# ***Đề gồm 15 câu trắc nghiệm, 15 câu tự luận viết hàm.***

1. **TRẮC NGHIỆM (Bắt buộc)**

# ***Câu 1: Câu 28***

|  |
| --- |
| In cây sau theo thứ tự preorder:    A. A B D G C E H I F  B. A B D G C H E I F  C. A B G D C F E H I  D. A B G D C E H I F |

# ***Câu 2: Câu 29***

|  |
| --- |
| In cây sau theo thứ tự postorder:    A. I E F J C G D K L H B A  B. I E J F C G H K L D B A  C. I E J F C G K L H D B A  D. I E J F C G D K L H B A |

# ***Câu 3: Câu 30***

|  |
| --- |
| In cây sau theo thứ tự inorder:    A. G D B A C H E I F  B. D G B A H E I C F  C. D G B A H I E C F  D. G D B A H E I C F |

# ***Câu 4: Câu 31***

|  |
| --- |
| Cho T là cây nhị phân đầy đủ gồm các node: a, b, c, d, e, f, g.  Duyệt cây T theo thứ tự preorder ta được: b, g, d, a, c, f, e.  Duyệt cây T theo thứ tự inorder ta được: d, g, c, a, f, b, e.  Hỏi các node ở mức cao nhất trong cây T gồm những node nào?  A. c, a  B. c, f  C. d, a  D. d, f |

# ***Câu 5: Câu 32***

|  |  |  |  |
| --- | --- | --- | --- |
| Cấp độ | 2 Analyzing | Thời gian | 2 |
| CĐR: | | | |
| Hãy vẽ cây nhị phân T biết rằng thứ tự duyệt cây T theo inorder và preorder như sau:  Duyệt theo inorder: D, G, B, H, E, A, F, I, C  Duyệt theo preorder: A, B, D, G, E, H, C, F, I   |  |  | | --- | --- | | A. | B. | | C. | D. | | | | |

# ***Câu 6:***

|  |
| --- |
| Khi sao chép dữ liệu từ cây nhị phân tìm kiếm sang một danh sách có thứ tự, cách tốt nhất là duyệt cây T theo thứ tự:  A. inorder  B. postorder  C. levelorder  D. preorder |

# ***Câu 7:***

|  |
| --- |
| Kiểu dữ liệu phù hợp để biểu diễn quan hệ phân cấp giữa các phần tử là kiểu:  A. Queue  B. Tree  C. List  D. Graph |

# ***Câu 8:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Định nghĩa kiểu dữ liệu Node trong cây nhị phân tìm kiếm nào sau đây là đúng?   |  |  | | --- | --- | | Đoạn 1.  struct Node {  int key;  NODE \*pLeft;  NODE \*pRight;  } NODE; | Đoạn 2.  typedef struct Node {  int key;  Node \*pLeft;  Node \*pRight;  } NODE; | | Đoạn 3.  struct Node {  int key;  Node\* pLeft;  Node\* pRight;  }; | Đoạn 4.  typedef struct Node {  int key;  NODE\* pLeft;  NODE\* pRight;  } NODE; |   A. Đoạn 1 và 3  B. Đoạn 2 và 3  C. Đoạn 2, 3 và 4  D. Cả 4 đoạn trên |

# ***Câu 9:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Khai báo biến T lưu trữ cây nhị phân tìm kiếm nào sau đây là sai?   |  |  | | --- | --- | | Đoạn 1.  typedef Node\* TREE;  TREE T; | Đoạn 2.  Node\* T; | | Đoạn 3.  Node\* TREE;  TREE T; | Đoạn 4.  typedef Node\* T; |   A. Đoạn 1 và 2  B. Đoạn 2 và 4  C. Đoạn 3 và 4  D. Đoạn 2, 3 và 4 |

# ***Câu 10:***

|  |  |  |  |
| --- | --- | --- | --- |
| Cấp độ | 2 Applying | Thời gian | 1 |
| CĐR: | | | |
| Tạo cây BST từ dãy số theo thứ tự đầu vào như sau: 40, 30, 35, 100, 80, 85, 60, 50, 10, 15, 20, 6, 3. Hỏi có bao nhiêu node lá trong cây?  A. 4  B. 5  C. 6  D. 7 | | | |

# ***Câu 11:***

|  |  |  |  |
| --- | --- | --- | --- |
| Cấp độ | 2 Applying | Thời gian | 1 |
| CĐR: | | | |
| Tạo cây BST từ dãy số theo thứ tự như sau: 40, 30, 35, 100, 80, 85, 60, 50, 10, 15, 20, 6, 3. Hỏi có bao nhiêu node có bậc 2 trong cây?  A. 3  B. 4  C. 5  D. 6 | | | |

# ***Câu 12:***

|  |
| --- |
| Cho cây nhị phân tìm kiếm như sau:    Hỏi node thay thế khi thực hiện thao tác xóa node 30 là node nào?  A. 25 hoặc 27  B. 25 hoặc 40  C. 27 hoặc 40  D. 29 hoặc 40 |

# ***Câu 13:***

|  |
| --- |
| Cho cây nhị phân tìm kiếm như sau:    Con trái của node 30 sau khi thực hiện xóa node 25 là:  A. 26  B. 27  C. 29  D. Không có con nào |

# ***Câu 14:***

|  |
| --- |
| Xóa node 40 trên cây nhị phân bên dưới (Lưu ý: Nếu nút cần xóa có 2 cây con thì thay thế bằng node con bên trái cùng của cây con phải node cần xóa). Sau khi xóa thì số lượng node có 1 cây con trên cây sẽ là:    A. 2  B. 3  C. 4  D*.* 5 |

# ***Câu 15:***

|  |
| --- |
| Lần lượt xóa các node 30, 35, 40 trên cây nhị phân bên dưới (Lưu ý: Nếu nút cần xóa có 2 cây con thì thay thế bằng node con bên trái cùng của cây con phải node cần xóa). Sau khi xóa, thực hiện duyệt cây theo thứ tự preorder sẽ được dãy số:    A. 50, 10, 6, 3, 15, 20, 60, 80, 85, 100  B. 50, 10, 3, 6, 15, 20, 80, 60, 85, 100  C. 50, 10, 6, 3, 15, 20, 80, 60, 100, 85  D. 3, 6 20, 15, 10, 60, 85, 100, 80, 50 |