Name : Hrithik Paul

Roll no : 20CS011017

Jisu Id: JISU/2020/0513

Assignment 20:

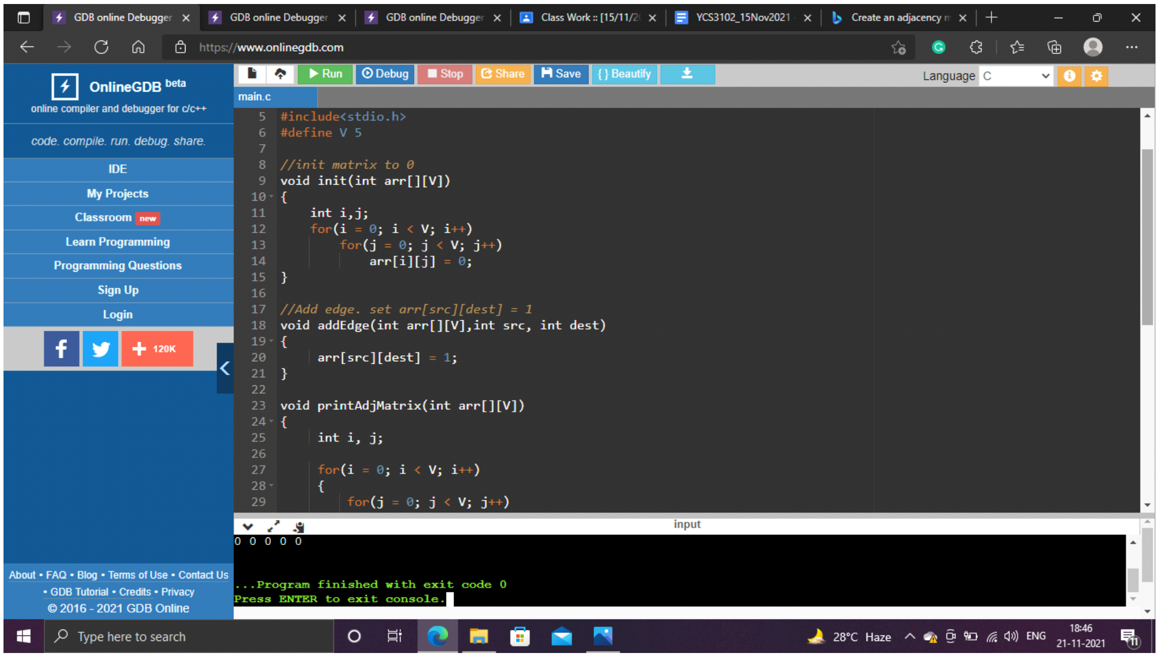
Create an adjacency matrix of a graph from the user given edge set

Algo:

Step 1:Create a matrix A of size NxN and initialise it with zero.

