Operations on Binary Tree - Part II

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Vishvadeep Gothi • Lesson 47 • Jan 25, 2023

{ Data structure PYQ Vishvadeep

{ stack , queue ⇒ PYQs

Data Structure:
Tree 4

By: Vishvadeep Gothi

Hello!

I am Vishvadeep Gothi

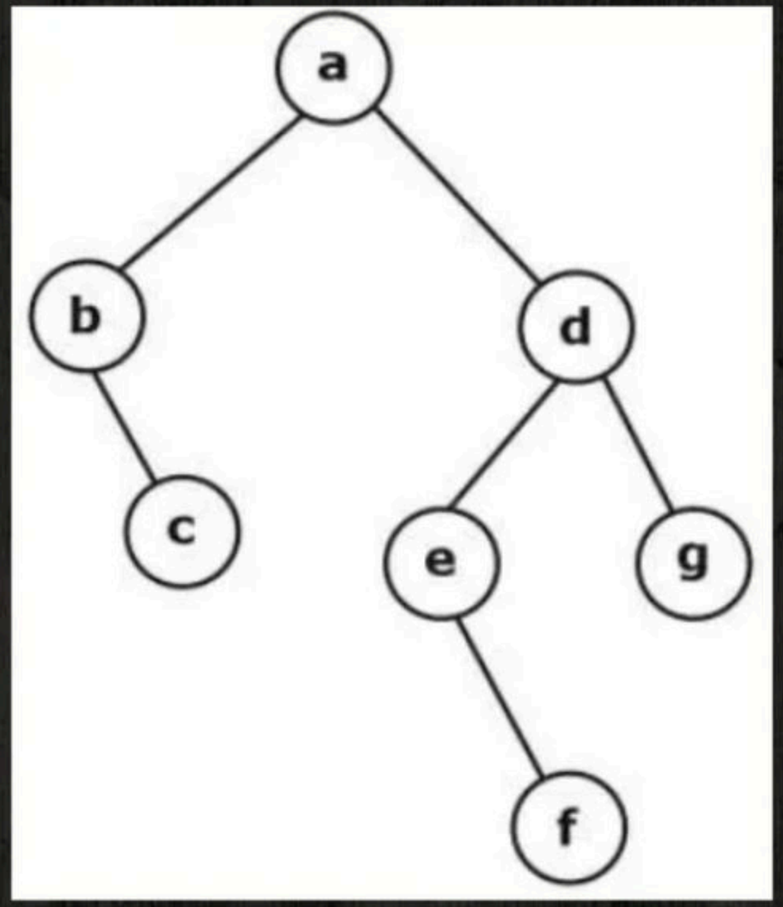
I am here because I love to teach

Question

What does the following return

```
int getvalue (struct BTNode *t)
{
int value = 0;
if (t)
   {
        If (((t → Lc) = = Null) & & (t → Rc = = Null))
                Value = 1;
        else
                value = value + Getvalue (t → Lc) + Getvalue (t → Rc);
   }
return value;
}
```

Question



Consider the following routine on binary tree.

```
Void Do (Struct btnode *t)
{
    if (t)
    {
        Do (t → Leftchild);
        Do (t → Rightchild);
        Swap(t → Leftchild, t → Rightchild);
    }
}
```

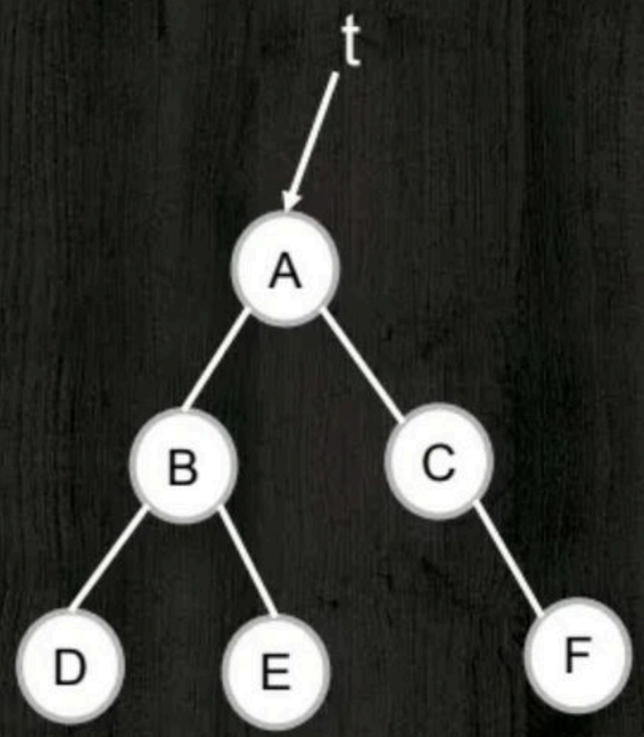
What will be the final tree after the above function is executed for given tree.

Question

What does the following return

```
void m(struct BTNode *t)
{
  If(t)
  {
      n(t → Leftchild);
      printf("%c", t → data);
      n(t → Rightchild);
  }
}
```

```
void n(struct BTNode *t)
{
  If(t)
  {
      printf("%c", t → data);
      m(t → Leftchild);
      m(t → Rightchild);
  }
}
```



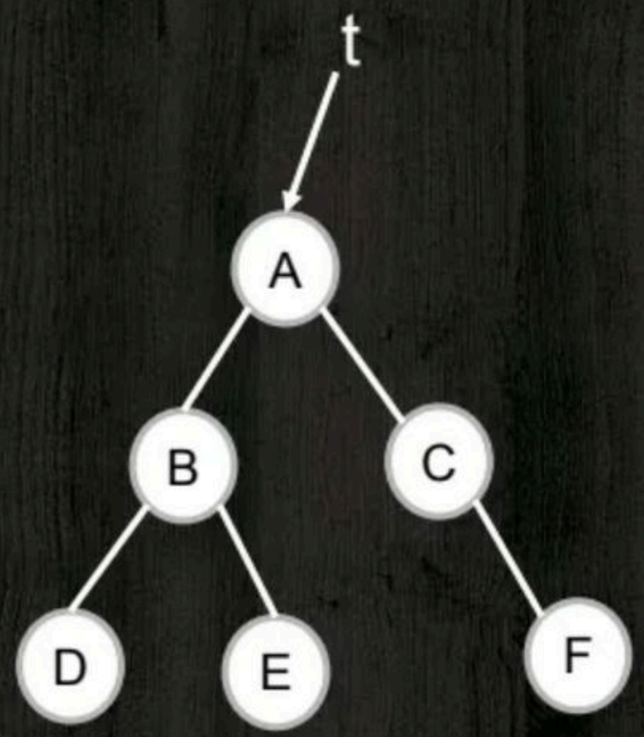
Get the value of m(t) for the given tree?

