Q1) **What are some of the applications for the tree data structure?**

Answer:

1- Manipulation of the arithmetic expressions.
2- Symbol table construction.
3- Syntax analysis.

…………………………………………………………………

Q2) **There are 8, 15, 13, and 14 nodes in 4 different trees. Which one of them can form a full binary tree?**

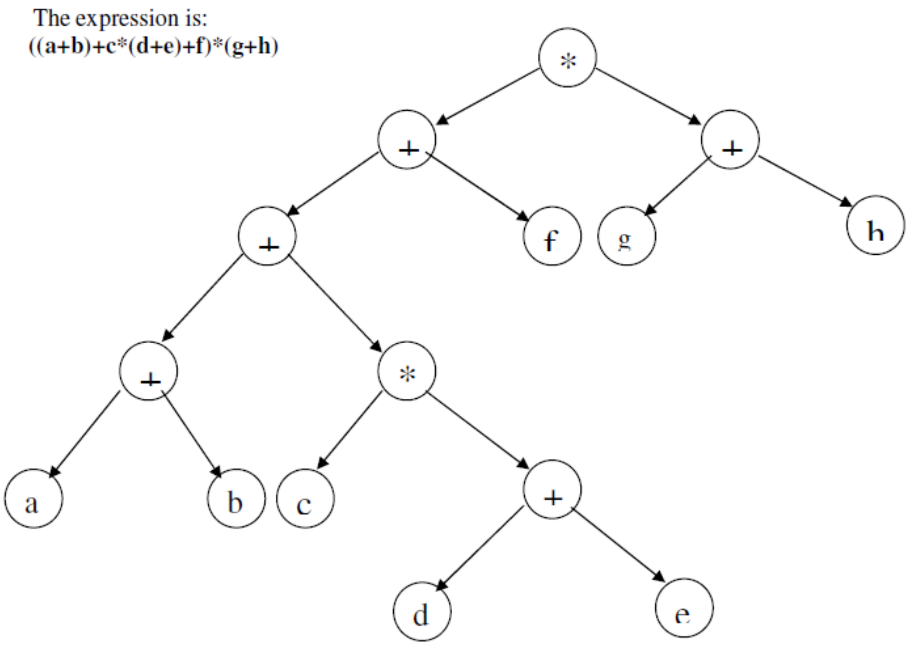
Answer:

The answer is the tree with **15 nodes**.

Full binary trees contain odd number of nodes, so there cannot be full binary trees with 8 or 14 nodes. Moreover, with 13 nodes you can form a complete binary tree but not a full binary tree. Thus, the correct answer is 15.

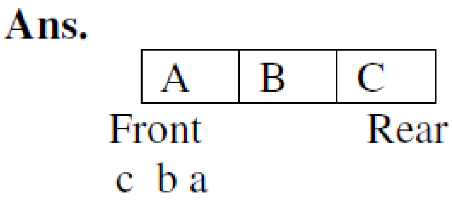
……………………………………………………………………………………….

Q3) Draw the expression tree of the following infix expression.


The expression is:
((a+b)+c*(d+e)+f)*(g+h)

…………………………………………………………………………………………

Q4) Assume that a queue is available for pushing and popping elements. Given an input sequence **a**, **b**, **c**, give the output sequence of elements if the rightmost element given above is the first to be popped from the queue.



Ans.

A	B	C

Front Rear

c b a

……………………………………………………………………………..

Q5) How many nodes in a tree have no ancestors.

(A) 0 **(B)** 1
(C) 2 **(D)** n

Ans. (B)

………………………………………………………………………………

Q6) What is an algorithm? What are the characteristics of a good algorithm?

Ans:
An **algorithm** is “a step-by-step procedure for accomplishing some task" An algorithm can be given in many ways. For example, it can be written down in English (or French, or any other "natural" language). However, we are interested in algorithms which have been precisely specified using an appropriate mathematical formalism--such as a programming language.

Every algorithm should have the following five characteristics:

1.Input: The algorithm should take zero or more input.
2. Output: The algorithm should produce one or more outputs.
3. Definiteness: Each and every step of algorithm should be defined unambiguously.
4. Effectiveness: A human should be able to calculate the values involved in the procedure of the algorithm using paper and pencil.
5. Termination: An algorithm must terminate after a finite number of steps.

……………………………………………………………………………………………………..