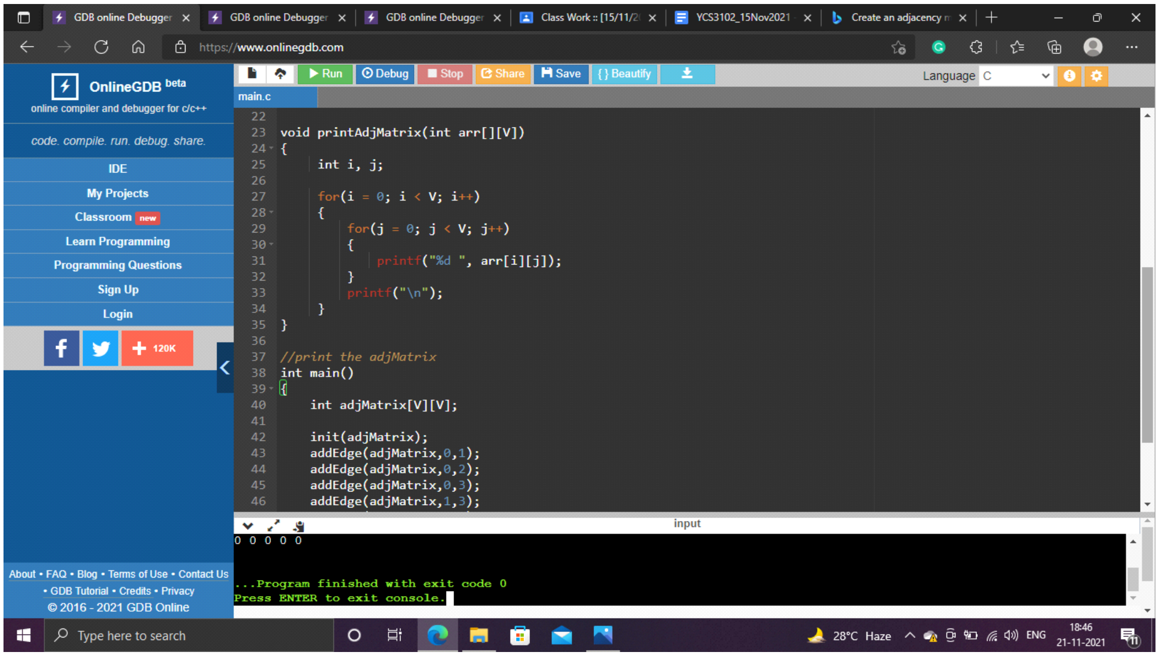
Step 2: Iterate over each given edge of the form (u,v) and assign 1 to A[u][v]. Also, If graph is undirected then assign 1 to A[v][u].

Output ss:



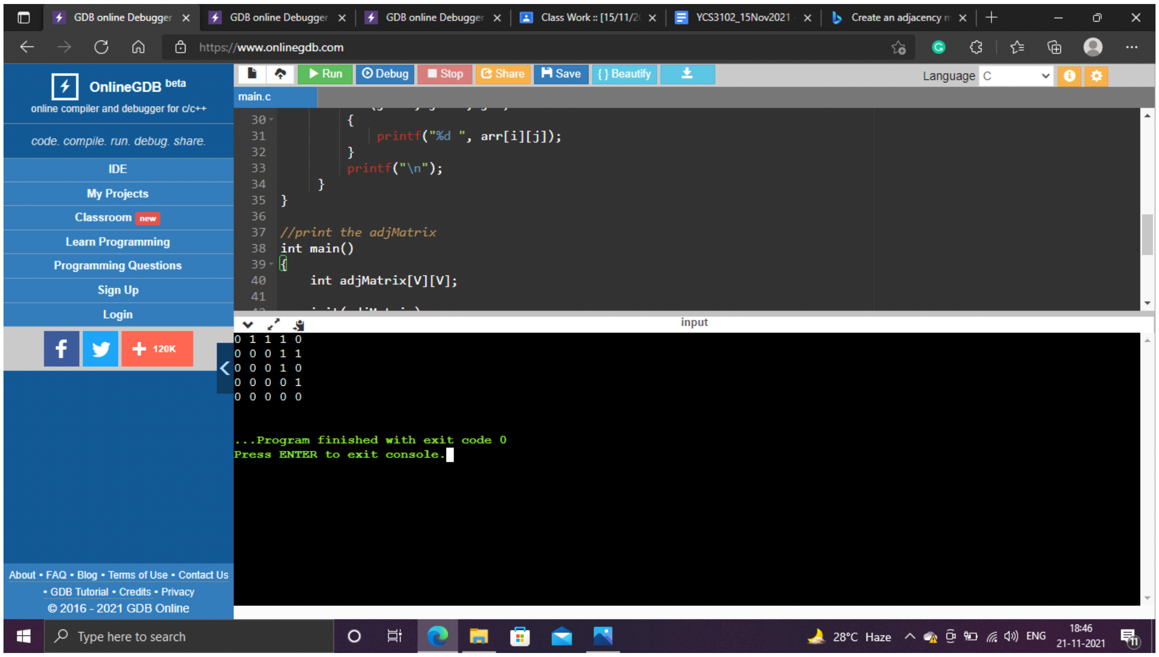
```
#include<stdio.h>
#define V 5

//init matrix to 0
void init(int arr[][V])
{
    int i,j;
    for(i = 0; i < V; i++)
        for(j = 0; j < V; j++)
            arr[i][j] = 0;
}

//Add edge. set arr[src][dest] = 1
void addEdge(int arr[][V],int src, int dest)
{
     arr[src][dest] = 1;
}

void printAdjMatrix(int arr[][V])
{
     int i, j;

     for(i = 0; i < V; i++)
     {
         for(j = 0; j < V; j++)
```

```
0 0 0 0 0


...Program finished with exit code 0
Press ENTER to exit console.
```



```
void printAdjMatrix(int arr[][V])
{
    int i, j;

    for(i = 0; i < V; i++)
    {
        for(j = 0; j < V; j++)
        {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
}

//print the adjMatrix
int main()
{
    int adjMatrix[V][V];

    init(adjMatrix);
    addEdge(adjMatrix,0,1);
    addEdge(adjMatrix,0,2);
    addEdge(adjMatrix,0,3);
    addEdge(adjMatrix,1,3);
```

```
0 0 0 0 0


...Program finished with exit code 0
Press ENTER to exit console.
```



```
        {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
}

//print the adjMatrix
int main()
{
    int adjMatrix[V][V];

