboratório: **cria\_lista**, **lista\_vazia**, **lista\_cheia**, **insere\_ord**, **remove\_ord**, **imprime\_lista** e **intercala**, além de incorporar as operações a seguir:

- **Remover ímpares**: remove todos os elementos pares da lista.
- **Menor**: retorna o maior elemento da lista
- **Tamanho**: retorna o número de elementos da lista
- **Iguais**: recebe duas listas ordenadas e verifica se elas são iguais

5) Implementar o TAD **lista ordenada** usando alocação **dinâmica/encadeada COM cabeçalho**. A TAD deve conter todas as operações vistas em sala e laboratório: **cria\_lista**, **lista\_vazia**, **lista\_cheia**, **insere\_ord**, **remove\_ord**, **imprime\_lista** e **tamanho**, além de incorporar as operações a seguir:

- **Remover ímpares**: remove todos os elementos pares da lista.
- **Menor**: retorna o maior elemento da lista
- **Iguais**: recebe duas listas ordenadas e verifica se elas são iguais.
- **Intercalar**: recebe duas listas ordenadas (L1 e L2) e retorna uma nova lista L3 com os elementos de L1 e L2 intercalados conforme a ordenação.

6) Implementar o TAD **lista não ordenada** usando alocação **dinâmica com encadeamento CÍCLICO**. A TAD deve conter todas as operações vistas em sala e laboratório: **cria\_lista**, **lista\_vazia**, **lista\_cheia**, **insere\_elem**, **remove\_elem**, **imprime\_lista** e **tamanho**, além de incorporar as operações a seguir:

- **Inserir no início**: inserir o elemento no início da lista
- **Inserir na posição**: insira o elemento em uma posição definida na chamada da função. A operação deve verificar se a posição desejada é válida.
- **Remove elemento da posição**: remover o elemento que se encontra na posição definida na chamada da função. Se a posição não existir na lista, a operação deve indicar falha.
- **Maior**: retorna o maior elemento da lista.

7) Implementar o TAD **lista não ordenada** usando alocação **dinâmica com encadeamento duplo**. A TAD deve conter todas as operações vistas em sala e laboratório: **cria\_lista**, **lista\_vazia**, **lista\_cheia**, **insere\_elem**, **remove\_elem**, **imprime\_lista** e **tamanho**, além de incorporar as operações a seguir:

- **Remover todos**: remove todas as ocorrências de um elemento da lista
- **Remover maior**: remove o maior elemento encontrado na lista.
- **Múltiplos de 3**: retornar uma lista L2 formada pelos elementos da lista de entrada L, que são múltiplos de 3.