Assignment #3

Write a program to implement the Floyd-Warshall Algorithm to find All-Pair Shortest path. You must maintain two arrays

Date of Performance: 17/06/2023

Date of Submission: 10/06/2023

Student ID: 20210104108

fName: Sharjil Shabab Khan

Group: C1

```
#include <bits/stdc++.h>
using namespace std;
int n, e;
void inti(vector<vector<int>> dis, vector<vector<int>> &pi)
{
   for (int k = 0; k < n; k++)
   {
      for (int j = 0; j < n; j++)
      {
         if (dis[k][j] == INT_MAX || dis[k][j] == 0)
            pi[k][j] = -1;
         else
            pi[k][j] = k;
      }
   }
}

int main()
{
   cout << "Enter the numbers of nodes and edges" << endl;
   cin >> n >> e;

   vector<vector<int>> d
      {
         {0, 3, 8, INT_MAX, -4},
         {INT_MAX, 0, INT_MAX, 1, 7},
         {INT_MAX, 4, 0, INT_MAX, INT_MAX},
         {2, INT_MAX, -5, 0, INT_MAX},
         {INT_MAX, INT_MAX, INT_MAX, 6, 0}};

   vector<vector<vector<int>>> dis(n + 1, d);
   vector<vector<vector<int>>> pi(n + 1, d);

   dis[0] = d;
   inti(dis[0], pi[0]);

   for (int k = 1; k <= n; k++)
   {
      dis[k] = dis[k - 1];
      pi[k] = pi[k - 1];
      for (int i = 0; i < n; i++)
      {
         for (int j = 0; j < n; j++)
         {
            if (dis[k - 1][i][k - 1] != INT_MAX && dis[k - 1][k - 1][j] != INT_MAX)
               if ((dis[k - 1][i][k - 1] + dis[k - 1][k - 1][j]) < dis[k - 1][i][j])
```

```
                {
                    dis[k][i][j] = dis[k - 1][i][k - 1] + dis[k - 1][k - 1][j];
                    pi[k][i][j] = pi[k - 1][k - 1][j];
                }
            }
        }
    }

    for (int k = 1; k <= n; k++)
    {
        cout << "D" << k << endl;
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if (dis[k][i][j] == INT_MAX)
                {
                    cout << "I"<< " ";
                }
                else
                {
                    cout << dis[k][i][j] << " ";
                }
            }
            cout << endl;
        }
        cout << endl;
    }

    for (int k = 1; k <= n; k++)
    {
        cout << "pi" << k << endl;
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                if (pi[k][i][j] == -1)
                    cout << "NIL"<< "\t";

                else
                    cout << pi[k][i][j] + 1 << "\t";
            }
            cout << endl;
        }
    }
}
```