```

```
0 1 1 1 0
0 0 0 1 1
0 0 0 1 0
0 0 0 0 1
0 0 0 0 0


...Program finished with exit code 0
Press ENTER to exit console.
```

Assignment 21:

Algo:Step 1: SET STATUS = 1 (ready state)

for each node in G

Step 2: Enqueue the starting node A

and set its STATUS = 2

(waiting state)

Step 3: Repeat Steps 4 and 5 until QUEUE is empty

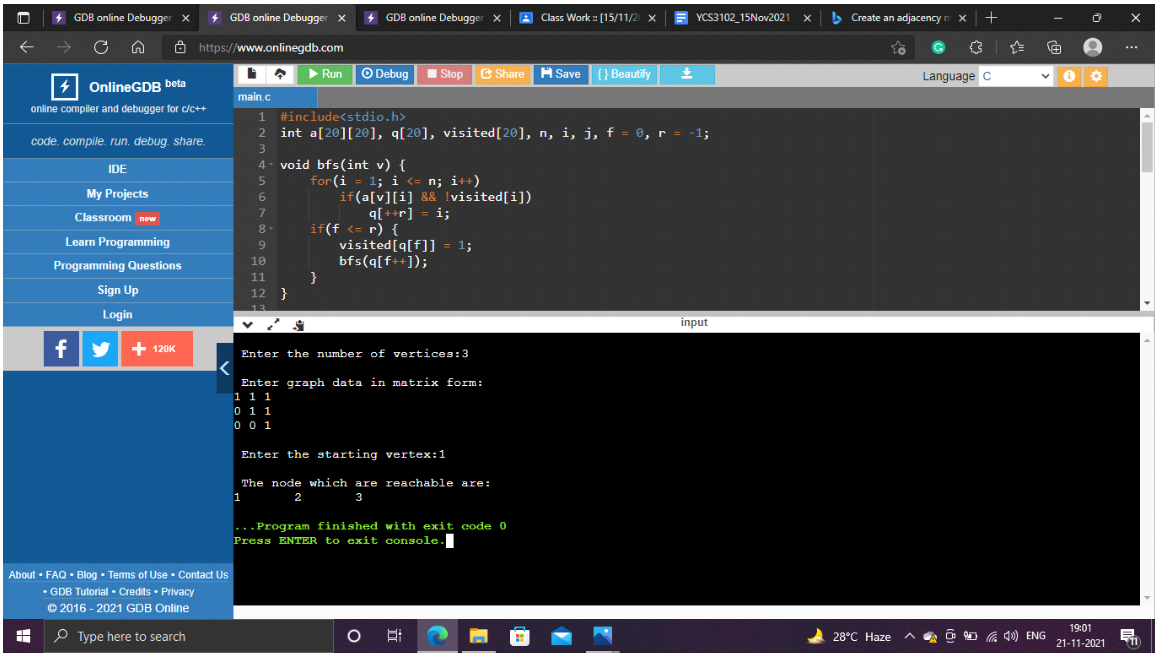
Step 4: Dequeue a node N. Process it and set its STATUS = 3 (processed state).

