

TreeExample-1

create()

1

1

|

2

1
/
2
(no branch)

1
/ \
2 3
(no branch)

1 -> p
/ \
2 3
(no branch)

preorder()

1 -> t
/ \
2 3

Output:

1

2

3

inorder()

1 -> t
/ \
2 3

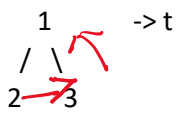
Output:

2

1

3

postorder()



Output:

2

3

1

TreeExample-2-char

create()

-

-

|

+

-

|

+

|

a

-

|

+

|

a

(no branch)

-

/

+

/ \

a *

$$\frac{-}{\frac{+}{\frac{a}{\backslash} *}} \frac{1}{b}$$

$$\frac{-}{\frac{+}{\frac{a}{\backslash} *}} \frac{1}{b}$$

(no branch)

$$\frac{-}{\frac{+}{\frac{a}{\backslash} *}} \frac{1}{b -}$$

$$\frac{-}{\frac{+}{\frac{a}{\backslash} *}} \frac{1}{b -} \frac{1}{c}$$

$$\frac{-}{\frac{+}{\frac{a}{b} \cdot \frac{c}{d}}}$$

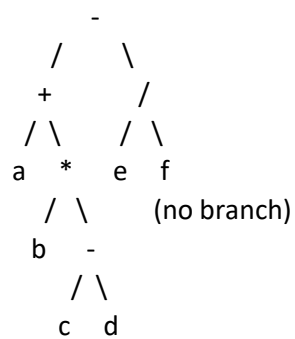
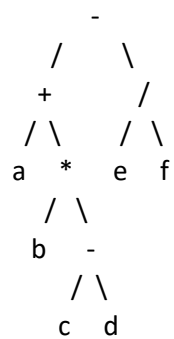
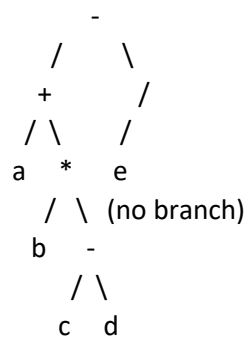
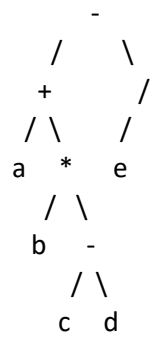
(no branch)

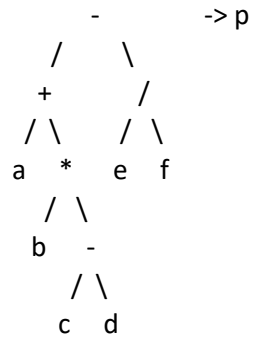
$$\frac{-}{\frac{+}{\frac{a}{b} \cdot \frac{c}{d}}}$$

$$\frac{-}{\frac{+}{\frac{a}{b} \cdot \frac{c}{d}}}$$

(no branch)

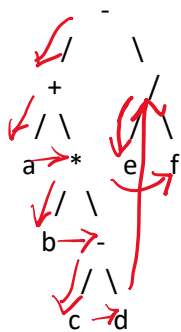
$$\frac{-}{\frac{+}{\frac{a}{b} \cdot \frac{c}{d}}}$$





preorder()

-> root (displayed on screen as it moves through the tree)

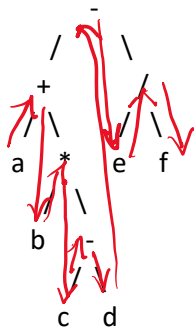


Output:

-+a*b-cd/ef

inorder()

a is the root

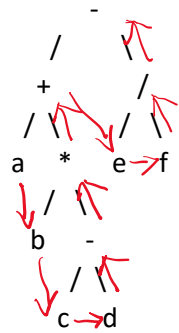


Output:

a+b*c-d-e/f

postorder()

a is the root



Output:

abcd-*+ef/-

TreeExample-2-int

create()

1

1

|

2

1
/
2
(no branch)

1
/ \
2 3
(no branch)

1 -> p
/ \
2 3
(no branch)

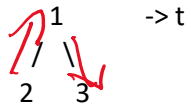
preorder()

1 -> t
/ \
2 3

Output:

1
2
3

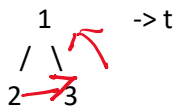
inorder()



Output:

2
1
3

postorder()



Output:

2
3
1

TreeExample-3

create()

10

10
/
11

10
/
11
/
13

```

      10
     /
    11
   /
  13
 /
17
(no branch)

```

```

      10
     /
    11
   /
  13
 / \
17  18
(no branch)

