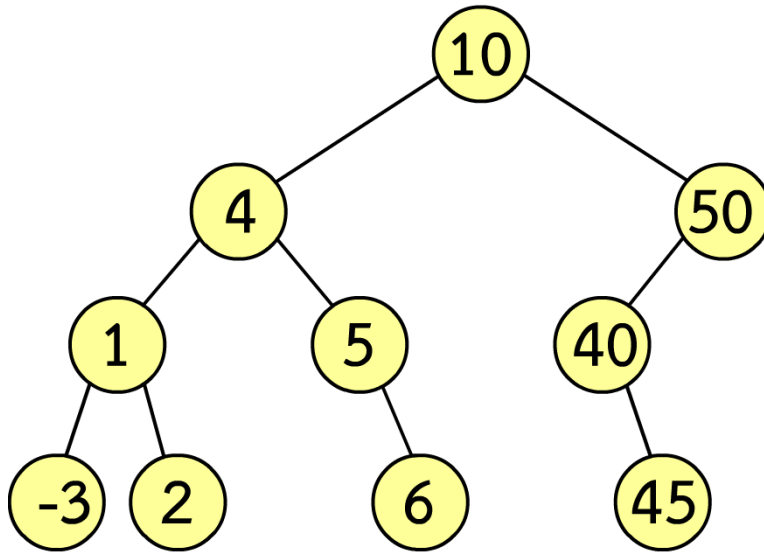


# Binary Tree

1. จาก tree ต่อไปนี้จงตอบคำถาม



1. Node ที่เป็น Root ได้แก่ .....
2. Node ที่เป็น leaf ได้แก่ .....
3. เมื่อพิจารณา Node ที่มีค่าเป็น 4 Node ไต่บ้างที่เป็น Child ของ node ดังกล่าว .....
4. เมื่อพิจารณา Node ที่มีค่าเป็น 4 Node ไต่บ้างที่เป็น Parent ของ node ดังกล่าว .....
5. เมื่อพิจารณา Node ที่มีค่าเป็น 5 Node ไต่บ้างที่เป็น Parent ของ node ดังกล่าว .....
6. เมื่อพิจารณา Node ที่มีค่าเป็น 5 Node ไต่บ้างที่เป็น Parent ของ node ดังกล่าว .....
7. หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น .....
8. หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น .....
9. หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น .....

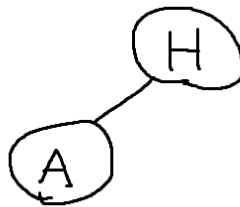
2. จงเขียนแผนภาพของการทำงานของ Binary Tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว

```
1. Node *root = new Node('H');
2. root->left = new Node('A');
3. root->right = new Node('I');
4. (root->left) ->left = new Node('R');
5. (root->left) ->right = new Node('U');
6. (root->right)->left = new Node('5');
7. ((root->left)->right)->left = new Node('H');
8. ((root->right)->left)->right = new Node('5');
```

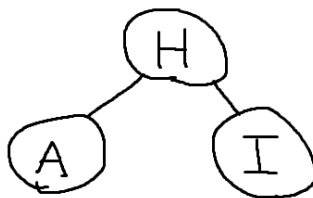
1.



2.



3.



4.

5.

6.

7.

8.

หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น .....

หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น ..... R A H U H 5 5 I.....

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น .....

3. จงเขียนแผนภาพของการทำงานของ Binary Tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว (ออกแบบบรรทัดเองเลยครับ)

```
1. Node *root = new Node('n');
2. root->left = new Node('A');
3. root->right = new Node('a');
4. (root->left) ->left = new Node('t');
5. (root->left) ->right = new Node('k');
6. (root->right)->left = new Node('i');
7. ((root->left)->right)->left = new Node('c');
8. ((root->right)->left)->right = new Node('t');
9. ((root->left)->left)->left = new Node('A');
10. ((root->left)->left)->right = new Node('t');
11. ((root->left)->right)->right = new Node('O');
12. ((root->right)->left)->left = new Node('T');
13. ((root->right)->right) = new Node('4');
14. ((root->right)->right)->left = new Node('n');
15. ((root->right)->right)->right = new Node('4');
```

หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น .....

หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น .....

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น .....









4. จงเขียนแผนภาพของการทำงานของ Binary Tree ในโปรแกรมต่อไปนี้ที่ละบรรทัด และตอบคำถามเกี่ยวกับการท่อง (Traversal) ไปใน tree ดังกล่าว (ออกแบบบรรทัดเองเลยครับ)

```
1. Node *root = new Node('H');
2. root->right = new Node('i');
3. ((root->right)->right) = new Node('g');
4. ((root->right)->right)->right = new Node('h');
5. (((root->right)->right)->right)->right = new Node('5');
```

หาก travers tree ดังกล่าว แบบ Pre-order จะได้ output เป็น .....

หาก travers tree ดังกล่าว แบบ In-order จะได้ output เป็น .....

หาก travers tree ดังกล่าว แบบ Post-order จะได้ output เป็น .....







5. Tree มีข้อดีกว่า Linked list อย่างไร (ขอสั้นๆ)

.....

.....

6. Linked list มีข้อดีกว่า Tree อย่างไร (ขอสั้นๆ)

.....

.....

7. ในการ insert และ find ข้อมูลใน binary tree มีความต่างในด้านประสิทธิภาพต่างจาก linked list queue และ stack หรือไม่ (ขอยาวๆ)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....