1. *Design a binary tree and then write an algorithm to print the least(nearest) two common parent(if 2 parents exist otherwise 1  common parent) node between 2 nodes of a binary tree for given 2 key values which are present in binary tree.*

***Input****:  If below tree and values 1 and 5 given*

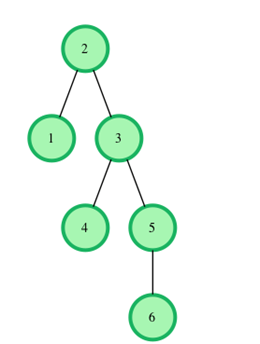
***Output****: 2*

***Input****:  If below tree and values 3 and 6 given*

***Output****: 2*

***Input****:  If below tree and values 4 and 6 given*

***Output****: 2 and 3*



[**URL:https://ide.geeksforgeeks.org/IwKHeOVLKU**](URL:https://ide.geeksforgeeks.org/IwKHeOVLKU)

1. *Design a system (Python/Java) and write the  code  which can download hundreds  of csv files and parse and put them in Database(You can choose mySQL or MongoDB etc). You have been given only one Machine with Linux ( with Multi-core CPU – 8 vCPU)  installed. File should be downloaded from remote server at some IP using SFTP.  And remote server****allows maximum only one SFTP connection at a time****from any remote machine(One IP only One SFTP  connection allowed).*

***The format of data  in CSV File is given below. And like below there will be thousands of different files. Cell Id and Timestamp would be unique for each row.  Each file name is also unique as illustrated below.***

*Files will have Names****Perf-1-2016-10-21 13:45.csv or  Perf-1-2016-10-21 13:30.csv  i.e. the format of file is   <FileNo>-TimeStamp.csv***

***A Sample File csv is pasted below.  There may be thousands of files like this.***

*Result Time,      Granularity Period,   Object Name,                                                           Cell ID,       CallAttemps*

*2016-10-21 13:45,15,"LIMRNC03/BSC6900UCell:Label=LHU30047c1\_Mandingo”, 30047,            20*

*2016-10-21 13:45,15,"LIMRNC03/BSC6900UCell:Label=LHU29267c1\_Las\_Moras”, 99267,           30*

*2016-10-21             3:45,15,"LIMRNC03/BSC6900UCell:Label=LHU29277c1\_Huanuco\_Centro”,29277, 40*

**URL:** [**https://ide.geeksforgeeks.org/EpgRRRM3SK**](https://ide.geeksforgeeks.org/EpgRRRM3SK)

1. *Design a  binary tree and then print the breadth first order traversal of this tree. In breadth first order traversal, we visit the nodes of same height first then go to nodes of next height or level.*

*1*

*\*

*2*

*\*

*5*

*/  \*

*3    6*

*\*

*4*

*For the above tree, the level order traversal is 1 -> 2 -> 5 -> 3 -> 6 -> 4.*

**URL: https://ide.geeksforgeeks.org/P7BopHUA3t**

1. *Encoding a string the below format to reduce the size of the string.*

*A string of lowercase characters in range ascii[‘a’..’z’]. We want to reduce the string to its shortest length by doing a series of operations. In each operation we select a pair of adjacent lowercase letters that match, and delete them. For instance, the string aab could be shortened to b in one operation. Now we have to delete as many characters as possible using this method and print the resulting string. If the final string is empty, print Empty String*

***Function Description***

*Complete the MaxReducedString function. It should return the super reduced string or Empty String if the final string is empty.*

*superReducedString has the following parameter(s):*

* *s: a string to reduce*

***Output Format***

*If the final string is empty, print Empty String; otherwise, print the final non-reducible string.*

***Input     Output***

*aaabccddd will get reduced to→ abccddd  will get reduced to → abddd  will get reduced to → abd*

**URL:** [**https://ide.geeksforgeeks.org/UkPT76Cdgx**](https://ide.geeksforgeeks.org/UkPT76Cdgx)