**CODE:**

#include <stdio.h>

#include <string.h>

#include <vector>

using namespace std;

const int N = 105, L = 50;

bool tag[N];

vector<int> adj\_list[N];

vector<int> component[N];

int component\_count;

void init(int node\_count);

void dfs(int u);

void print\_ans();

int main()

{

char fname[L];

int node\_count, edge\_count, u, v;

printf("Enter the name of the file: ");

scanf("%s", fname);

freopen(fname, "r", stdin);

scanf("%d%d", &edge\_count, &node\_count);

init(node\_count);

for(int i = 0; i < edge\_count; i++)

{

scanf("%d%d", &u, &v);

adj\_list[u].push\_back(v);

adj\_list[v].push\_back(u);

}

for(int i = 1; i <= node\_count; i++){

if(!tag[i]){

component\_count++;

tag[i] = true;

dfs(i);

}

}

print\_ans();

return 0;

}

void init(int node\_count){

for(int i = 1; i <= node\_count; i++){

adj\_list[i].clear();

component[i].clear();

}

memset(tag, 0, sizeof(tag));

component\_count = 0;

}

void dfs(int u){

component[component\_count - 1].push\_back(u);

int adj\_count = adj\_list[u].size();

for(int i = 0; i < adj\_count; i++)

{

int v = adj\_list[u][i];

if(!tag){

tag[v] = true;

dfs(v);

}

}

}

void print\_ans()

{

for(int i = 0; i < component\_count; i++){

int component\_size = component[i].size();

printf("Connected Component #%d:\n", i + 1);

for(int j = 0; j <component\_size; ++j){

if(j){

printf(" ");

}

printf("%d", component[i][j]);

}

printf("\n");

}

}

**OUTPUT:**

**File 1**

Connected Component #1:

1 2 4 8 5 10 9 6

Connected Component #2:

3

Connected Component #3:

7

**File 2**

Connected Component #1:

1 15 5 7 27 26 14 13 12 21 25 16 6 19 9 17 2 10 18 28 29 30 23 24 20 3

Connected Component #2:

4 8

Connected Component #3:

11

Connected Component #4:

22

Connected Component #5:

31 42 46 32 37 41 47 39 40 49

Connected Component #6:

33 48 45 44

Connected Component #7:

34 43 36

Connected Component #8:

35

Connected Component #9:

38

Connected Component #10:

50

**File 3**

Connected Component #1:

1 11 37 62 23 46 63 4 55 85 52 75 12 87 67 70 97 30 43 77 34 96 47 33 79

Connected Component #2:

2 10

Connected Component #3:

3 15 44 25 26 54 22 82 27 59 29 39 24 51 35 89 9 72 68 74 99 76 93 71 90 80 84 64 92 6

Connected Component #4:

5 21

Connected Component #5:

7

Connected Component #6:

8 73 40 61 16 83 100 19 45 48 57 81 50 32

Connected Component #7:

13

Connected Component #8:

14

Connected Component #9:

17

Connected Component #10:

18

Connected Component #11:

20

Connected Component #12:

28 60 42 65 53 66 88

Connected Component #13:

31

Connected Component #14:

36

Connected Component #15:

38

Connected Component #16:

41

Connected Component #17:

49

Connected Component #18:

56

Connected Component #19:

58

Connected Component #20:

69 98

Connected Component #21:

78

Connected Component #22:

86

Connected Component #23:

91

Connected Component #24:

94

Connected Component #25:

95

**File 4**

Connected Component #1:

1 4 7 8 9 12 13 14 16 20 22 23 25 26 30 32 34 38 40 41 42 44 45 46 48 49 50 52 5

3 54 55 56 59 62 65 66 67 70 73 75 77 81 82 87 88 89 90 91 93 100

Connected Component #2:

2 3 5 6 10 11 15 17 18 19 21 24 27 28 29 31 33 35 36 37 39 43 47 51 57 58 60 61

63 64 68 69 71 72 74 76 78 79 80 83 84 85 86 92 94 95 96 97 98 99