**Iterative Deepening Search**

**Aim:** Write a program to implement iterative deepening search.

**Code:**

graph = {

'a' : ['b', 'c', 'e'],

'b' : ['d', 'f'],

'c' : ['g','a'],

'e' : ['f'],

'f' : ['e'],

}

def IDDFS(root, goal):

depth = 0

while True:

print ("LOOPING AT DEPTH %i " % (depth))

result = DLS(root, goal, depth)

print ("RESULT: %s, GOAL: %s" % (result, goal))

if result == goal:

return result

depth = depth +1

def DLS(node, goal, depth):

print ("NODE: %s, GOAL %s, DEPTH: %i" % (node, goal, depth))

if depth == 0 and node == goal:

print( "GOAL FOUND ,RETURN TO")

return node

elif depth > 0:

print ("LOOPING THROUGH CHILD NODES: %s" % (graph.get(node, [])))

for child in graph.get(node, []):

if goal == DLS(child, goal, depth-1):

return goal

IDDFS('a', 'g')

**Output:**