Question

What does the following return

```
void m(struct BTNode *t)
{
  If(t)
  {
      n(t → Leftchild);
      printf("%c", t → data);
      n(t → Rightchild);
  }
}
```

```
void n(struct BTNode *t)
{
  If(t)
  {
      printf("%c", t → data);
      m(t → Leftchild);
      m(t → Rightchild);
  }
}
```



Get the value of n(t) for the given tree?

Question

What does the following return

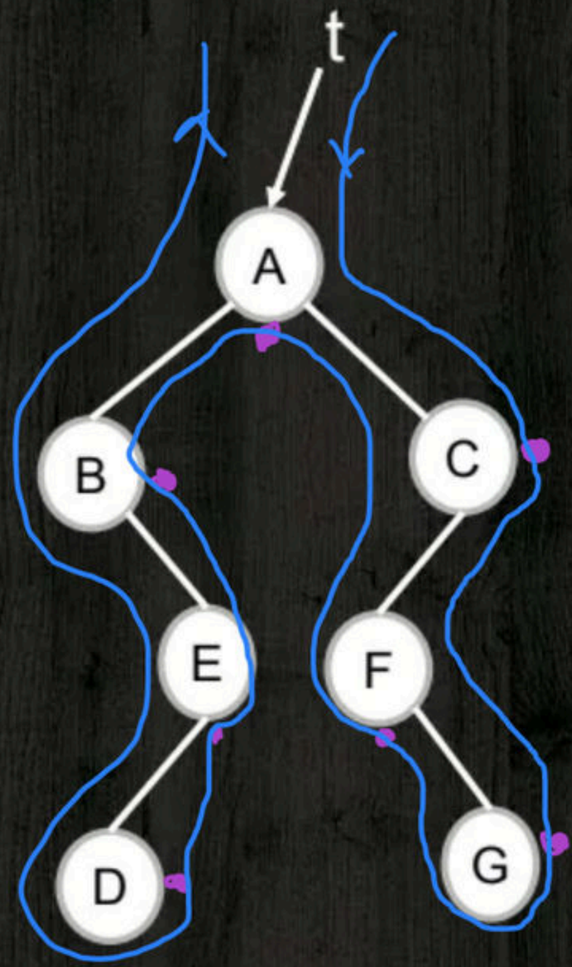
```
void m(struct BTNode *t)
{
  If(t)
  {
     n(t → Rightchild);
     printf("%c", t → data);
     n(t → Leftchild);
  }
}
```

Converse inorder

```
void n(struct BTNode *t)
{
  If(t)
  {
     printf("%c", t → data);
     m(t → Rightchild);
     m(t → Leftchild);
  }
}
```

Converse Pre

Get the value of m(t) for the given tree?



CGFABED

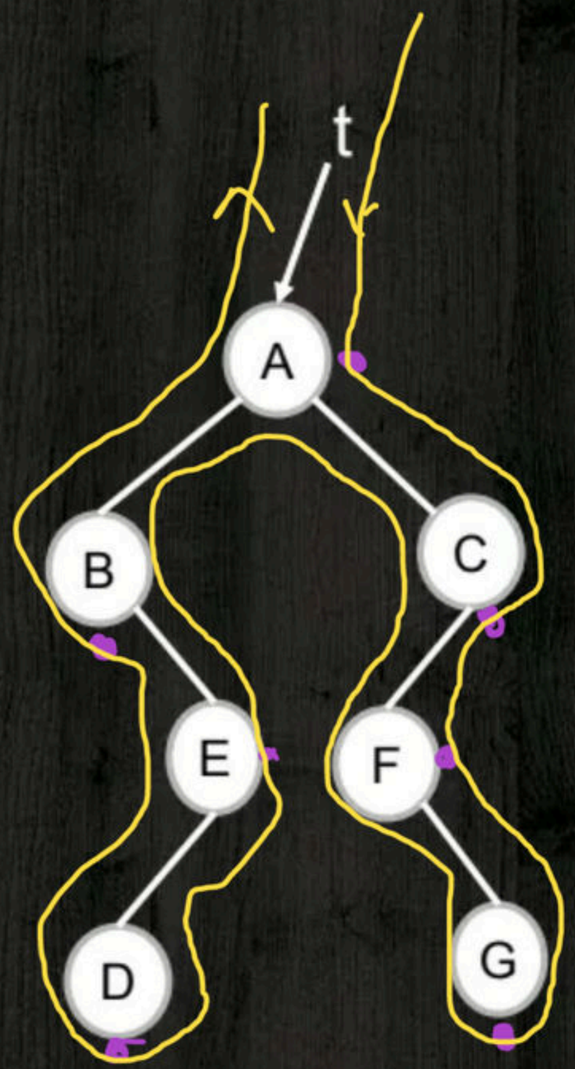
Question

What does the following return

```
void m(struct BTNode *t)
{
  If(t)
  {
      n(t → Rightchild);
      printf("%c", t → data);
      n(t → Leftchild);
  }
}
```

```
void n(struct BTNode *t)
{
  If(t)
  {
      printf("%c", t → data);
      m(t → Rightchild);
      m(t → Leftchild);
  }
}
```

Get the value of n(t) for the given tree?



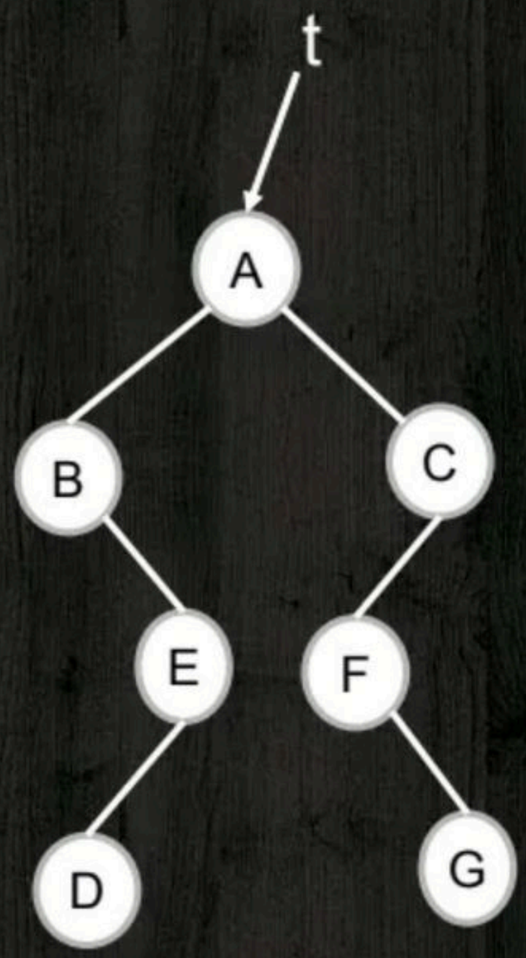
ACFGEDB

Question 3 pp 1

What does the following return

```
void m(struct BTNode *t)
{
  If(t)
  {
      n(t → Leftchild);
      printf("%c", t → data);
      n(t → Rightchild);
  }
}
```

```
void n(struct BTNode *t)
{
  If(t)
  {
      printf("%c", t → data);
      m(t → Rightchild);
      m(t → Leftchild);
  }
}
```



Get the value of m(t) for the given tree?

