|  |  |
| --- | --- |
| **Experiment No.** | **9** |
| **Aim** | **Experiment based on Approximation Algorithms** |
| **Name** | **Rucha Sudhir Kulkarni** |
| **UID No.** | **2021300067** |
| **Class & Division** | **SE Computer Engineering (Div:A)(Batch:D)** |
| **Date of Performance:** | **26.04.2023** |
| **Date of Submission:** | **28.04.2023** |

**Aim: – To implement vertex cover problem.**

**Algorithm:**

1. **Start**
2. **Take input n as no. of vertices and m as no. of edges.**
3. **Input the vertices having connection i.e edges and store into graph[][] which is adjacency matrix of the graph.**
4. **Starting from vertex 0 consider each vertex one by one and find the edges it is connected to if it is unvisited i.e visited[i]==false.**
5. **Mark those adjacent vertices also as visited.**
6. **Print the vertices of visited array as answer.**
7. **Stop**

**Program:**

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#define MAX\_VERTICES 10

int graph[MAX\_VERTICES][MAX\_VERTICES];

bool visited[MAX\_VERTICES];

void add\_edge(int u, int v) {

    graph[u][v] = graph[v][u] = 1;

}

void print\_vertex\_cover(int n) {

    int i, j;

    for (i = 0; i < n; i++) {

        if (!visited[i]) {

            for (j = 0; j < n; j++) {

                if (graph[i][j] && !visited[j]) {

                    visited[i] = visited[j] = true;

                    break;

                }

            }

        }

    }

    for(int i=0;i<n;i++){

        if(i==0){

            printf("{");

        }

        if(visited[i])

          printf(" %d",i);

         if(i==n-1){

            printf("}");

        }

    }

}

int main() {

    int n, m;

    int i, u, v;

    printf("Enter the number of vertices: ");

    scanf("%d", &n);

    printf("Enter the number of edges: ");

    scanf("%d", &m);

    for (i = 0; i < m; i++) {

        printf("Enter edge %d: ", i+1);

        scanf("%d %d", &u, &v);

        add\_edge(u, v);

    }

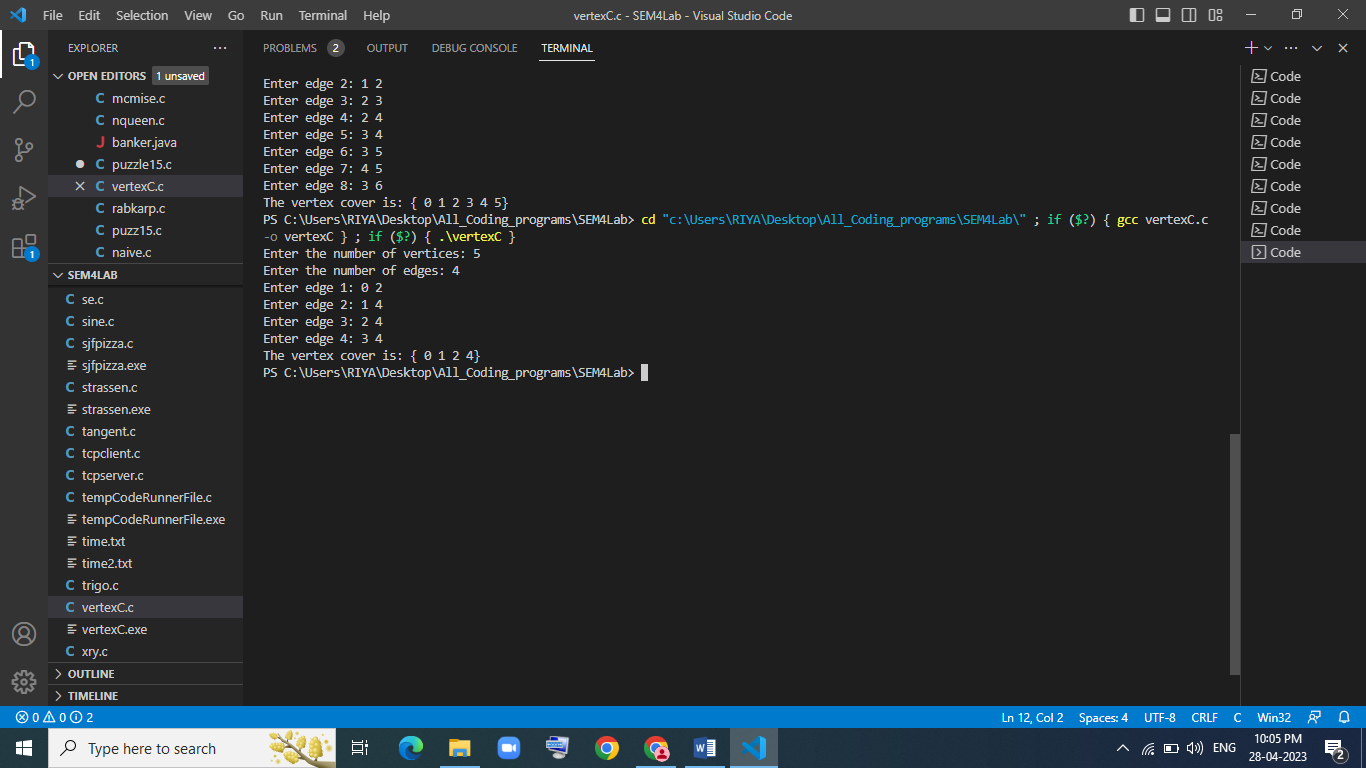
    printf("The vertex cover is: ");

    print\_vertex\_cover(n);

    return 0;

}

**Output and Observation:**



**Conclusion:**

After performing the above experiment, I got to know the vertex cover problem and how to implement it using an approximation algorithm.