Universidade Federal Fluminense
Luís Felipe Ignácio Cunha

Maratona 1 - Programação de Computadores

As implementações devem, rigorosamente, ser feitas com o conteúdo que foi ensinado até a aula 5. Não deve utilizar comandos prontos de python, tais como sort

Parte A: Básicos (aquecimento)

1. **Criação e tamanho.** Leia 5 números (um por linha) para uma lista e imprima a lista e seu tamanho.
2. **Acesso e atualização.** Dada uma lista de 5 números lidos, troque o primeiro elemento com o último e imprima a lista resultante.
3. **Soma e média.** Leia n, depois n números; calcule a soma e a média percorrendo a lista com um laço.
4. **Mínimo e máximo.** Leia um número por linha (um número arbitrário de números, parando a leitura somente quando nada for digitado); encontre o menor e o maior elemento.
5. **Contagem de pares/ímpares.** Para a lista lida, conte quantos elementos são pares e quantos são ímpares.
6. **Índices das ocorrências.** Leia uma lista e um valor x. Imprima todos os índices onde x aparece (ou "não encontrado").
7. **Remover ocorrências.** Remova *todas* as ocorrências de um valor dado em uma lista. Se o valor não existir, então deve retornar "valor não encontrado".

Parte B:

8. **Lista a partir de uma linha.** Leia uma linha com números separados por espaço, use `split()` e converta para `int`. Imprima a lista.
9. **Ordenação.** Ordene a lista e imprima-a.
10. **Segundo menor e segundo maior (com repetidos).** Após ordenar, encontre o *segundo* menor e o *segundo* maior. Considere que elementos repetidos contam para a posição (ex.: em $[1, 1, 2, 3, 3]$, segundo menor $= 1$; segundo maior $= 3$).
11. **Sem duplicatas (preservando ordem).** Construa uma *nova* lista sem repetir elementos, mantendo a ordem de aparição da lista original.

12. **União e interseção.** Dadas duas listas A e B (ambas lidas com `split()`):
União: elementos de A seguidos dos de B que ainda não estejam na união.
Interseção: elementos que aparecem em **ambas**, sem repetir, na ordem de A.

13. **Rotação à direita por k.** Gire a lista k posições à direita. Pode usar fatiamento (`lista[-k:] + lista[:-k]`) ou laços. Ex.:

    1 2 3 4 5 6
    3
    $[4, 5, 6, 1, 2, 3]$

14. **Inversão *in place*** Inverta a lista trocando elementos das "pontas para o meio" com dois índices (i e j). Ex.: 1 2 3 4 $\rightarrow [4, 3, 2, 1]$.

Parte C:

15. **Crescente estrita.** Verifique se a lista está estritamente crescente (cada próximo $>$ anterior). Imprima `True` ou `False`.

16. **Comprimir consecutivos (RLE simples).** Para uma lista de inteiros, gere pares [valor, contagem] de *runs* consecutivos. Ex.: $[1, 1, 1, 2, 2, 5] \rightarrow [[1, 3], [2, 2], [5, 1]]$.

17. **Mesclar alternando.** Dadas A e B, crie C alternando elementos ($A[0], B[0], A[1], B[1], \ldots$). Se uma acabar, anexe o resto da outra.

18. **Histograma de notas (0 a 10).** Leia inteiros de 0 a 10 e conte quantas vezes cada nota aparece, usando **uma** lista de contagem com 11 posições.

19. **Matriz 3×3 (lista de listas).** Leia 9 números em uma única linha e monte uma matriz 3×3 (linhas de 3). Imprima: soma de cada linha, de cada coluna e das duas diagonais.

20. **Palíndromo em lista.** Leia uma palavra/frase *sem acentos nem pontuação*. Construa uma lista só com as letras (ignorar espaços) e verifique se é palíndromo comparando elementos espelhados.

Desafios

- **D1) K-maiores sem ordenar tudo.** Dada a lista e um k, construa *outra* lista com os k maiores **sem** ordenar a original inteira. Dica: mantenha uma lista auxiliar ordenada de tamanho $\leq k$.
- **D2) Particionar por pivô.** Dado um pivô, reordene a lista colocando primeiro os $<$ pivô, depois os $=$ pivô e, por fim, os $>$ pivô (pode usar três listas auxiliares).
- **D3) Sublista crescente contígua mais longa.** Encontre o comprimento da maior sequência *contígua* estritamente crescente (em termos de índices consecutivos).
- **D4) Diferença simétrica.** Elementos que estão em **exatamente uma** das listas A ou B, preservando a ordem de aparição (primeiro os que aparecem em A, depois os de B que ainda não estiverem no resultado).
- **D5) Mini "Forca".** Use uma lista para as letras da palavra e outra para o estado revelado (`_`), atualizando a cada palpite. Limite de tentativas com `while`.