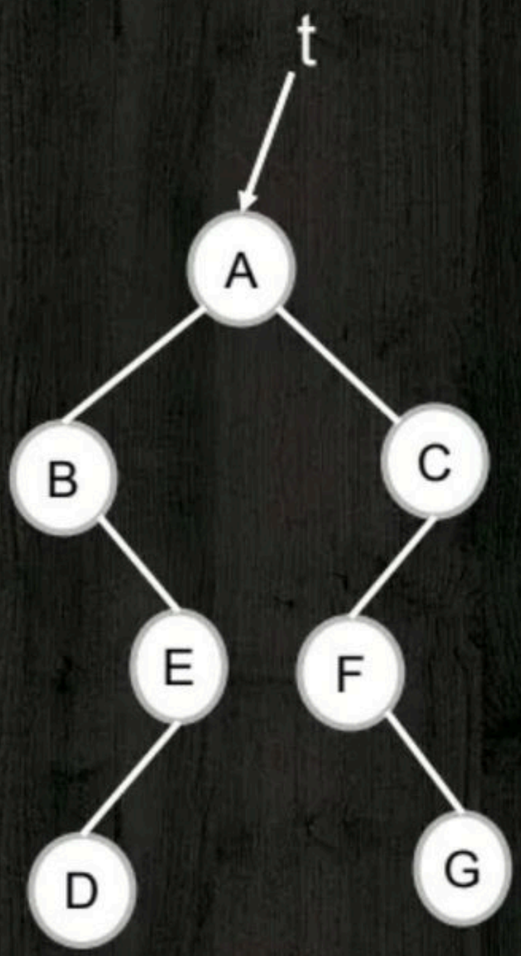
Question 1 PP 2

What does the following return

```
void m(struct BTNode *t)
{
  If(t)
  {
      n(t → Rightchild);
      printf("%c", t → data);
      n(t → Leftchild);
  }
}
```

```
void n(struct BTNode *t)
{
  If(t)
  {
      printf("%c", t → data);
      m(t → Leftchild);
      m(t → Rightchild);
  }
}
```

Get the value of m(t) for the given tree?



