#include <stdio.h>

#define STACK\_SIZE 20

#define MAX 20

void dfs(int Adj[][MAX], int n, int source);

int main(void) {

//Adj matrix

int Adj[][MAX] = {

{0,1,0,0},

{0,1,1,1},

{1,0,0,1},

{0,0,1,0}

};

int n = 4; //no. of vertex

int starting\_vertex = 3;

dfs(Adj, n, starting\_vertex);

return 0;

}

void dfs(int Adj[][MAX], int n, int source) {

//variables

int i, j;

bool change = false;

//visited array to flag the vertex that

//were visited

int visited[MAX];

//top of the stack

int tos = 0;

//stack

int stack[STACK\_SIZE];

//set visited for all vertex to 0 (means unvisited)

for(i = 0; i < MAX; i++) {

visited[i] = 0;

}

//mark the visited source

visited[source] = 1;

//push the vertex into stack

stack[tos] = source;

//print the vertex as result

printf("%d ", source);

//continue till stack is not empty

while(tos >= 0) {

//to check if any vertex was marked as visited

change = false;

//get vertex at the top of the stack

i = stack[tos];

for(j = 0; j < n; j++) {

if(visited[j] == 0 && Adj[i][j] == 1) {

//mark vertex as visited

visited[j] = 1;

//push vertex into stack

tos++;

stack[tos] = j;

//print the vertex as result

printf("%d ", j);

//vertex visited

change = true;

break;

}

}

if(change == false) {

tos--;

}

}

printf("\n");

}