Atma Ram Sanatan Dharma College

Class Assignment

Practical File

Question 8

SUBMITTED BY

Name : Ankit Sarawag

Course : Bsc.(Hons) Computer Science

Roll no : 22/28006

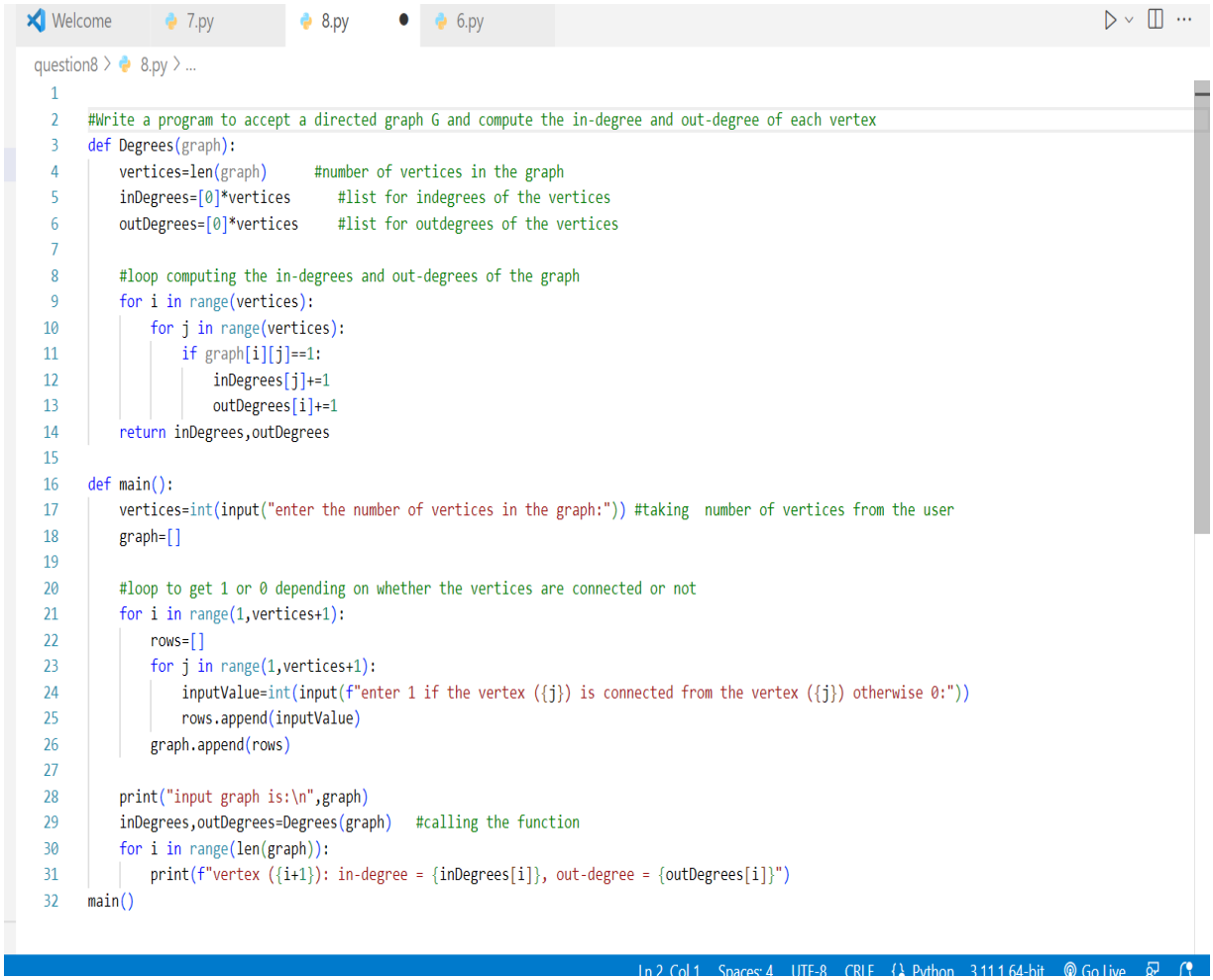
Semester : 2

Subject : Discrete Mathematical Structures

Teacher : Dr. Shalini Gupta(Faculty Of Computer Science)