~~Conversion~~ of General Tree to Binary

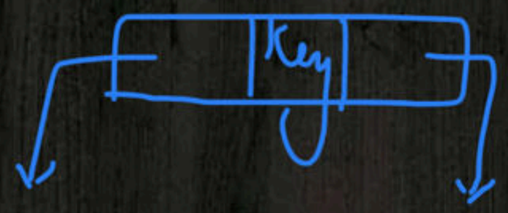
Representation

Binary Tree

every node

⇓

max 2 children



General tree

every node ⇒ max children

⇓

unlimited

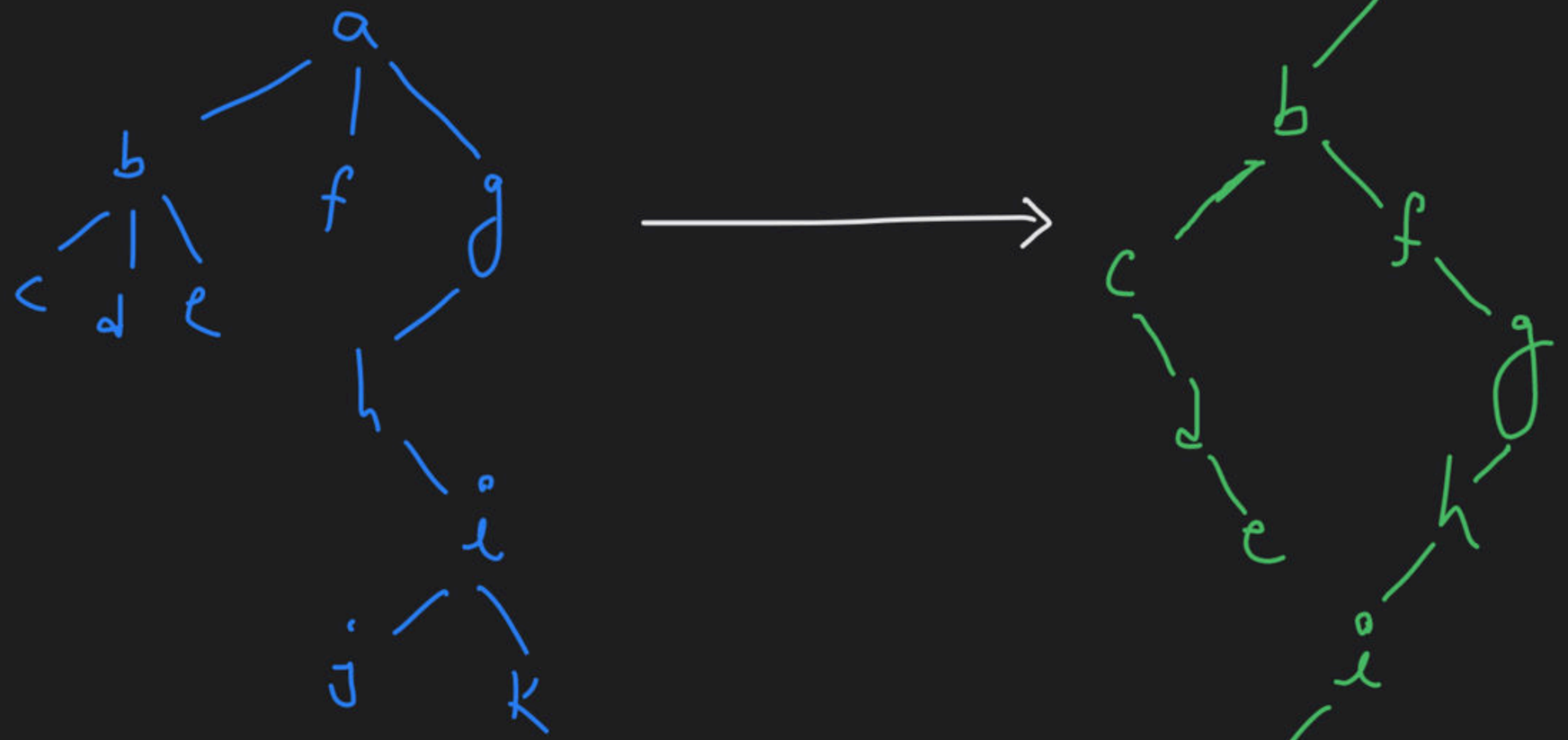
linked representation

not possible

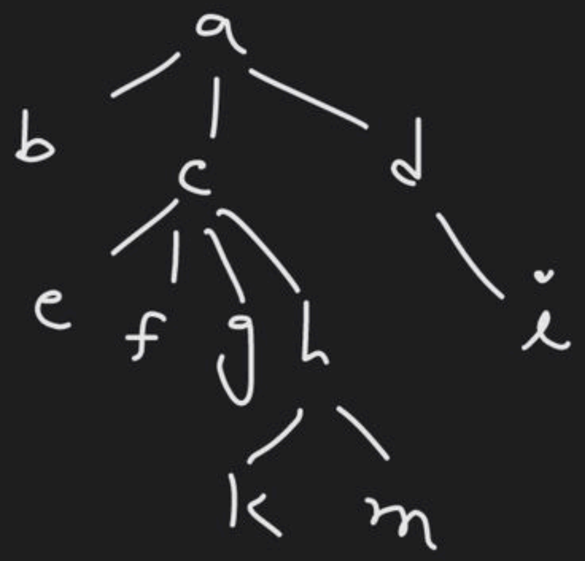
⇓

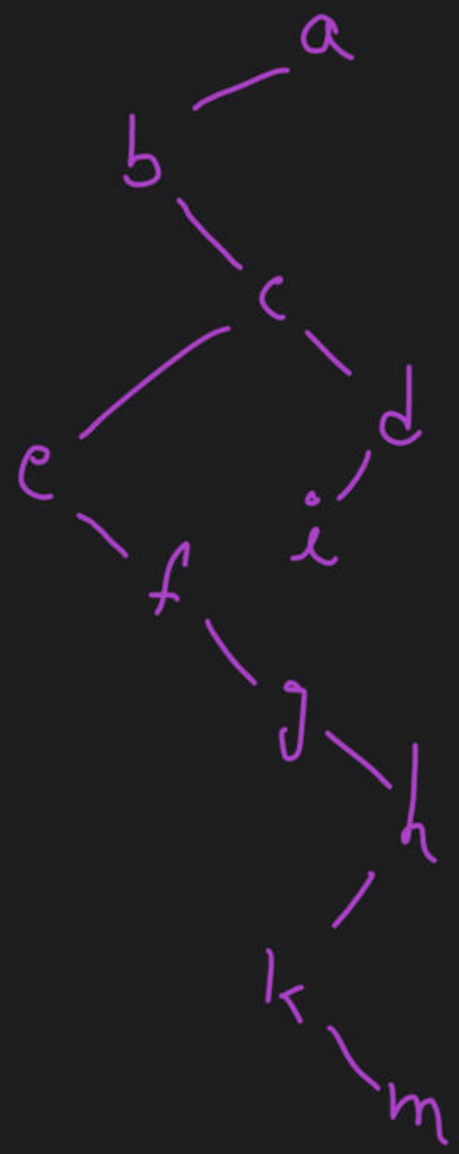
better parenthesis representation

Leftmost child - Rightsibling Representation:-



```
struct LMCRSnode
{
 char key;
 struct LMCRSnode *leftmostChild;
 struct LMCRSnode *RightSibling;
};
```