Step 5: Enqueue all the neighbours of N that are in the ready state (whose STATUS = 1) and set their STATUS = 2 (waiting state)

[END OF LOOP]

Step 6: EXIT

Output :



```
#include<stdio.h>
int a[20][20], q[20], visited[20], n, i, j, f = 0, r = -1;

void bfs(int v) {
    for(i = 1; i <= n; i++)
        if(a[v][i] && !visited[i])
            q[++r] = i;
    if(f <= r) {
        visited[q[f]] = 1;
        bfs(q[f++]);
    }
}
```

```
 Enter the number of vertices:3

 Enter graph data in matrix form:
1 1 1
0 1 1
0 0 1

 Enter the starting vertex:1

 The node which are reachable are:
1       2       3

...Program finished with exit code 0
Press ENTER to exit console.
```

Assignment 22:

Algo:

Step 1: SET STATUS = 1 (ready state) for each node in G

Step 2: Push the starting node A on the stack and set its STATUS = 2 (waiting state)

Step 3: Repeat Steps 4 and 5 until STACK is empty

Step 4: Pop the top node N. Process it and set its STATUS = 3 (processed state)

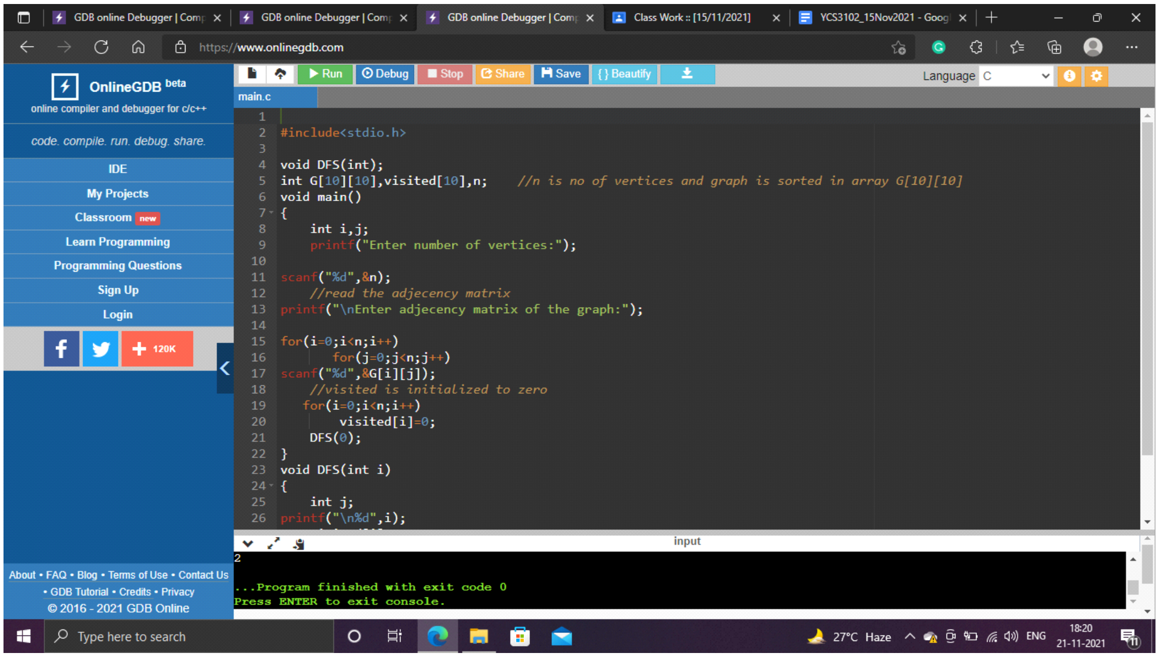
Step 5: Push on the stack all the neighbours of N that are in the ready state (whose STATUS = 1) and set their

