DEPTH FIRST GEARCH Ex. NO.02 DATE: To implement Depth first deard to traverse a graph and explore all vertices by visiting as fal along each branch as possible before backtracking ALGORITHM: Step 1: Start step 2: Initialize an empty stack x a list to keep track of visited Step 3: Push the starting node onto stack x mark visits. step 4: While the stack is not empty, repeat step 5 to step 7. Step 5: Pop the top node from the stack. stack. step 6: Print / process the popped step 7: for each adjacent unvisited neighbour of the popped node. Step 8: Mark neighbour as visited.

step 9: Push the unvisited neighborn onto the stack. step 10: Repeat until all reachable node are visited. step 11: stop. ROURAM: published (Jan 1000) 290 def dfs (graph, start): (1384)A Stack = [Start] Visited = set () while Stack: node=stack.pop() if node not in vivited: print (node, end:"") visited. add (mode) for neighbour in graph [node]: y neighbour not in visited: Stack append (neighborn) graph = { A': [B' C'], (B': ['D', 'E'], `C':['F'], \p':[], 'E': ['f'], (F':[] }

No Dim.

print ("DFS Traversal starting from node 'A':") in and interpret ofs (graph, 'A') · fair in year. OUTPUT: DFS Traversal starting from node 4':
ACFBED Charle Marke () too = bodishing Audithe stack. apoling of the specific the land of the dear ly. (" nelevis streve) decimant. Waster Landson ing the service of the service of y may home met in in still app or the rgriforitil. 11911:50

Min I . Val