Leaf node :-

`if (t -> Leftmostchild == NULL)`

t points to leaf node

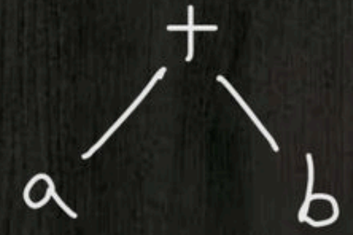
Expression Tree

operator => root

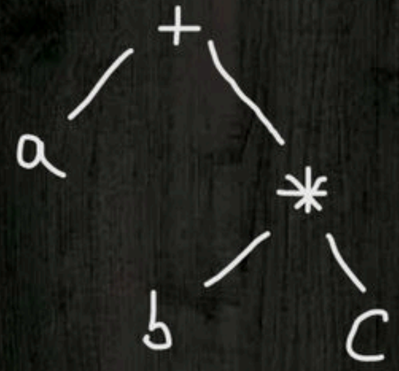
left operand => left child

Right -"- => Right -"-

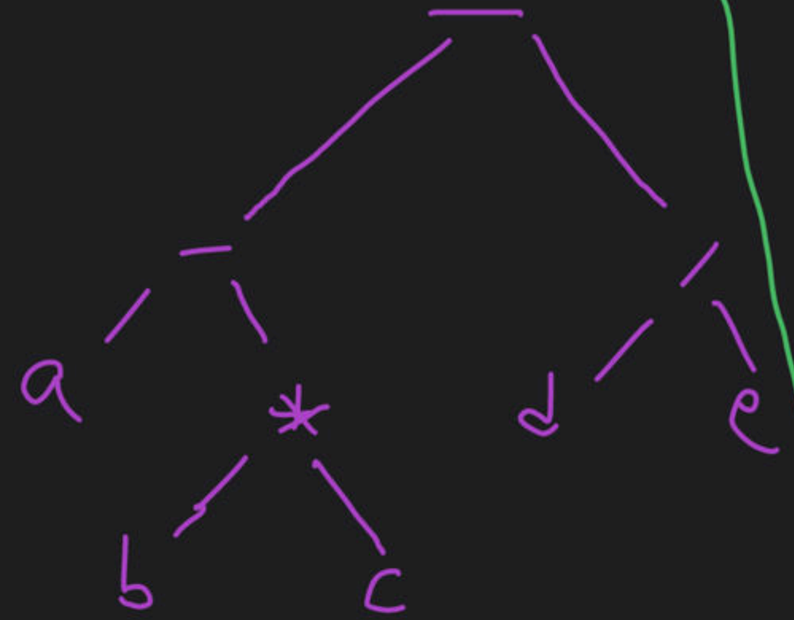
a+b



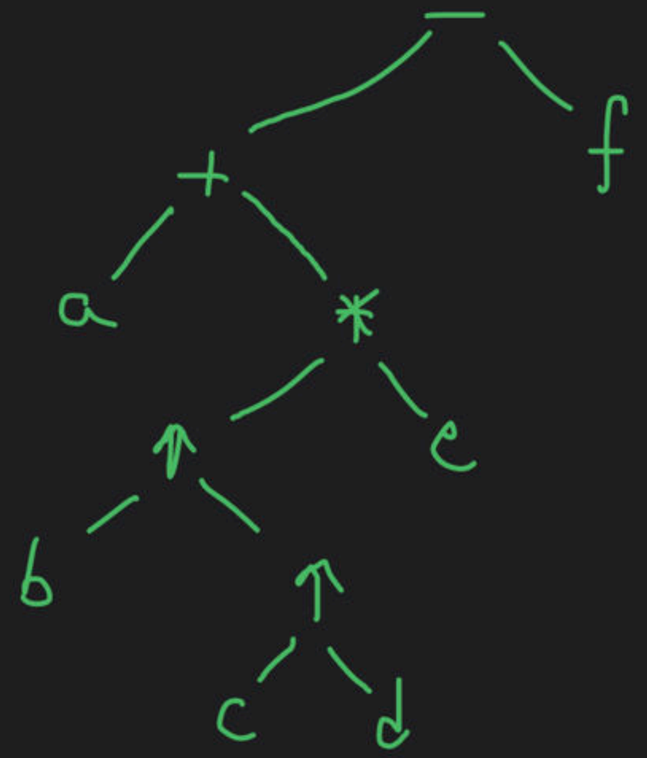
a + b * c



a - b * c - d / e



a + b ↑ c ↑ d * e - f

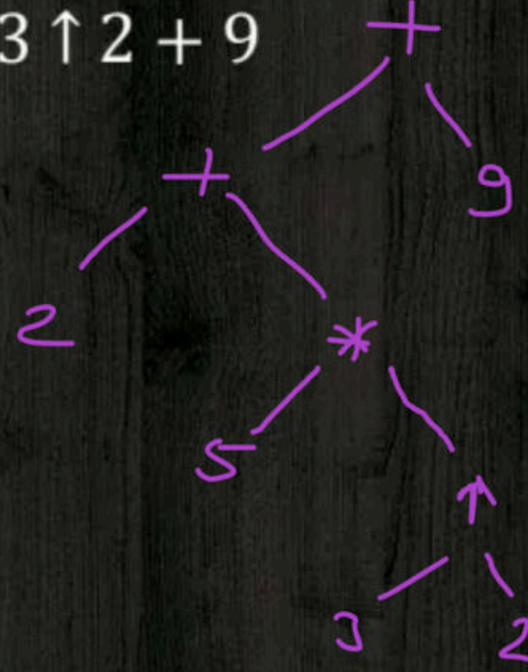


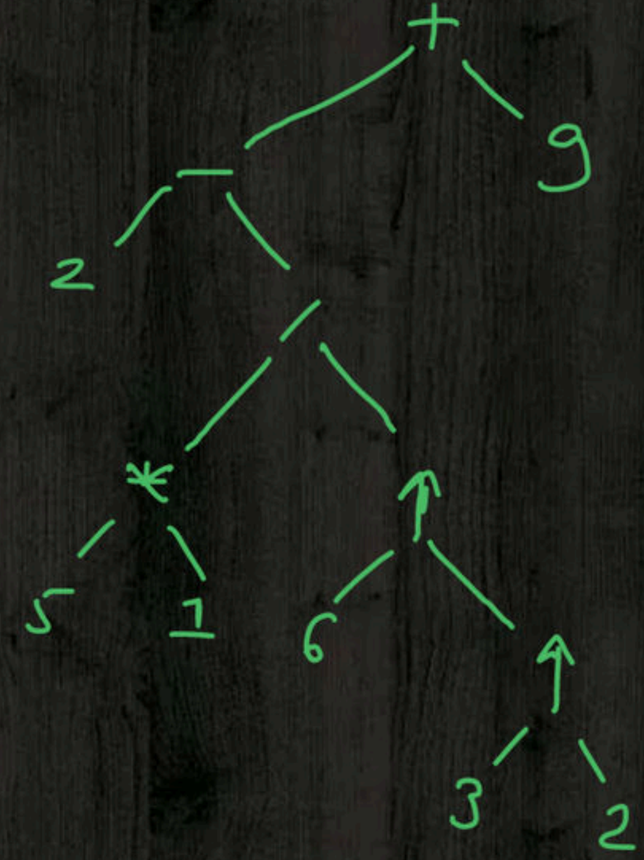
Question

Draw expression tree for:

