ST.XAVIER’S COLLEGE

(Affiliated to Tribhuvan University)

Maitighar, Kathmandu



**AI LAB ASSIGNMENT #4 and #5**

**SUBMITTED BY:**

Rahul Raj Dahal

017BSCIT049

4th Sem, 2nd Year

**SUBMITTED TO:**

Mr. Balkrishna Subedi

Lecturer, Dept. of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**1) Write a program to implement BFS and DFS.**

**Source Code**

# -\*- coding: utf-8 -\*-

"""DFS/BFS.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1U\_tAGnYrbJk-REWBryxFXr5LZ7OnmZO7

"""

class Stack:

def \_\_init\_\_(self):

self.items=[]

def push(self,x):

self.items.append(x)

def pop(self):

return self.items.pop()

class queue:

def \_\_init\_\_(self):

self.items=[]

self.front=0

self.rear=0

def push(self,x):

self.items.append(x)

self.rear=self.rear+1

def pop(self):

temp= self.items[self.front]

self.front=self.front+1

return temp

def display(self):

for i in range(self.front,self.rear):

print(self.items)

graph={'A':['C','B'],'B':['E','D'],'D':None,'E':None,'C':['F','G'],'F':None,'G':None}

#depth for search

def dfs():

s=Stack()

s.push('A')

temp='A'

while(temp!=None):

try:

temp=s.pop()

print(temp)

except:

print("The queue/list is empty")

break

if(temp=='G'):

print("Goal node found")

break

if(graph[temp]==None):

pass

else:

for i in graph[temp]:

if(i!=None):

s.push(i)

#breadth first search

def bfs():

q=queue()

q.push('A')

temp='A'

while(temp!=None):

try:

temp=q.pop()

print(temp)

except:

print("Goal node not found")

break

if(temp=='G'):

print("Goal node found")

break

if(graph[temp]==None):

pass

else:

for i in graph[temp]:

if(i!=None):

q.push(i)

choice=input("Which Operation Do You Want To Do BFS/DFS. ").lower()

if(choice=='dfs'):

dfs()

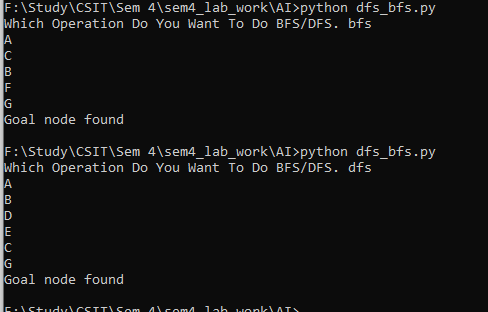
elif(choice=='bfs'):

bfs()

else:

print("Invalid Input")

**Screenshots**

****