```

```

      10
     /
    11
   / \
  13  14
 / \ (no branch)
17  18

```

```

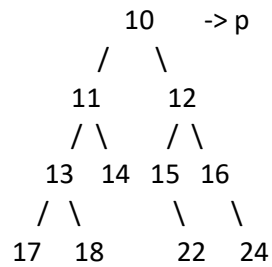
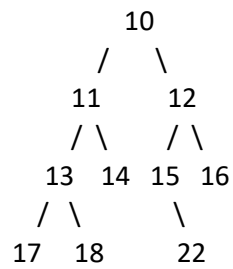
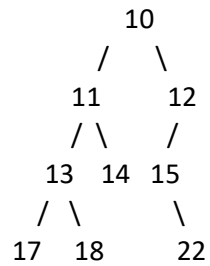
      10
     / \
    11  12
   / \
  13  14
 / \
17  18

```

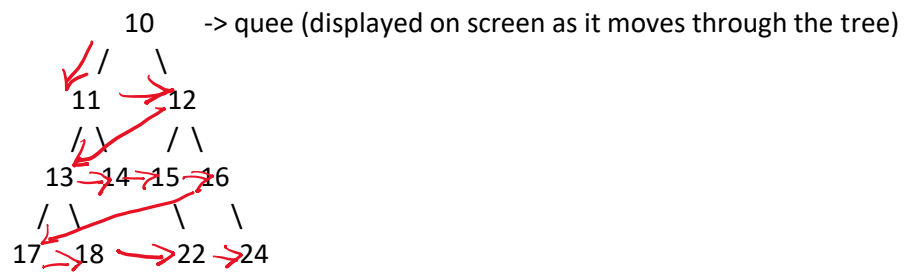
```

      10
     / \
    11  12
   / \ /
  13  14 15
 / \
17  18

```



Levelorder()



Output:

10 11 12 13 14 15 16 17 18 22 24

TreeExample-4

create()

left branch = if label % 2 == 0

- (1)

0 /

+(2)

```

      - (1)
    0 /   \ 1
  + (2)   / (3)

```

```

      - (1)
    0 /   \ 1
  + (2)   / (3)
0 /
a (4)

```

```

      - (1)
    0 /   \ 1
  + (2)   / (3)
0 / \ 1
a (4) b (5)

```

```

      - (1)
    0 /   \ 1
  + (2)   / (3)
0 / \ 1  0 /
a (4) b (5) e (6)

```

```

      - (1)  -> T
    0 /   \ 1
  + (2)   / (3)
0 / \ 1  0 / \ 1
a (4) b (5) e (6) f (7)

```

preorder()

- (1) -> root (displayed on screen as it moves through the tree)

```

    0 / \ 1
  + (2) / (3)
0 / \ 1 0 / \ 1
a (4) b (5) e (6) f (7)

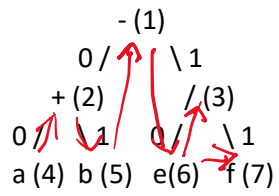
```

Output:

-+ab/ef

inorder()

a as root

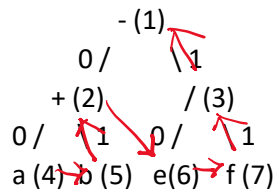


Output:

a+b-e/f

postorder()

a as root



Output:

ab+ef/-

TreeExample-5

create()

-

-
/
+

-
/
+
/
a
(no branch)

-
/
+
/ \
a *

$$\begin{array}{c} - \\ / \\ + \\ / \quad \backslash \\ a \quad * \\ / \\ b \end{array}$$

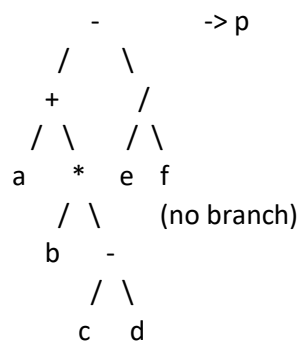
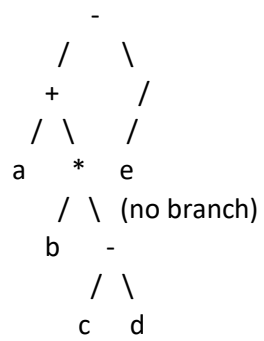
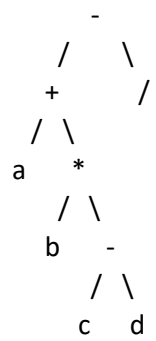
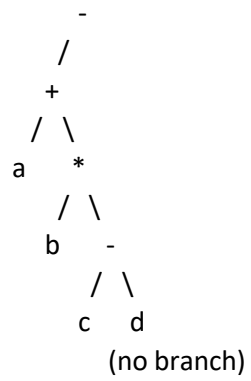
$$\begin{array}{c} - \\ / \\ + \\ / \quad \backslash \\ a \quad * \\ / \\ b \end{array}$$

(no branch)

$$\begin{array}{c}
 - \\
 / \\
 + \\
 / \quad \backslash \\
 a \quad * \\
 / \quad \backslash \\
 b \quad -
 \end{array}$$