1. $2 + 5 * 3 \uparrow 2 + 9$

2. $2 - 5 * 1 / 6 \uparrow 3 \uparrow 2 + 9$



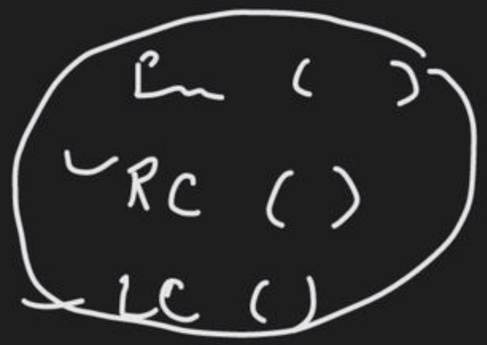


→ Inorder traversal of expression-tree ⇒ Infix notatn

→ Preorder — 11 — 11 — ⇒ Prefix — 11 —

→ Postorder — 11 — ⇒ Postfix — 11 —

Eg



Converse preorder

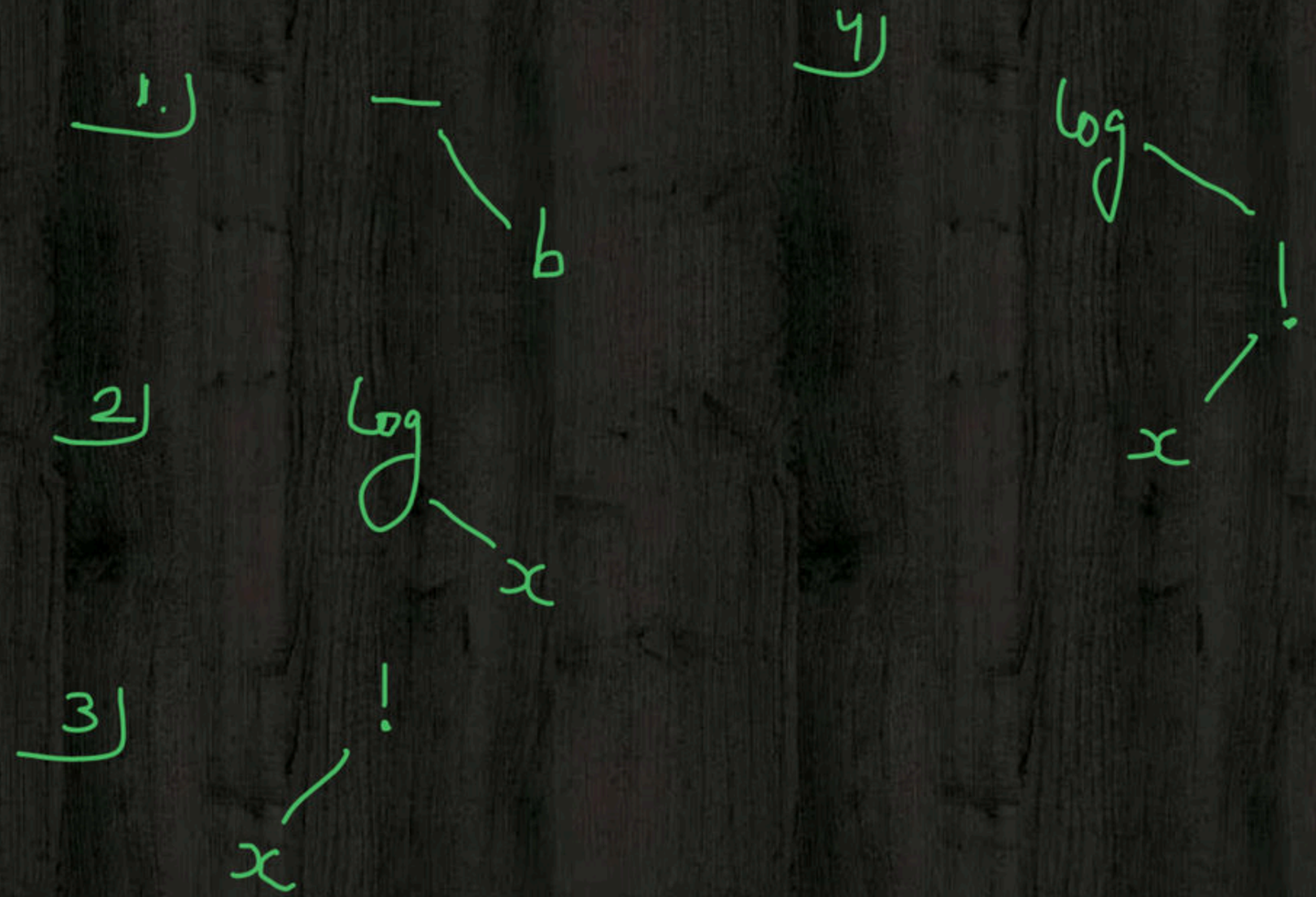
Reverse

⇒ Postfix

Question

Draw expression tree for:

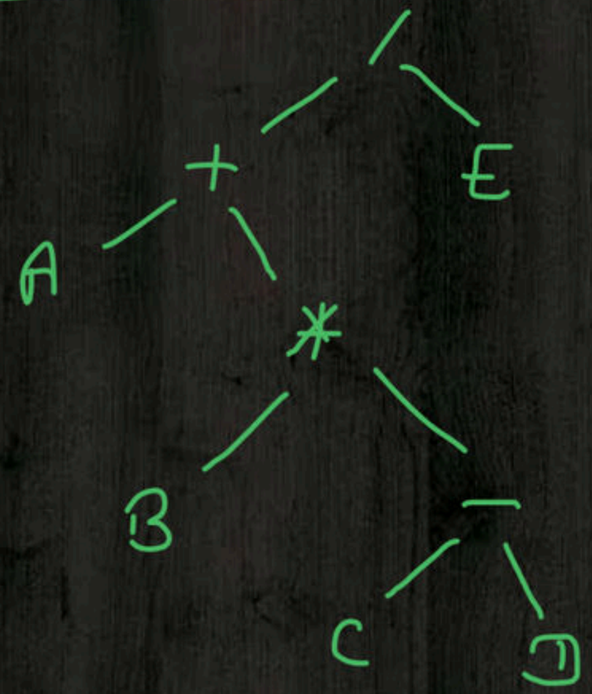
1. −b
2. *log x*
3. *x!*
4. *log x!*



Question

Draw expression tree for:

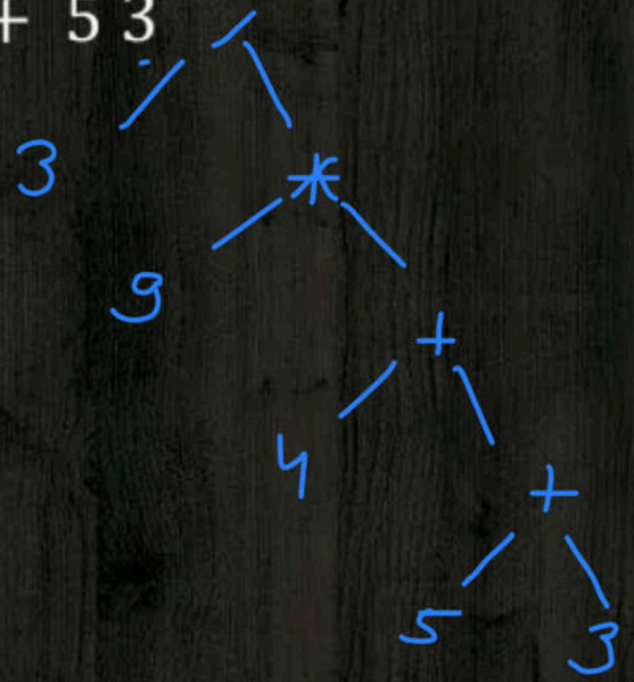
$A\ B\ C\ D - * + E\ /$



Question

Draw expression tree for:

