



817

AP2
2024.2

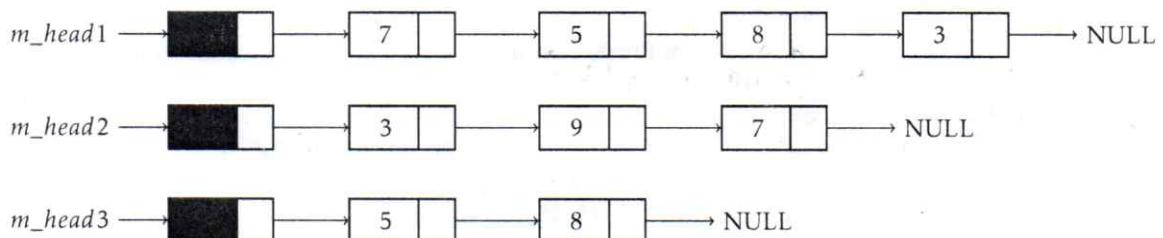
Nome: SAMMUEL KERMYSON DIAS SILVA Matrícula: 565544

- [2.5 pontos] 1. Para essa questão, suponha uma **lista simplesmente encadeada com nó sentinelas**. Escreva uma função em C++ que recebe como entrada **dois ponteiros para os nós sentinelas de duas listas simplesmente encadeadas** e retorna um ponteiro para uma nova lista contendo todos os elementos da primeira lista que não estejam na segunda lista. As duas listas usadas na criação da terceira NÃO devem ser modificadas. Pode supor que numa lista não há elementos repetidos.

O protótipo da função é dado por:

```
Node* diferença(Node *m_head1, Node *m_head2);
```

Um exemplo da operação que deve ser realizada entre as duas listas é ilustrado abaixo. Veja que os únicos elementos que estão na primeira lista e que não estão na segunda lista são os números 5 e 8. Os números na terceira lista que será produzida podem estar em qualquer ordem.



- [2.5 pontos] 2. Dada uma fila de caracteres chamada `fila1` formada por uma sequência alternada de letras e dígitos, construa uma função em C++ que receba `fila1` como parâmetro e retorne uma outra fila na qual as letras são mantidas na sequência original e os dígitos são colocados na ordem inversa. Exemplos:

A 1 E 5 T 7 W 8 G → A E T W G 8 7 5 1
 7 M 3 C 9 H 4 Q 6 → M C H Q 6 4 9 3 7

Como é mostrado nos exemplos, na nova fila temos que **as letras devem ser mostradas primeiro, seguidas dos dígitos**. Requisitos da questão:

- Usar fila e pilha para resolver a questão
- Use a fila e a pilha do C++ (Não é para implementar as estruturas de dados, é para usar. **Se você implementar alguma ED a questão será anulada**)
- Suponha que exista uma função chamada `is_digit(char c)` que retorna `true` caso o caractere `c` seja um dígito, e retorna `false` caso contrário. Você pode usar essa função no seu programa.

Observação 1: A sua função deve se chamar `novaFila` e o protótipo da sua função deve ser assim:

```
Queue novaFila(Queue fila1);
```

Observação 2: Não preciso me entregar função main, eu quero só o código da função acima.

Observação: Para a questão 3 a seguir, considere o nó de uma árvore binária definido como o struct a seguir:

```

1 struct Node {
2     int value;
3     Node *left;
4     Node *right;
5 }
```

- [2 pontos] ③ Escreva uma função recursiva que soma todos os números ímpares contidos em uma árvore binária qualquer e retorna o valor da soma. Sua função deve obedecer ao protótipo:

1.9

```
int somaImpares(Node *node);
```

- [1.5 pontos] ④ (ENADE 2023) Considere que os números 27, 34, 40, 18, 23, 5, 25, 36, 10, 7, -2 foram todos inseridos nesta ordem em uma **Árvore Binária de Busca**, inicialmente vazia, elemento a elemento. Assinale a alternativa correta.

1.5

1. A árvore resultante terá 5 níveis de altura, com ~~6~~ elementos à esquerda da raiz principal (inicial) e ~~5~~ elementos à direita.
2. O percurso da árvore em Pré-ordem irá processar os elementos na seguinte ordem (do primeiro ao último): ~~-2, 7, 10, 5, 25, 23, 18, 36, 40, 34, 27~~.
3. O percurso da árvore em Em-ordem (simétrico) irá processar os elementos na seguinte ordem (do primeiro ao último): ~~-2, 5, 7, 10, 18, 23, 25, 27, 34, 36, 40~~. ✓
4. O percurso da árvore em Pós-ordem irá processar os elementos na seguinte ordem (do primeiro ao último): ~~27, 18, 5, -2, 10, 7, 23, 25, 34, 40, 36~~.
5. O número máximo de elementos que essa árvore poderá ter com 10 níveis será de 1024 elementos.

- [1.5 pontos] ⑤ (POSCOMP 2019) Sobre listas, analise as assertivas abaixo:

1.5

- I. Objetos podem ser inseridos em uma pilha a qualquer momento, mas apenas o que foi inserido mais recentemente (isto é, o último) pode ser removido a qualquer momento. ✓
- II. Em uma fila, os elementos podem ser inseridos a qualquer momento, mas apenas o elemento que está a mais tempo na fila pode ser removido. ✓
- III. Em uma fila, os elementos são inseridos e removidos de acordo com o princípio “o último que entra é o primeiro que sai”. ✗

Quais estão corretas?

- (A) Apenas I.
- (B) Apenas II.
- (C) Apenas III.
- ✓ (D) Apenas I e II.
- (E) I, II e III.