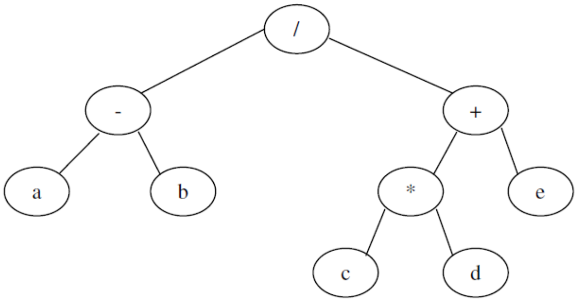
Q7) What are expression trees? Represent the following expression using a tree.

(a – b) / ((c * d) + e))

Ans:

The leaves of an expression tree are operands, such as constants or variable names, and the other nodes contain operators. This particular tree happens to be binary, because all of the operations are binary, and although this is the simplest case, it is possible for nodes to have more than two children. It is also possible for a node to have only one child, as is the case with the unary minus operator. We can evaluate an expression tree, T, by applying the operator at the root to the values obtained by recursively evaluating the left and right subtrees.

The expression tree for the expression: **(a – b) / ((c * d) + e))**



……………………………………………………………………………………………

Q8) Taking a suitable example explains how a general tree can be represented as a Binary Tree.

Ans:

Conversion of general trees to binary trees:

A general tree can be converted into an equivalent binary tree. This conversion process is called the natural correspondence between general and binary trees.

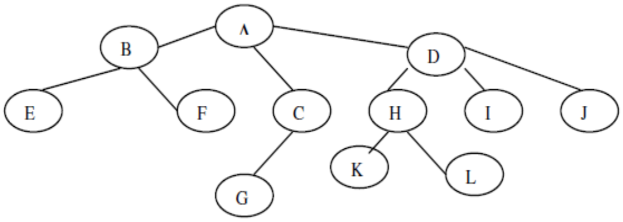
The algorithm is as follows:

(a) Insert edges connecting siblings from left to right at the same level.

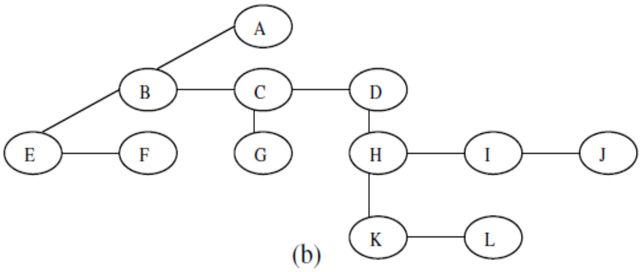
(b) Delete all edges of a parent to its children except to its left most offspring.

(c) Rotate the resultant tree 45_0 to mark clearly left and right subtrees.

A general tree shown in figure (a) converted into a binary tree shown in (b)



(a)



(b)

…………………………………………………………………………………….

Q9) Reverse the order of elements on a stack S by using two additional stacks.

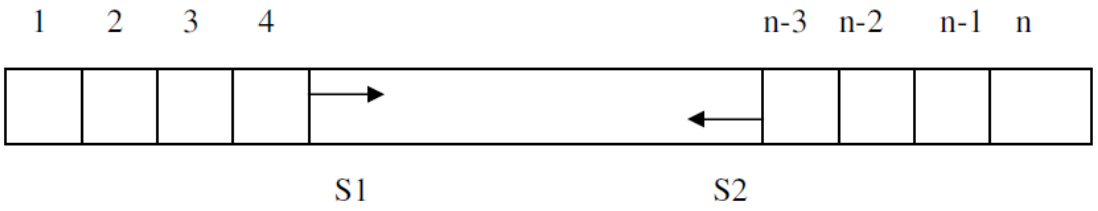
Ans: Let S be the stack having n elements. We need to reverse the elements of S by using two additional stack S1 and S2

```
while not empty (S)
{ C=pop(S);
push(S1,C);
}
While not empty(S1)
{ C=pop(S1);
push(S2,C);
}
While not empty(S2)
{ C=pop(S2);
push(S,C);
}
```

…………………………………………………………………………………………….

Q10) Explain how to implement two stacks in one array A[1..n] in such a way that neither stack overflows unless the total number of elements in both stacks together is n. The PUSH and POP operations should run in O(1) time.

Ans:
Two stacks s1 and s2 can be implemented in one array A[1,2,…,N] as shown in the following figure



We define A[1] as the bottom of stack S1 and let S1 “grow” to the right and we define A[n] as the bottom of the stack S2 and S2 “grow” to the left. In this case, overflow will occur only S1 and S2 together have more than n elements. This technique will usually decrease the number of times overflow occurs. There will separate push1, push2, pop1 and pop2 functions to be defined separately for two stacks S1 and S2.

…………………………………………………………………………………………………

Q.11) Two Binary Trees are similar if they are both empty or if they are both nonempty and left and right sub trees are similar. Write an algorithm to determine if two Binary Trees are similar.

Ans:
We assume two trees as tree1 and tree2. The algorithm to determine if two Binary trees are similar or not is as follows:

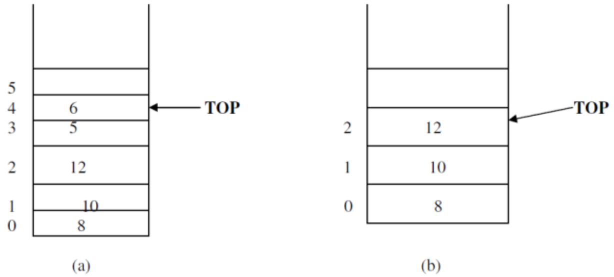
```
Similar (*tree1,*tree2)
{
While ((tree1!=null) && (tree2!=null))
{
If ((tree1->root) == (tree2->root))
similar (tree1->left,tree2->left);
similar (tree1->right,tree2->right);
elseif (tree->root!= tree2->node)
{
Printf (“Trees are not equal”);
return; }
}
}
```

……………………………………………………………………………

Q12). Define a stack. Describe ways to implement stack.

Ans.

STACK:
A stack is one of the most commonly used data structure. A stack is also called a Last-In-First-Out (LIFO) system, is a linear list in which insertion or deletion can take place only at one end called the top. This structure operates in the same way as the stack of trays If we want to place another tray, it can be placed only at the top. Similarly, if we want to remove a tray from stack of trays, it can only be removed from the top. The insertion and deletion operations in stack terminology are known as PUSH and POP operation.



(a) Stack after pushing elements 8,10,12,5,6
(b) Stack after popping 2 elements.

Implementation of Stack

Stacks can be implemented in the following way:
Arrays

To implement a stack we need a variable called top that holds the index of the top element of stack and an array to hold the element of the stack.

For example:-

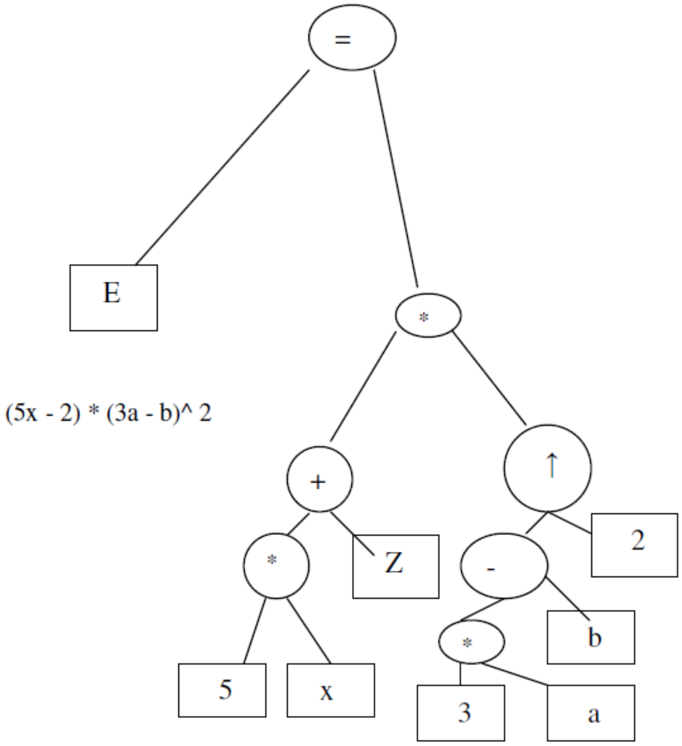
```
 #define MAX 100
Typedef struct
{ int top;
int elements [MAX]
} stack;
```

……………………………………………………………………………………..

Q13). Consider the algebraic expression **$E = (5x+z)\ (3a-b)^2$,** Draw the expression tree corresponding to E.

Ans.

E = (5x+z) $(3a\text{-}b)^2$


=
E
*
(5x - 2) * (3a - b)^ 2
+
↑
*
Z
-
2
5
x
*
b
3
a