Instructions: Please read carefully

- Please rename this file as only your ID number **(e.g. 18-*****-1 Homework.doc or 18-*****-1 Homework.pdf).**
- Submit the file before **23-04-2021** in the MS TEAMS Assignment section labeled **Homework.**

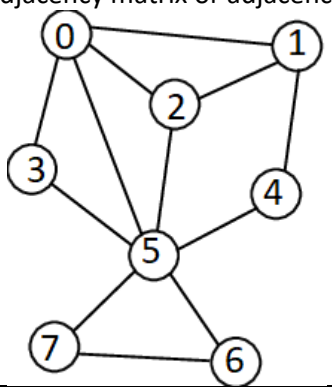
Do Not Copy!!!

Name:- Amit Podder

ID:- 20-42273-1

Section:- [F]

1. Represent the following graph with adjacency matrix or adjacency list.

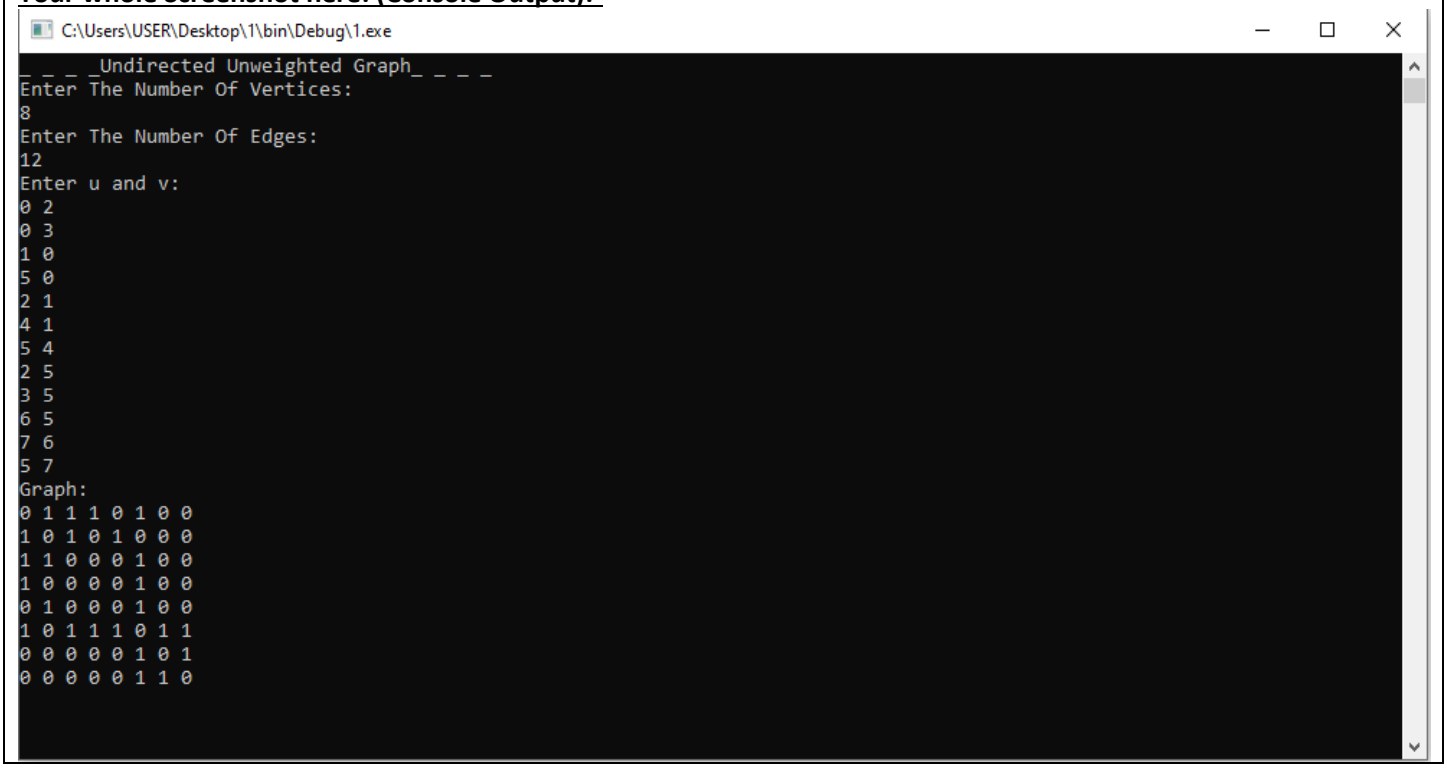


Your code here:-

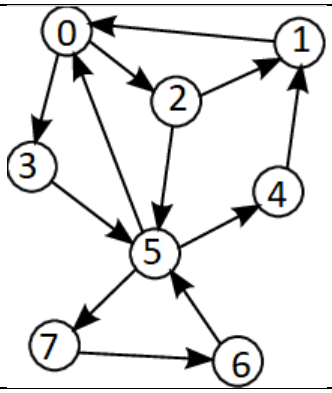
```
#include <iostream>

using namespace std;

int Graph[100][100];
int vertices, edges;
int u,v;
int i,j;

void InputGraph()
{
cout<<"Enter The Number Of Vertices:\n";
cin>>vertices;

cout<<"Enter The Number Of Edges:\n";
cin>>edges;

for(i=0;i<vertices; ++i)
for(j=0;j<vertices; ++j)
Graph[i][j] = 0;

cout<<"Enter u and v:\n";
for(i=0;i<edges; ++i)
{
cin>>u;
cin>>v;

Graph[u][v]=Graph[v][u] = 1;
}
```

```
}

void PrintGraph()
{
cout<<"Graph:\n";
for(i=0;i<vertices; ++i)
{
for(j=0;j<vertices; ++j)

cout<<Graph[i][j]<<" ";
cout<<endl;
}
cout<< "\n" <<endl;
}

int main()
{
cout<<"_ _ _ _Undirected Unweighted Graph_ _ _ _\n";

InputGraph();
PrintGraph();

return 0;
}
```

