

Aluno(a):

1. Considere a seguinte sequência de operações:

`push(7), push(2), pop(), push(4), pop(), push(5),
push(8), pop(), pop(), push(3), push(1), pop()`.

- A soma dos elementos remanescentes em uma *fila*, inicialmente vazia, após essa sequência de operações será _____.
- A soma dos elementos remanescentes em uma *pilha*, inicialmente vazia, após essa sequência de operações será _____.

2. Seja $push(n)$ a operação que insere um número n em uma *pilha*. Considere a execução desta operação, com n assumindo os valores 1, 2, 3, 4 e 5, nesta ordem. Em meio às inserções, pode haver remoções (*pop*). Indique a ordem das operações $push(n)$ e $pop()$ para que a sequência de números retirados da pilha com a operação *pop* seja:

(a) 3,4,2,5,1: _____

(b) 1,4,5,3,2: _____

3. Considere as seguintes estruturas utilizadas para implementar uma fila:

```
typedef struct{          typedef struct{
    int dado;             item *inicio;
    struct item *next;    item *fim;
}item;                   }fila;
```

Complete o código abaixo para implementar a função *pop* em uma fila, assumindo que o parâmetro **f* é um ponteiro para o primeiro elemento da estrutura.

```
1 int pop(fila *f){
2     int dado = f->inicio->dado;
3     _____;
4     if(f->inicio==NULL)
5         _____;
6     return dado;
7 }
```

Linha 3: _____

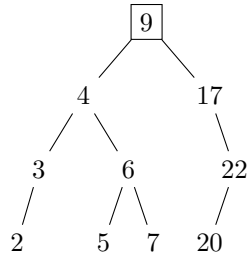
Linha 5: _____

4. Assinale a alternativa correta a respeito de tabelas hash:

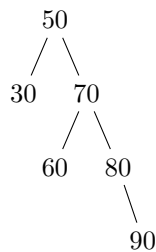
- (a) Em tabelas livres de colisões, tem-se a garantia, independente da função hash utilizada, de que os elementos estarão dispostos na ordem em que foram inseridos. O primeiro elemento inserido ocupará a primeira posição da tabela, o segundo elemento ocupará a segunda posição e assim por diante.
- (b) Seja uma tabela de tamanho m populada com n elementos ($n < m$). Se colisões forem resolvidas por sondagem linear, então a inserção de um elemento requer, no mínimo, n comparações até encontrar a posição adequada.
- (c) Em tabelas livres de colisões, a busca por um elemento tem, no pior caso, complexidade igual à complexidade da busca em árvores binárias de pesquisa.
- (d) Na resolução de colisões por encadeamento, a busca por um elemento, no pior caso, terá complexidade semelhante à busca em listas encadeadas

5. Considere a inserção das seguintes chaves na dada ordem em uma tabela hash de tamanho 7: {**282, 44, 251, 22, 129, 266, 300**} usando o método da divisão com a função hash $h(x) = x \bmod 7$. Mostre as tabelas resultantes resolvendo as colisões por encadeamento e por sondagem linear. Responda no seguinte formato: 0:[...], 1:[...], ..., n:[...]

Para as questões 6 a 7, considere a árvore da figura abaixo:



6. Escreva abaixo a sequência de números que será impressa caso a árvore seja percorrida
 - a) Em pré-ordem: _____
 - b) Em ordem: _____
 - c) Em pós-ordem: _____
7. Caso o nó armazena o número 4 seja excluído e a árvore permaneça sendo uma árvore binária de pesquisa, a *soma* do valor de suas folhas será _____.
8. Transforme a árvore abaixo em uma árvore AVL.



9. Desenhe a árvore binária de pesquisa (não balanceada) gerada pela inserção da seguinte sequência de números: 50, 30, 80, 90, 60, 95, 55, 85, 65, 87, 70, 82, 61, 89.
10. Desenhe a árvore AVL gerada pela inserção da seguinte sequência de números: 50, 30, 80, 90, 60, 95, 55, 85, 65, 87, 70, 82, 61, 89.