

```

//
// EE878C High Level Synthesis Algorithm for CAD
//
// Homework1 : programming (Maximum Clique)
// Student ID : 20023404 Name : Lee, Dongsoo
//
// Input Format : 1. Number of Vertices
//                2. Edges (undirected) by two vertices
// Output Format : Maximum cliques (list of included vertices)
//

#include <stdio.h>

int edge_table[50][50]; // To save edges
int check_vertex[50]; // To test for each vertex
int n; // Number of total vertices

void max_clique(int vertex)
{
    // num : Number of selected vertices
    // num2 : To compare with num for checking maximum clique

    int i,j,num,flag2,num2;

    // Condition for end is when vertex is n
    if(vertex == n) {

        num = 0;
        // Count selected vertices
        for(i=0;i<n;i++)
            if (check_vertex[i]) num++;
        if (num>2) { // only over two vertices selected

            flag2 = 0;

            // check whether this is maximum clique
            // by test of finding another vertex which can be added
            for(i=0;i<n;i++) {
                if (!check_vertex[i]) { // test vertex
                    num2 = 0;
                    for(j=0;j<n;j++)
                        // count vertices which are connected with
                        // test vertex
                        if(check_vertex[j] && edge_table[i][j])
                            num2++;

                    // when 'num2' is 'num', test vertex can be added
                    // and this is not maximum clique
                    if (num2 == num) flag2 = 1;
                }
            }

            // if this is maximum clique then print the list
            if (!flag2) {
                for(i=0;i<n;i++)
                    if (check_vertex[i]) printf(" %d",i);
                printf("\n");
            }
        }
        return;
    }
}

```

```

// test by not selecting this vertex
check_vertex[vertex] = 0;
// test next vertex
max_clique(vertex + 1);

flag2 = 0;
// check whether this vertex can be added to current list('check_vertes')
for (i=0;i<vertex;i++)
    if (check_vertex[i] && !edge_table[i][vertex]) flag2 = 1;
// if all connected then test by selecting this vertex
if (!flag2) {
    check_vertex[vertex] = 1;
    // test next vertex
    max_clique(vertex + 1);
}
return;
}

main()
{
    int i,j;
    int flag,v1,v2;

    // initializing
    for(i=0;i<50;i++) {
        for(j=0;j<50;j++)
            edge_table[i][j] = 0;
        check_vertex[50]=0;
    }

    printf("\n Homework #1 : Maximum cliques\n");
    printf(" 20023404 Lee,Dongsoo\n\n");
    printf(" Number of Vertices(Max=50): ");
    scanf("%d",&n);
    printf("\n Vertex is from 0 to %d\n",n-1);
    printf("\n Input Undirected Edges [vertex vertex], end -> 99 99\n\n");

    flag = 0;
    while(flag == 0) {
        scanf("%d %d",&v1,&v2);
        if (v1 == 99 && v2 == 99) flag = 1;           // input ending condition
        else {
            // save edges with table
            edge_table[v1][v2] = 1;
            edge_table[v2][v1] = 1;
        }
    }

    printf("\n < All Maximum Cliques >\n\n");

    max_clique(0);
    printf("\n End of Program....\n");
}

```