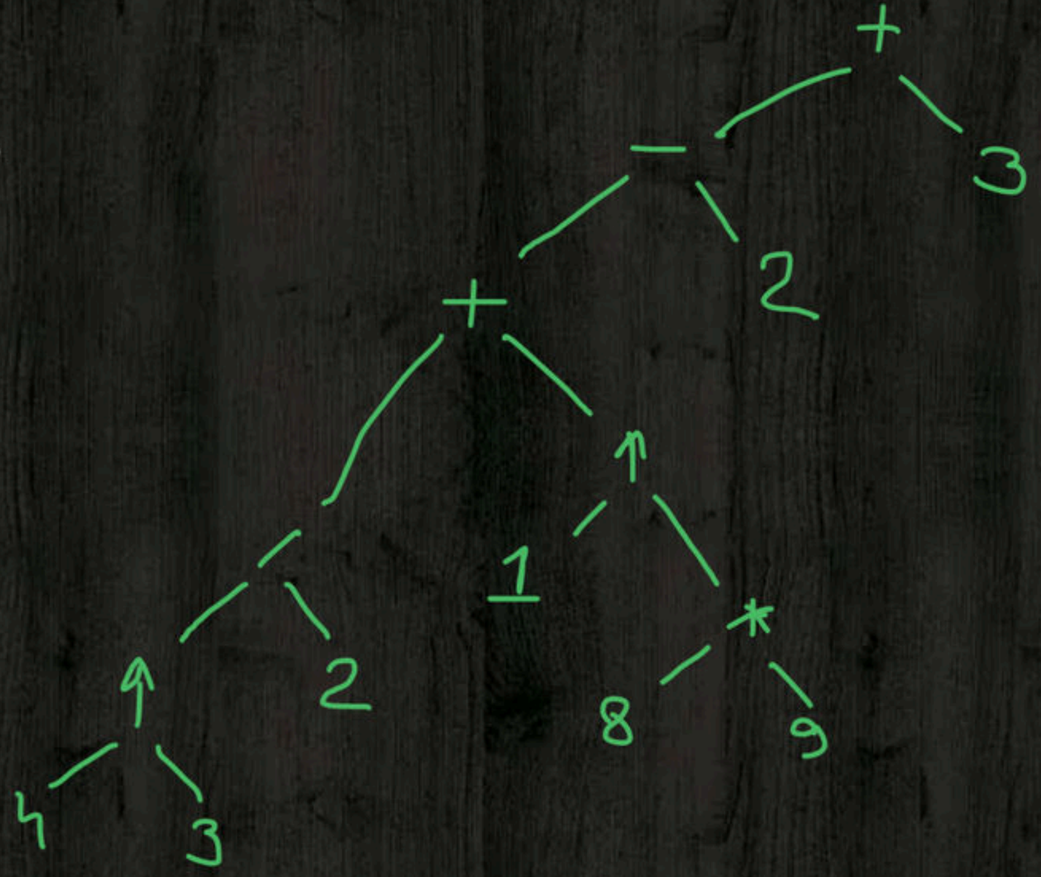
/ 3 * 9 + 4 + 5 3



Question

Draw expression tree for:

4 3 ↑ 2 / 1 8 9 * ↑ + 2 - 3 +

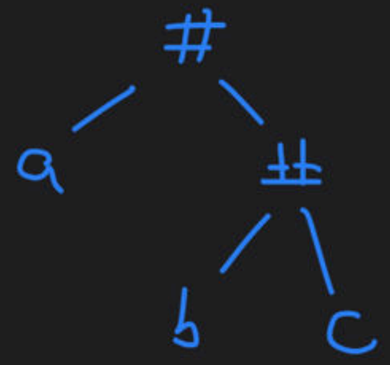


Ques:- Given following expression tree. which operator has higher precedence?

$ => lower precedence

=> higher precedence

Ques) Given expression tree :-



a # (b # c)

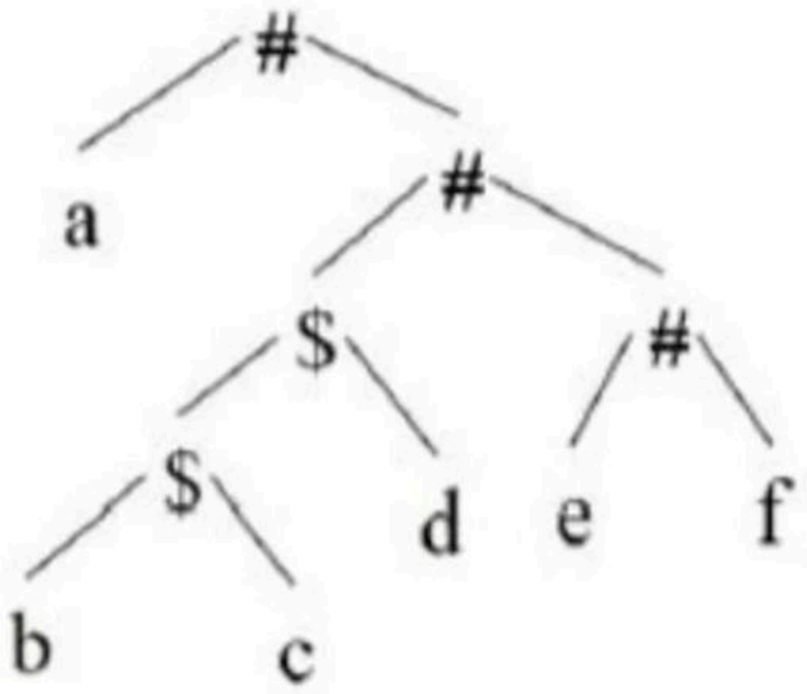
associativity of #

⇓

Right to left

GATE - 2018

Consider the following parse tree for the expression a#bcd#e#f, involving two binary operators $ and #.



$ ⇒ higher

$ ⇒ left ass.

⇒ left ass.

Which one of the following is correct for the given parse tree?

(A) $ has higher precedence and is left associative; # is right associative

(B) # has higher precedence and is left associative; $ is right associative

(C) $ has higher precedence and is left associative; # is left associative

(D) # has higher precedence and is right associative; $ is left associative

Consider the following New-order strategy for traversing a binary tree:

- Visit the root;
- Visit the right subtree using New-order;
- Visit the left subtree using New-order;

Converse preorder

The New-order traversal of the expression tree corresponding to the reverse polish expression

