



Nome: \_\_\_\_\_ Matrícula: \_\_\_\_\_

**ATENÇÃO:** Escolha apenas 4 questões e resolva. Cada questão vale até 2,5 pontos. Não corrigirei mais do que quatro questões.

**Observação:** Para todas as questões de árvore binária desta prova, considere o nó de uma árvore binária definido como a seguir:

```
1 struct no_arv {  
2     int chave;  
3     struct no_arv *esq;  
4     struct no_arv *dir;  
5 };  
6  
7 typedef no_arv NoArv;
```

1. Escreva uma **função recursiva** em C que retorne a quantidade de nós de uma árvore binária que possuem exatamente um filho. Essa função deve obedecer ao protótipo:

```
int um_filho(NoArv *node);
```

2. Escreva uma **função recursiva** em C que apaga todas as folhas de uma árvore binária que tenham a chave igual a um valor dado. Sua função deve obedecer ao protótipo:

```
NoArv* apaga_folhas_com_valor(NoArv *node, int valor);
```

3. Duas árvores binárias são **idênticas** quando possuem os mesmos dados e a disposição dos dados também é a mesma. Esse conceito é ilustrado na Figura 1. Escreva uma função **recursiva** em C que recebe duas árvores binárias como entrada e devolve **true** se elas forem idênticas; e **false**, caso contrário. O protótipo da função é dado por:

```
bool arvores_identicas(NoArv *a1, NoArv *a2);
```

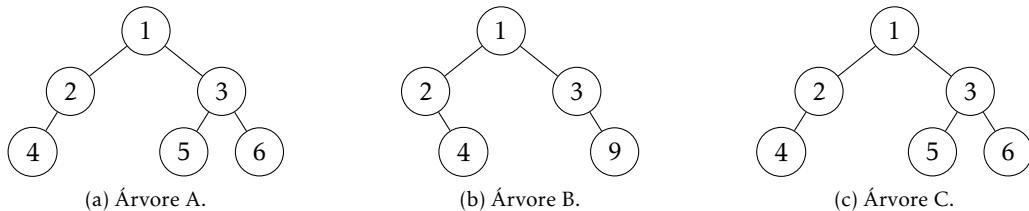


Figura 1: As árvores A e C são idênticas. Porém as árvores A e B não são idênticas.

4. 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. A árvore resultante terá 5 níveis de altura, com 6 elementos à esquerda da raiz principal (inicial) e 4 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.

5. Sobre as **árvores binárias de busca**, é correto afirmar que:

  - (i) Seja  $x$  um nó em uma árvore de busca binária. Se  $y$  é um nó na subárvore esquerda de  $x$ , então  $y.chave \geq x.chave$ . Se  $y$  é um nó na subárvore direita de  $x$ , então  $x.chave \leq y.chave$ .
  - (ii) A propriedade de árvore de busca que permite imprimir todas as chaves em sequência ordenada por meio de um simples algoritmo recursivo é denominada percurso de árvore em pré-ordem.
  - (iii) Para excluir um nó  $z$  de uma árvore de busca binária  $T$ , se  $z$  tem apenas um filho, então simplesmente o removemos modificando seu pai de modo a substituir  $z$  por nulo como seu filho.
  - (iv) Para excluir um nó  $z$  de uma árvore de busca binária  $T$ , se  $z$  tem dois filhos, então elevamos o primeiro filho para que ocupe a posição de  $z$  na árvore modificando o pai de  $z$  de modo a substituir  $z$  pelo filho de  $z$ .
  - (v) Para encontrar um nó em uma árvore de busca binária cuja chave é um mínimo, deve-se seguir os ponteiros de filhos da esquerda desde a raiz até encontrar um valor nulo.