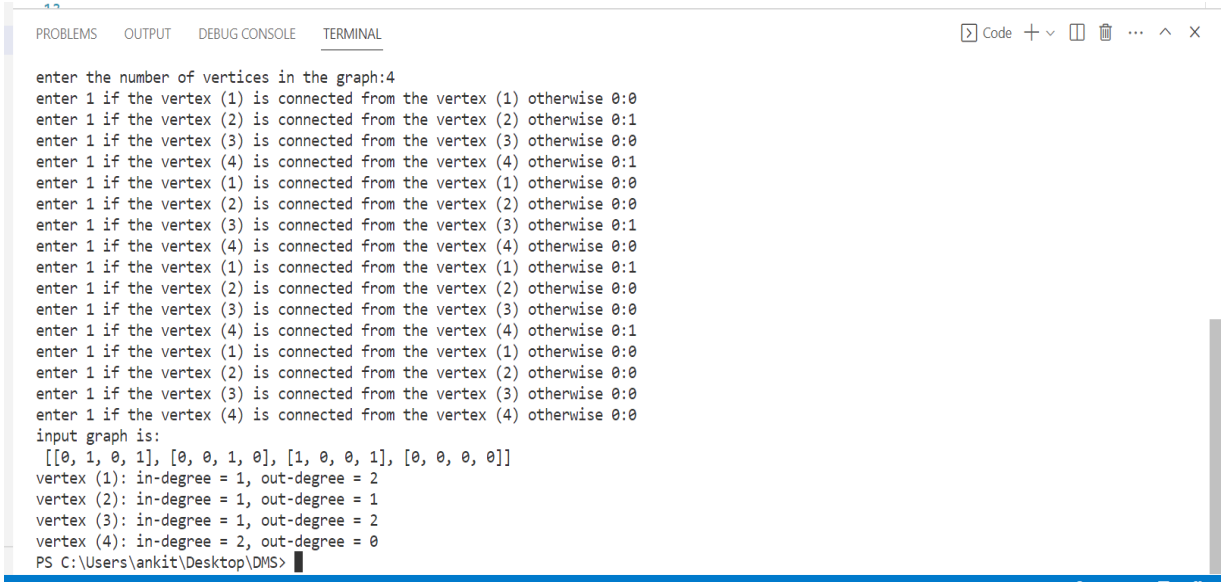
8) Write a program to accept a directed graph G and compute the in-degree and out-degree of each vertex.

CODE



```python

#Write a program to accept a directed graph G and compute the in-degree and out-degree of each vertex
def Degrees(graph):
    vertices=len(graph)      #number of vertices in the graph
    inDegrees=[0]*vertices      #list for indegrees of the vertices
    outDegrees=[0]*vertices     #list for outdegrees of the vertices

    #loop computing the in-degrees and out-degrees of the graph
    for i in range(vertices):
        for j in range(vertices):
            if graph[i][j]==1:
                inDegrees[j]+=1
                outDegrees[i]+=1
    return inDegrees,outDegrees

def main():
    vertices=int(input("enter the number of vertices in the graph:")) #taking  number of vertices from the user
    graph=[]

    #loop to get 1 or 0 depending on whether the vertices are connected or not
    for i in range(1,vertices+1):
        rows=[]
        for j in range(1,vertices+1):
            inputValue=int(input(f"enter 1 if the vertex ({j}) is connected from the vertex ({j}) otherwise 0:"))
            rows.append(inputValue)
        graph.append(rows)

    print("input graph is:\n",graph)
    inDegrees,outDegrees=Degrees(graph)   #calling the function
    for i in range(len(graph)):
        print(f"vertex ({i+1}): in-degree = {inDegrees[i]}, out-degree = {outDegrees[i]}")
main()
```

Output



```
enter the number of vertices in the graph:4
enter 1 if the vertex (1) is connected from the vertex (1) otherwise 0:0
enter 1 if the vertex (2) is connected from the vertex (2) otherwise 0:1
enter 1 if the vertex (3) is connected from the vertex (3) otherwise 0:0
enter 1 if the vertex (4) is connected from the vertex (4) otherwise 0:1
enter 1 if the vertex (1) is connected from the vertex (1) otherwise 0:0
enter 1 if the vertex (2) is connected from the vertex (2) otherwise 0:0
enter 1 if the vertex (3) is connected from the vertex (3) otherwise 0:1
enter 1 if the vertex (4) is connected from the vertex (4) otherwise 0:0
enter 1 if the vertex (1) is connected from the vertex (1) otherwise 0:1
enter 1 if the vertex (2) is connected from the vertex (2) otherwise 0:0
enter 1 if the vertex (3) is connected from the vertex (3) otherwise 0:0
enter 1 if the vertex (4) is connected from the vertex (4) otherwise 0:1
enter 1 if the vertex (1) is connected from the vertex (1) otherwise 0:0
enter 1 if the vertex (2) is connected from the vertex (2) otherwise 0:0
enter 1 if the vertex (3) is connected from the vertex (3) otherwise 0:0
enter 1 if the vertex (4) is connected from the vertex (4) otherwise 0:0
input graph is:
 [[0, 1, 0, 1], [0, 0, 1, 0], [1, 0, 0, 1], [0, 0, 0, 0]]
vertex (1): in-degree = 1, out-degree = 2
vertex (2): in-degree = 1, out-degree = 1
vertex (3): in-degree = 1, out-degree = 2
vertex (4): in-degree = 2, out-degree = 0
PS C:\Users\ankit\Desktop\DMS>
```