

Data:

Aluno: \_\_\_\_\_ Nota: \_\_\_\_\_

## 1ª Prova

### Instruções

- Leia atentamente todas as questões;
- Todas as respostas devem ser implementadas e submetidas via run.codes;
- Alunos envolvidos em plágio terão notas zeradas (tanto quem forneceu o código quanto quem copiou) - A ferramenta run.codes detecta plágios!

- 1 Considere uma implementação de lista encadeada com cabeça e sem cauda que armazena valores reais. Implemente a função **conc**, que concatena duas listas L1 e L2 passadas como parâmetro, intercalando seus elementos. Ao final, imprima o resultado na tela.

Exemplo:

L1: [1.2 3.5 9.78]

L2: [5 2.9]

Saída: [1.2 5 3.5 2.9 9.78]

- 2 Implemente uma lista duplamente encadeada com elementos que armazenem valores inteiros. Além disso, implemente uma função que recebe um vetor de inteiros e os insere na lista de maneira ordenada. Observação: seu algoritmo não pode recomeçar a partir da cabeça a busca pela posição correta após cada inserção na lista, exceto na primeira inserção. Ex.:

Lista atual: [0 1 3 5 6]

Inserir vetor: [4 2 7]

Para inserir o elemento 4, a busca deve começar a partir do elemento 0. Ao inserir o elemento 4 após o 3 [0 1 3 4 5 6], o algoritmo não deve voltar para o primeiro elemento 0 para buscar a posição do valor 2 que será inserido na sequência. O algoritmo deve fazer essa inserção usando o encadeamento para trás.

- 3 A turma de ED estava “batendo um baba” na praia quando um novo estudante chegou e disse “15 minutos, 2 gols!”. Após 15 minutos, o time que perdeu de 1x0 precisou escolher alguém para sair. Para decidir qual estudante deveria dar o lugar, todos do time perdedor fizeram um círculo e cada um disse um número aleatório de 0-5. Somaram, então, os números dos estudantes e, em seguida, escolheram alguém do círculo para começar a contagem. A partir desse estudante, no sentido horário, começaram a contar até chegar na soma total. A pessoa, na qual a contagem finalizou, deveria dar o lugar ao estudante que chegou.

Exemplo de entrada:

[5 0 2 0 4 3] //números aleatórios escolhidos pelos estudantes time perdedor  
3 //posição que deve iniciar a contagem. Posições: [0 1 2 3 4 5]

Execução

[ 0 1 2 3 4 5]  
[        1 2 3]  
[ 4 5 6 7 8 9]  
[10 11 12 13 14 ]

Saída:

4 // a pessoa que estava na posição 4 deve dar lugar ao novo estudante

- 4 Implemente uma lista duplamente encadeada circular e duas funções que insere (operação 1) e remove (operação 0) elementos em uma determinada posição específica.

Exemplo:

// [operação] [valor, se operação 1; posição, se operação 0] [posição, se operação 1, vazio se operação 0]

1 1 0

1 2 1

1 5 2

1 3 3

0 2

1 7 2

Lista antes da remoção:

[ 1 ⇌ 2 ⇌ 5 ⇌ 3 ]

↑ \_\_\_\_\_|

Lista após a remoção:

[ 1 ⇌ 2 ⇌ 3 ]

↑ \_\_\_\_\_|

Lista após a nova inserção:

[ 1 ⇌ 2 ⇌ 7 ⇌ 3 ]

↑ \_\_\_\_\_|

- 5 Implemente uma pilha e suas funções de empilhar e desempilhar. Em seguida, implemente uma função que retorne uma a pilha cujo topo contém o elemento que estava na base. Utilize uma pilha auxiliar para resolver o problema. Exemplo:

Entrada

[ 4]

[17]

[ 9]

[23]

Saida

[23]

[ 4]

[17]

[ 9]