STATUS = 2 (waiting state)

[END OF LOOP]

Step 6: EXIT

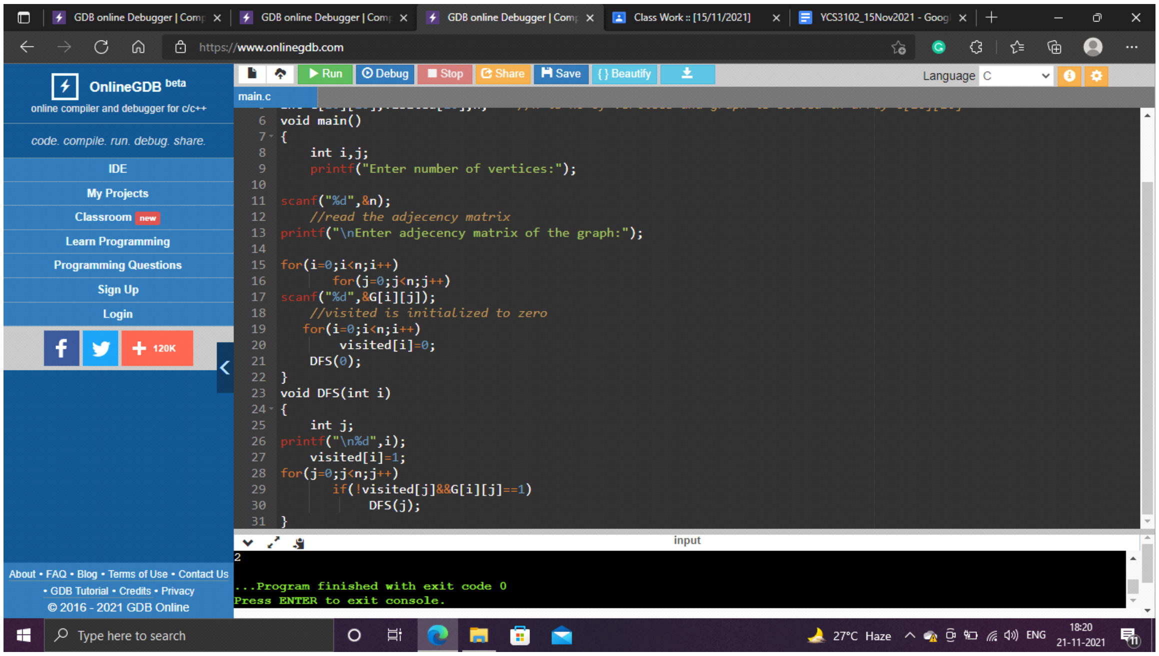
Output:



GDB online Debugger | Comp
GDB online Debugger | Comp
GDB online Debugger | Comp
Class Work :: [15/11/2021]
YCS3102_15Nov2021 - Goog
https://www.onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
120K
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
input
Type here to search
27°C Haze
ENG
18:20
21-11-2021

