

<pre>#include<bits/stdc++.h> using namespace std; #define ll long long int #define MAX 100000000 int main() { int n; cout<<"Enter the number of node : "; cin>>n; ll ara[n][n]; for(int i=0; i<n; i++) { for(int j=0; j<n; j++) { cin>>ara[i][j]; if(ara[i][j]==0) { ara[i][j]=MAX; } } } cout<<"Graph represented matrix : \n"; for(int i=0; i<n; i++) { cout<<"\t"; for(int j=0; j<n; j++) { if(ara[i][j]==MAX) cout<<"inf"<<" "; </pre>	<pre>else cout<<ara[i][j]<<" "; } cout<<endl; } for(int k=0; k<n; k++) { for(int i=0; i<n; i++) { for(int j=0; j<n; j++) { ara[i][j]=min(ara[i][k]+ara[k][j],ara[i][j]); } } } cout<<"Shortest Path: \n"; for(int i=0; i<n; i++) {cout<<"\t"; for(int j=0; j<n; j++) { if(ara[i][j]==MAX) cout<<"inf"<<" "; else cout<<ara[i][j]<<" "; } cout<<endl; } }</pre>
---	---

```
<global>
prime che
"D:\CSE\Algorithm_Lab\A_N_S_30_383\Floyd-Warshall algorithm.exe"
Enter the number of node : 3
1 2 0
3 0 9
4 3 6
Graph represented matrix :
1 2 inf
3 inf 9
inf 3 6
Shortest Path:
1 2 11
3 5 9
6 3 6
Process returned 0 (0x0)   execution time : 13.479 s
Press any key to continue.
44
45
46
47
```

```

#include<bits/stdc++.h>
using namespace std;
#define ll long long int
#define inf 1000000000
vector<pair<ll, ll>> g[101];
ll v, e;
void floydWarshall()
{
    vector<vector<ll>> dist(v,
vector<ll>(v, inf));
    for (ll i = 0; i < v; i++)
    {
        dist[i][i] = 0;
        for (const pair<ll, ll>& edge
: g[i])
        {
            ll j = edge.first;
            ll weight = edge.second;
            dist[i][j] = weight;
        }
        for (ll k = 0; k < v; k++)
        {
            for (ll i = 0; i < v; i++)
            {
                for (ll j = 0; j < v;
j++)
                {
                    if (dist[i][k] != inf
&& dist[k][j] != inf)
                        dist[i][j] =
min(dist[i][j], dist[i][k] +
dist[k][j]);
                }
            }
        }
        cout<<"Distance From Source '0'
to every vertex : \n";
        cout<<"Vertex\tDistance From
Source\n";
        for(ll i=0; i<v; i++)
        {
            cout<<i<<"\t\t"<<dist[0][i]<<endl;
        }
    }
}

```

```

cout<<endl;
    cout<<"Matrix form of shortest
path : \n";
    for (ll i = 0; i < v; i++)
    {
        for (ll j = 0; j < v; j++)
        {
            if (dist[i][j] == inf)
                cout << "inf" << " ";
            else
                cout << dist[i][j] <<
" ";
        }
        cout << endl;
    }
}

int main()
{
    cout << "Enter the number of
vertices and edges : ";
    cin >> v >> e;

    for (ll i = 0; i < e; i++)
    {
        ll x, y, c;
        cin >> x >> y >> c;
        g[x].push_back({y, c});
    }

    floydWarshall();

    return 0;
}

```

```

#include<bits/stdc++.h>
using namespace std;
#define ll long long int
ll solve(ll var_Mn[],ll
mod_v[],ll i)
{
    ll x=1;
    while(true)
    {
        ll
rem=(var_Mn[i]*x)%mod_v[i];
        if(rem==1)
        {
            return x;
            break;
        }
        x++;
    }
}
int main()
{
    ll n;
    cout<<"Enter the number of
equation : ";
    cin>>n;
    ll
ara[n],mod_v[n],var_Mn[n],var_M
_in[n];
    for(int i=0; i<n; i++)
    {
        cout<<"Enter the value
of a"<<i+1<<" = ";
        cin>>ara[i];
    }

```

```

cout<<endl;
    for(int i=0; i<n; i++)
    {
        cout<<"Enter the
value of m"<<i+1<<" = ";
        cin>>mod_v[i];
    }
    ll M=1;
    for(int i=0; i<n; i++)
    {
        M*=mod_v[i];
    }
    for(int i=0; i<n; i++)
    {
        var_Mn[i]=(M/mod_v[i]);
    }
    for(int i=0;i<n;i++)
    {
        var_M_in[i]=solve(var_Mn,mod
_v,i);
    }
    ll X=0;
    for(int i=0;i<n;i++)
    {
        X+=ara[i]*var_Mn[i]*var_M_in
[i];
    }
    cout<<"\nValue of X = ";
    cout<<X%M<<endl;
}

```

```

"D:\CSE\Algorithm_Lab\A_N_S_30_383\Bellman-Ford algorithm.exe"
<global>
Enter the number of vertices and edges : 5 8
Warshal sor 0 1 -1
24 0 2 4
25 1 2 3
26 1 3 2
27 1 4 2
28 3 2 5
29 3 1 1
30 4 3 -3
31 Distance From Source '0' to every vertex :
32 Vertex Distance From Source
33 0 0
34 1 -1
35 2 2
36 3 -2
37 4 1
38 Matrix form of shortest path :
39 0 -1 2 -2 1
40 inf 0 3 -1 2
41 inf inf 0 inf inf
42 inf 1 4 0 3
43 inf -2 1 -3 0

```

```

}
M*=mod v[i].
}
for(
Enter the number of equation : 3
{
Enter the value of a1 = 2
Enter the value of a2 = 3
V Enter the value of a3 = 2
}
for(
Enter the value of m1 = 3
{
Enter the value of m2 = 5
Enter the value of m3 = 7
V
}
Value of X = 23
ll X
for(
Process returned 0 (0x0) execution time : 12.174 s
{
Press any key to continue.
X
}

```