Your whole Screenshot here: (Console Output):-



2. Represent the following graph with adjacency matrix or adjacency list.

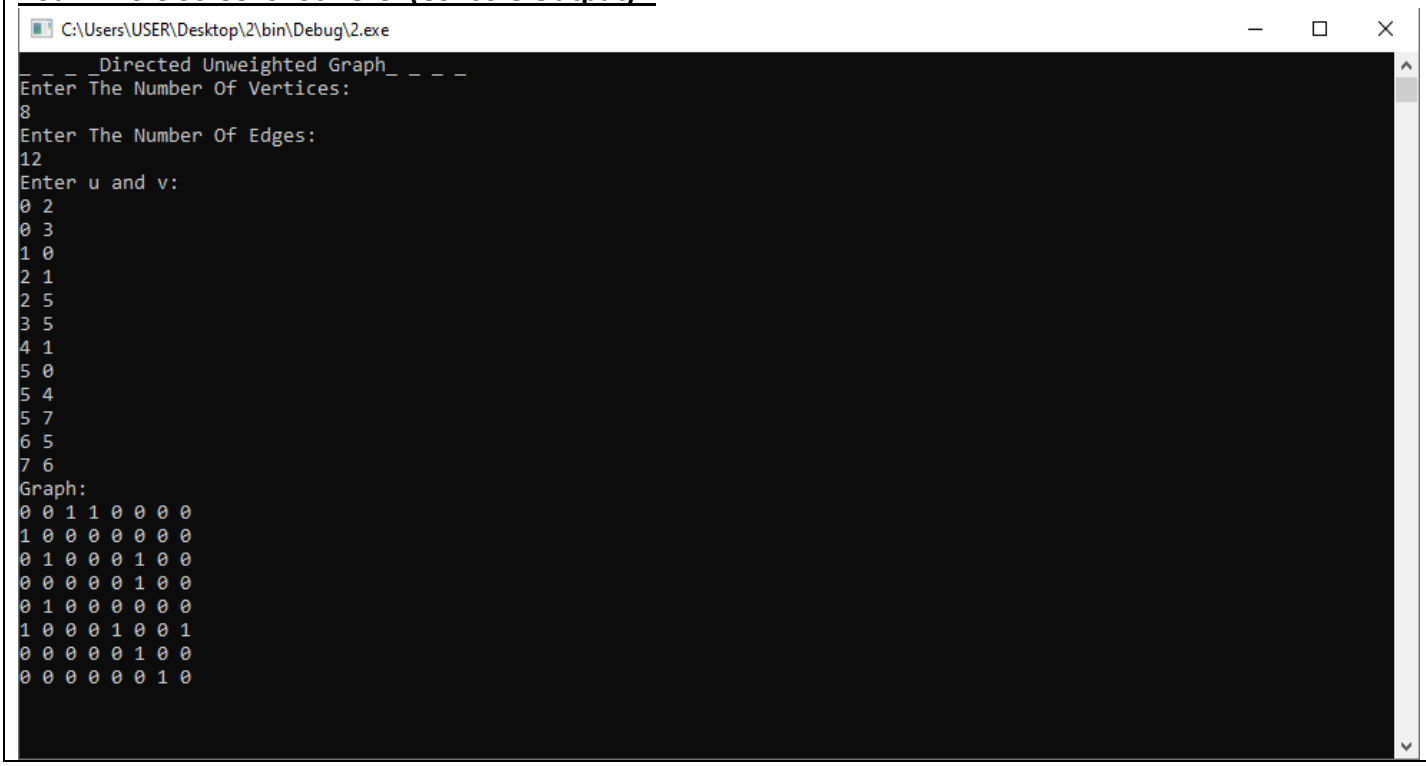


Your code here:-

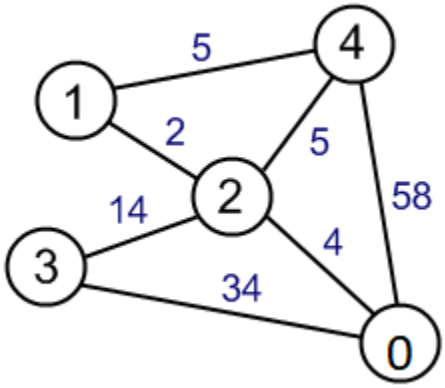
```
#include <iostream>

using namespace std;

int Graph[100][100];
int vertices, edges;
int u,v;
int i,j;

void InputGraph()
{
cout<<"Enter The Number Of Vertices:\n";
cin>>vertices;

cout<<"Enter The Number Of Edges:\n";
cin>>edges;

for(i=0;i<vertices; ++i)
for(j=0;j<vertices; ++j)
Graph[i][j]=0;

cout<<"Enter u and v:\n";
for(i=0;i<edges; ++i)
{
cin>>u;
cin>>v;

Graph[u][v]=1;
Graph[v][u]=0;
}
}

void PrintGraph()
{
cout<<"Graph:\n";
for(i=0;i<vertices; ++i)
{
for(j=0;j<vertices; ++j)

cout<<Graph[i][j]<<" ";
cout<<endl;
```

```
}
cout<< "\n" <<endl;
}

int main()
{
cout<<"_ _ _ _Directed Unweighted Graph_ _ _ _\n";

InputGraph();
PrintGraph();

return 0;
}
```

Your whole Screenshot here: (Console Output):-



3. Represent the following graph with adjacency matrix or adjacency list.



Your code here:-

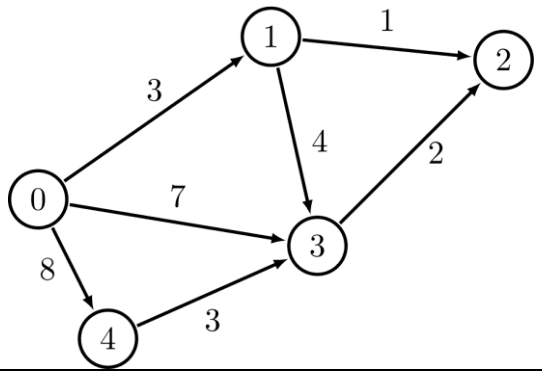
```
#include <iostream>

using namespace std;

int Graph[100][100];
int vertices, edges, weights;
int u,v,w;
int i,j;

void InputGraph()
{
cout<<"Enter The Number Of Vertices:\n";
cin>>vertices;

cout<<"Enter The Number Of Edges:\n";
cin>>edges;

cout<<"Enter The Number Of Weights:\n";
cin>>weights;

for(i=0;i<vertices; ++i)
for(j=0;j<vertices; ++j)
Graph[i][j]=0;

cout<<"Enter u,v and w:\n";
for(i=0;i<edges; ++i)
{
cin>>u;
cin>>v;
cin>>w;

Graph[u][v]=Graph[v][u]= w;
}
}

void PrintGraph()
{
cout<<"Graph:\n";