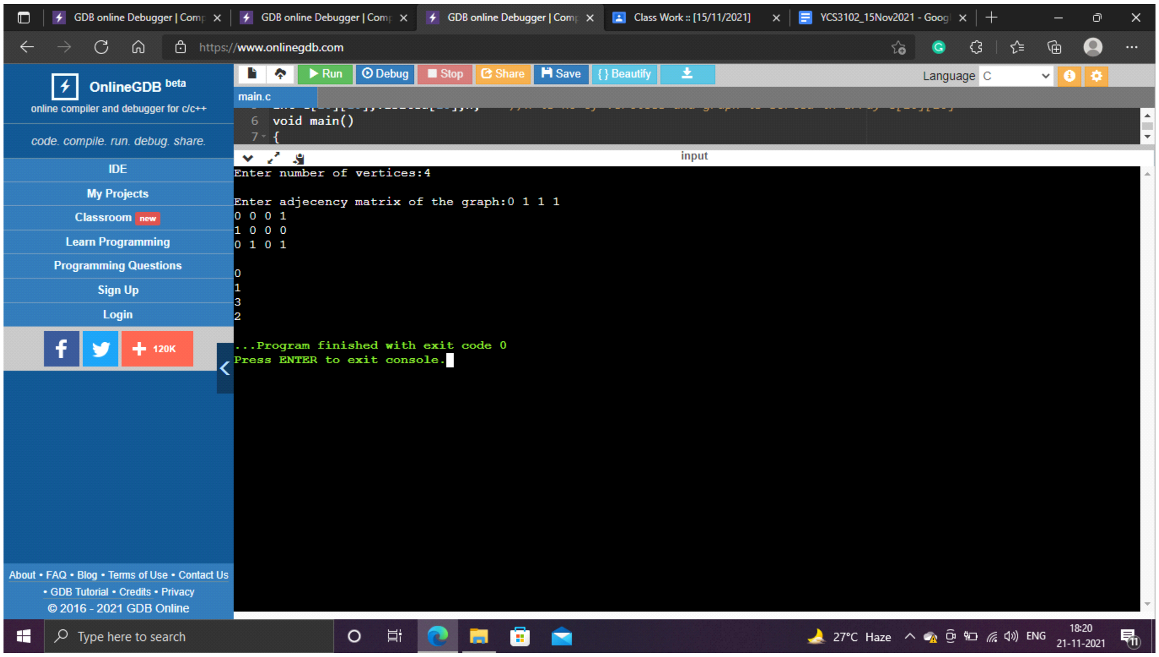
```

#include<stdio.h>

void DFS(int);
int G[10][10],visited[10],n;    //n is no of vertices and graph is sorted in array G[10][10]
void main()
{
    int i,j;
    printf("Enter number of vertices:");

scanf("%d",&n);
    //read the adjecency matrix
printf("\nEnter adjecency matrix of the graph:");

for(i=0;i<n;i++)
       for(j=0;j<n;j++)
scanf("%d",&G[i][j]);
    //visited is initialized to zero
   for(i=0;i<n;i++)
        visited[i]=0;
    DFS(0);
}
void DFS(int i)
{
    int j;
printf("\n%d",i);
```

```
2

...Program finished with exit code 0
Press ENTER to exit console.
```


GDB online Debugger | Comp
GDB online Debugger | Comp
GDB online Debugger | Comp
Class Work :: [15/11/2021]
YCS3102_15Nov2021 - Goog
https://www.onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
120K
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
input
Type here to search
27°C Haze
ENG
18:20
21-11-2021

```
void main()
{
    int i,j;
    printf("Enter number of vertices:");

scanf("%d",&n);
    //read the adjecency matrix
printf("\nEnter adjecency matrix of the graph:");

for(i=0;i<n;i++)
       for(j=0;j<n;j++)
scanf("%d",&G[i][j]);
    //visited is initialized to zero
   for(i=0;i<n;i++)
        visited[i]=0;
    DFS(0);
}
void DFS(int i)
{
    int j;
printf("\n%d",i);
    visited[i]=1;
for(j=0;j<n;j++)
       if(!visited[j]&&G[i][j]==1)
            DFS(j);
}
```

```
2

...Program finished with exit code 0
Press ENTER to exit console.
```


GDB online Debugger | Comp
GDB online Debugger | Comp
GDB online Debugger | Comp
Class Work :: [15/11/2021]
YCS3102_15Nov2021 - Goog
https://www.onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
120K
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
input
Type here to search
27°C Haze
ENG
18:20
21-11-2021

```
void main()
{
```

```
Enter number of vertices:4

Enter adjecency matrix of the graph:0 1 1 1
0 0 0 1
1 0 0 0
0 1 0 1

0
1
3
2


...Program finished with exit code 0
Press ENTER to exit console.
```