

RA: _____

Assinatura: _____

1. Esta prova é individual e sem consulta.
2. Esta prova pode ser feita a lápis ou a tinta.

Questão 1 [10] Explique como simular o funcionamento de uma pilha usando uma única fila e uma quantidade constante de memória extra. As operações disponíveis na fila são enfileirar e desenfileirar.

☐ Não sei esta questão. Assinatura: _____ [10% do valor da questão]

Uma fila se tornaria uma pilha no momento em que o último elemento a entrar na fila fosse o primeiro a sair. Ao entanto as operações de enfileirar posiciona o elemento ao final da fila e desenfileirar retira o "primeiro" elemento (o elemento mais à frente).

Deve-se acrescentar uma operação de inversão da fila, de modo que o último elemento a entrar seja o primeiro a sair.

Seja Q uma fila. Para acrescentar x à fila:

- enfileira x ;
- Enquanto x não for o primeiro da fila:
 - desenfileira y ;
 - enfileira y ;

Desta forma é produzida uma inversão na fila e a opção de desenfileirar é equivalente ao POP da pilha.

O uso de memória é constante, visto que se usa apenas o auxiliar y para inverter a fila.

Questão 2 [10] Uma tabela de hashing T com sondagem por hashing dupla e função definida pelo método da divisão terá tamanho $m = 13$.

- Escreva a definição completa da função de hashing para a tabela T .
- Complete a sequência abaixo com os dígitos do seu RA agrupados dois a dois a partir do primeiro.

18 55 21 49 62 50 07 47

Por exemplo, se o RA for 290338 então a sequência será

29 55 21 49 03 50 38 47

Insira dígitos na sequência, em ordem, em T . Apresente a tabela final.

☐ Não sei esta questão. Assinatura: _____ [10% do valor da questão]

a) $h_1(k) = (k) \% (m)$ b) $h_2(k) = (1 + (k) \% (m-2))$

$h(k, i) = (h_1(k) + i * h_2(k)) \% m$

definimos $m = 13$.

i é incremento

			55		18	68	07	21		49	50	47	
m	0	1	2	3	4	5	6	7	8	9	10	11	12

Questão 3 [10] Complete a sequência abaixo com os dígitos do seu RA agrupados dois a dois a partir do primeiro.

18 26 25 08 62 32 55 48 36 80 07 96 21 93

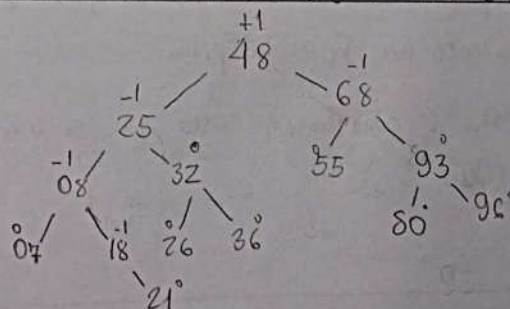
Por exemplo, se o RA for 290338 então a sequência será

29 26 25 08 03 32 55 48 36 80 38 96 21 93

Insira os dígitos na sequência, em ordem, em uma árvore AVL. Ao terminar, escreva as chaves na ordem dada por um percurso em largura na árvore.

☐ Não sei esta questão. Assinatura: _____ [10% do valor da questão]

07-08-18-21-25-26-32-36-48-55-68-80-93-96



Questão 4 [10] Uma árvore binária T é chamada equilibrada se:

- T é vazia ou
- T consiste de uma raiz r tendo T_e e T_d como subárvores esquerda e direita respectivamente e o número de nós em T_e difere do número de nós em T_d em no máximo uma unidade.

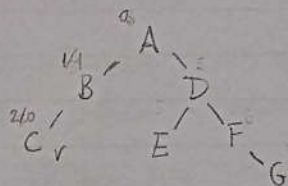
c. T_e e T_d tem que ser equilibradas

Escreva, na página seguinte, uma função em C que recebe como primeiro parâmetro um apontador para a raiz de uma árvore binária e como segundo parâmetro o endereço de um int onde a função deve armazenar o número de nós da árvore. A função deve retornar 1 se a árvore for equilibrada e 0 caso contrário. O nó da árvore e o cabeçalho da função estão definidos abaixo.

```
typedef struct node {
    int info;
    struct node *left, *right;
} node;

int equilibrada(node* t, int* n);
```

☐ Não sei esta questão. Assinatura: _____ [10% do valor da questão]



Questão 4

// ~~nao~~ na main n inicializado com zero na main

```

int equilibrada (node* t, int* n) {
    if (t == NULL) return 1; // (sub)arvore equilibrada
    int ne, nd, nantes, eq;
    nantes = *n; *n = *n + 1;
    eq = equilibrada (t->left, n);
    if (eq == 0) return 0; // arvore desequilibrada
    ne = *n - nantes - 1; // não é esquerda
    nantes = *n;
    eq = equilibrada (t->right, n);
    if (eq == 0) return 0; // arvore desequilibrada
    nd = *n - nantes; // não é direita
    if (ne - nd > 1 || ne - nd < -1)
        return 0;
    else
        return 1;
}

```

3

10,