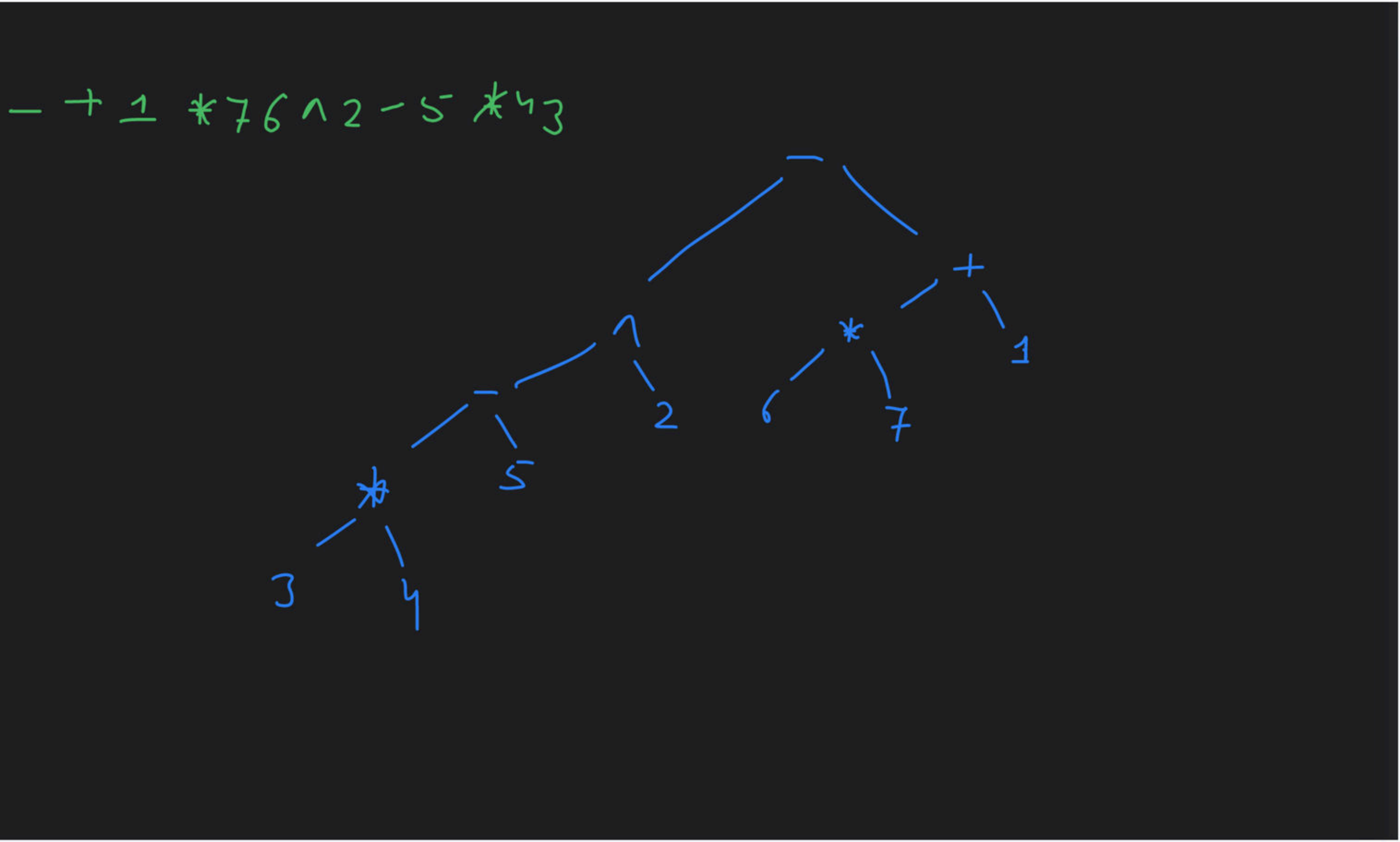
```
3 4 * 5 - 2 ^ 6 7 * 1 + -
```

postfix

postfix ⟹ (reverse) Converse prefix

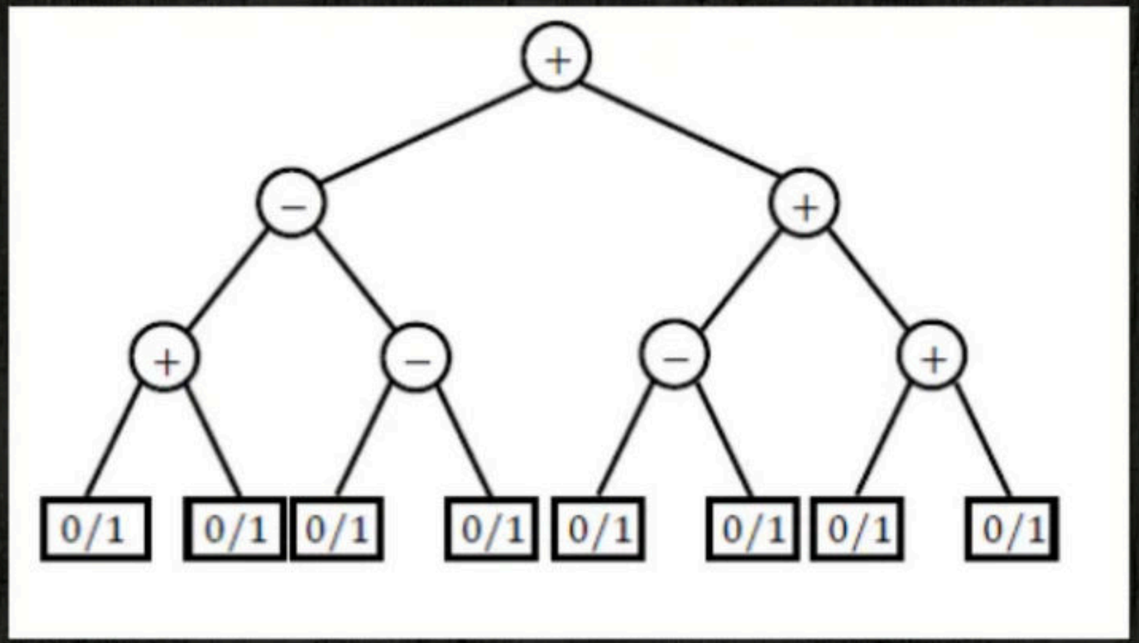
is given by:

A. $+-167*2\wedge5-34*$
B. $-+1*67\wedge2-5*34$
C. $-+1*76\wedge2-5*43$
D. $176*+2543*-\wedge-$


- + 1 * 7 6 ^ 2 - 5 * 4 3
-
+
1
*
7
6
^
2
-
5
*
4
3

Question GATE-2014

Consider the expression tree shown. Each leaf represents a numerical value, which can either be 0 or 1. Over all possible choices of the values at the leaves, the maximum possible value of the expression represented by the tree is ___.



DPP

Question

Draw expression tree for:

1. (A + B) * (C – D) / F – X * Y / Z
2. a + b * c – d ^ e ^ f
3. $\boldsymbol{A+B*(C+D)/F+D*E}$
4. $\boldsymbol{3*log(x+1)-a/2}$
5. a= − b + c / d ↑ e ↑ f + g * h / i − j * k

Happy Learning