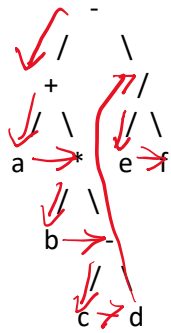
$$\begin{array}{c}
 - \\
 / \\
 + \\
 / \quad \backslash \\
 a \quad * \\
 / \quad \backslash \\
 b \quad - \\
 / \\
 c
 \end{array}$$

(no branch)



preorder()

-> root (displayed on screen as it moves through the tree)

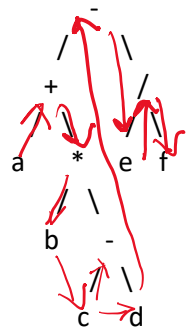


Output:

-+a*b-cd/ef

inorder()

a as root

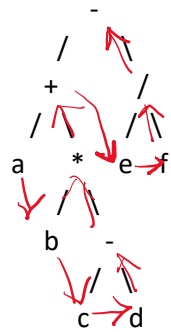


Output:

a+b*c-d-e/f

postorder()

a as root

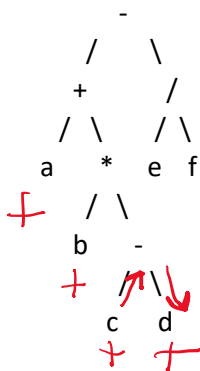
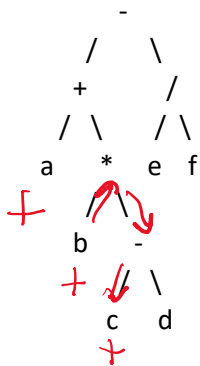
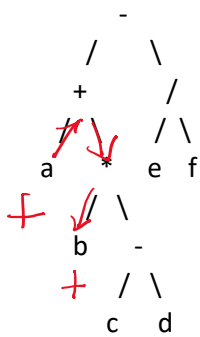
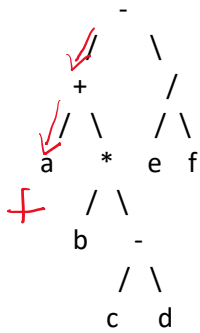


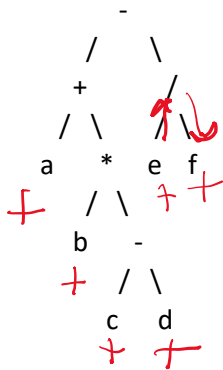
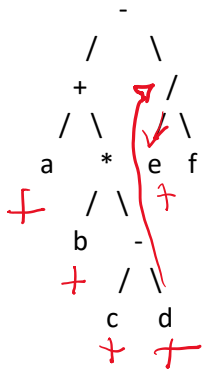
Output:

abcd-*+ef/-

search_leaves()

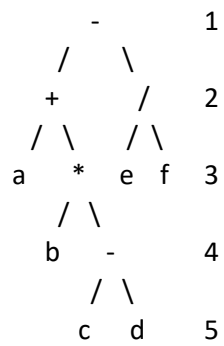
stack[top] moves around the tree and when reaching a leaf, num increases by 1 (as denoted by +)





Returns value of num which is 6

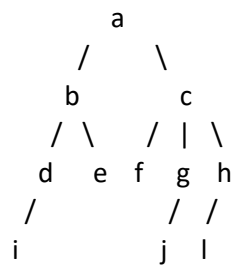
TreeDepth()



Return value of depth which is 5

Q2

{ (e,i), (b,e), (b,d), (a,b), (g,j), (c,g), (c,f), (h,l), (c,h), (a,c) }. Convert this tree structure to diagram.



In which level is the node b & node c?

2

What is depth of this tree?

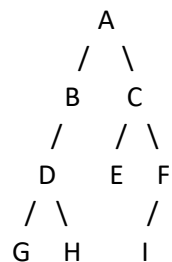
4

What is depth of subtree for the node c as root?

3

Q3

Given the pre-ordered / in-ordered traversal sequences are ABDGHCEFI and GDHBAECIF respectively in a binary tree, draw a tree structure diagram for it.



Q4

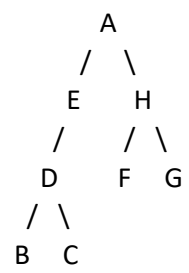
Draw a tree structure diagram and verify it by the programs if the in-ordered / post-ordered traversal sequences are BDCEAFHG and DECBHGFA respectively in a binary tree

(Needs two different trees to solve this question)

```
data {7 101 no branch}
Tree structure (inorder):
B D C E A F H G
Tree structure (postorder):
B C D E F G H A
Create second tree
```

```
Tree structure 2 (inorder):
D E B C A H F G
Tree structure 2 (postorder):
D E C B H G F A
```

Tree 1:



Tree 2:

