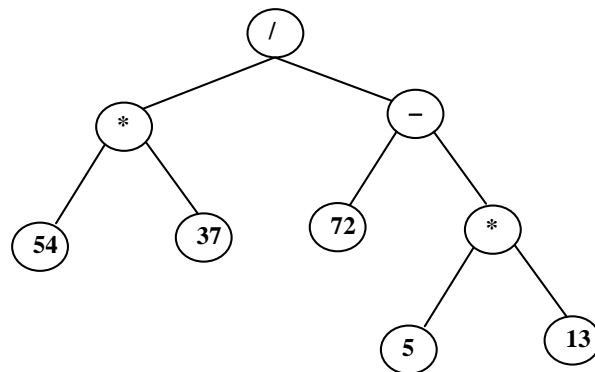1. A binary tree can store an arithmetic expression such that the parent of two sibling nodes is an operation and the two children of a node are operands of the operation of their parent node. For example, the following tree represents "$((54 + 37)/(72 - (5 * 13)))$".

Write a program that reads in an arithmetic expression, constructs a binary tree that represents the input expression, evaluates an arithmetic expression that consists of four binary operations (i.e. +, −, *, /) on integers and prints out the evaluation result. You may assume that the input expression is fully parenthesized. When there is a division by zero, your program should report it as an error.

2. Consider an ordered binary tree in which words are stored in an alphabetical order, i.e. words that are alphabetically less than a word are stored in the left subtree of the node for the word and words that are alphabetically greater than a word are stored in the right subtree of the node for the word. Here is an example.
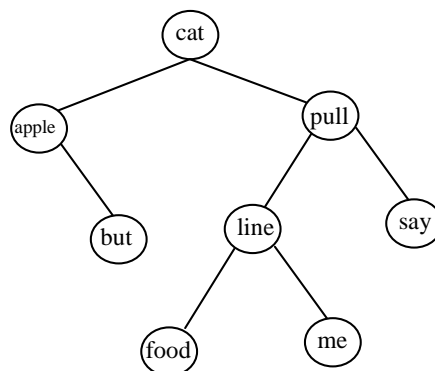
We would like to store such binary trees of words in a file as a sequence of words and restore it whenever it is necessary to perform adding, removing and finding words in it. In order to do that, it is necessary to traverse the tree to visit all the nodes and store the word of the visited node one by one in a file.

    (a)   Explain your algorithm of traversing the tree

    (b)   Write a program that reads the words in a file that was created as the result of the traversal of (a), restores the stored binary tree of words and prints out the tree with the breadth-first order. For example, for the binary tree above the following should be printed:

                cat

                apple pull

                but line say

                food me

3. Represent the following graph by

    (a) an adjacency matrix giving edge costs

    (b) a linked adjacency list with edge costs indicated
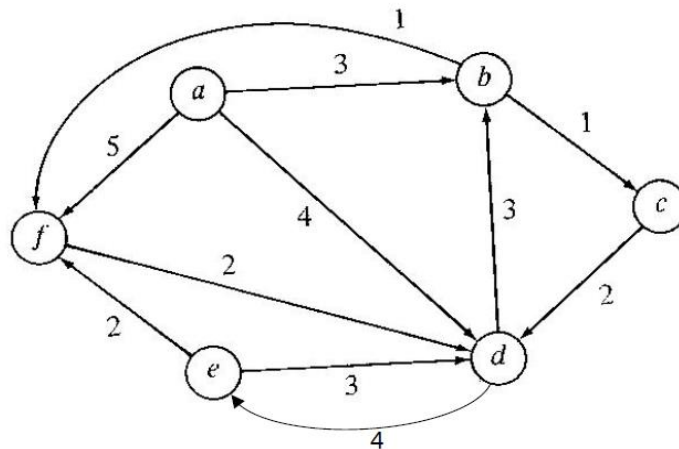
4. Show a minimum cost spanning tree obtained by applying each of the following algorithms to the graph in Problem 3:

    (a) the Prim's algorithm (Assume that a is the start node.)

    (b) the Kruskal's algorithm

    In (a) and (b), you should show the progress of the algorithms by drawing graphs step by step.

5. Show the adjacency matrix representation of the transitive closure of the graph in Problem 3.