for(i=0;i<vertices; ++i)
{
```

```
for(j=0;j<vertices; ++j)

cout<< Graph[i][j] <<"    ";
cout<<endl;
}
cout<< "\n" <<endl;
}

int main()
{
cout<<"_ _ _ _Undirected Weighted Graph_ _ _ _\n";

InputGraph();
PrintGraph();

return 0;
}
```

Your whole Screenshot here: (Console Output):-

```
C:\Users\USER\Desktop\3\main.exe
_ _ _ _Undirected Weighted Graph_ _ _ _
Enter The Number Of Vertices:
5
Enter The Number Of Edges:
7
Enter The Number Of Weights:
7
Enter u,v and w:
1 2 2
1 4 5
2 0 4
2 4 5
2 3 14
3 0 34
4 0 58
Graph:
0    0    4    34    58
0    0    2    0    5
4    2    0    14    5
34    0    14    0    0
58    5    5    0    0


Process returned 0 (0x0)   execution time : 63.122 s
Press any key to continue.
```

4. Represent the following graph with adjacency matrix or adjacency list.



Your code here:-

```
#include <iostream>

using namespace std;

int Graph[100][100];
int vertices, edges, weights;
int u,v,w;
int i,j;

void InputGraph()
{
cout<<"Enter The Number Of Vertices:\n";
cin>>vertices;

cout<<"Enter The Number Of Edges:\n";
cin>>edges;

cout<<"Enter The Number Of Weights:\n";
cin>>weights;

for(i=0;i<vertices; ++i)
for(j=0;j<vertices; ++j)
Graph[i][j]=0;

cout<<"Enter u,v and w:\n";
for(i=0;i<edges; ++i)
{
cin>>u;
cin>>v;
cin>>w;

Graph[u][v]=w;
Graph[v][u]=0;
}
}

void PrintGraph()
{
cout<<"Graph:\n";
for(i=0;i<vertices; ++i)
{
for(j=0;j<vertices; ++j)

cout<<Graph[i][j]<<" ";
cout<<endl;
}
cout<< "\n" <<endl;
}

int main()
{
cout<<"_ _ _ _Directed Weighted Graph_ _ _ _\n";
```

```
InputGraph();
PrintGraph();

return 0;
}
```

Your whole Screenshot here: (Console Output):-