

# ASSIGNMENT-7

```
① #include <iostream>
using namespace std;
void readGraph (int **graph, int n) {
    cout << "Enter matrix = ";
    for (int i=0; i<n; i++) {
        for (int graph[i] = new int[n];
        for (int j=0; j<n; j++) {
            cin >> graph[i][j];
        }
    }
}

bool isSafe (int **graph, int n, int *color, int n, int c) {
    for (int i=0; i<n; i++) {
        if (graph[i][i] && c == color[i])
            return false;
    }
    return true;
}

bool graphColouringUtil (int **graph, int n, int *color, int v) {
    if (v == n)
        return true;
    for (int c=0; c<3; c++) {
        if (isSafe (graph, n, color, v, c)) {
            color[v] = c;
            if (graphColouringUtil (graph, n, color, v+1)) {
                return true;
            }
            color[v] = -1;
        }
    }
    return false;
}
```

```
void graphColouring(int **graph, int n){
```

```
int *color = new int[n];
```

```
for (int i=0; i<n; i++)  
    color[i] = -1;
```

```
color[0] = 0;
```

```
if (graphColouringUtil (graph, n, color, 1)){
```

```
    cout << "Vertex colours";
```

```
    for (int i=0; i<n; i++){
```

```
        if (color[i] == 0)
```

```
            cout << "Red";
```

```
        else if (color[i] == 1)
```

```
            cout << "Blue";
```

```
        else if (color[i] == 2)
```

```
            cout << "Green";
```

```
    }
```

```
    }
```

```
    else
```

```
        cout << "Not valid";
```

```
}
```

```
int main(){
```

```
    int n;
```

```
    cin >> n;
```

```
    int **graph = new int*[n];
```

```
    readGraph (graph, n);
```

```
    graphColouring (graph, n);
```

```
    return 0;
```

```
}
```

## OUTPUT

6

0 1 0 1 1 0

1 0 1 1 0 1

0 1 0 1 1 1

1 1 1 0 1 0

1 0 1 1 0 0

0 1 1 0 0 0

Vertex 0 is coloured with Red

Vertex 1 is coloured with Blue

Vertex 2 is coloured with Red

Vertex 3 is coloured with Green

Vertex 4 is coloured with Blue

Vertex 5 is coloured with Green

```

② #include <iostream>
#include <windows.h>
using namespace std;
void readGraph (int ** graph, int n) {
    for (int i = 0; i < n; i++) {
        graph[i] = new int[n];
        for (int j = 0; j < n; j++)
            cin >> graph[i][j];
    }
}

bool isSafe (int ** graph, int n, int * color, int n, int v) {
    for (int i = 0; i < n; i++) {
        if (graph[v][i] && color[i] == color[v])
            return false;
    }
    return true;
}

bool graphColouringUtil (int ** graph, int n, int * color, int v) {
    if (v == n)
        return true;
    for (int c = 0; c < 3; c++) {
        if (isSafe (graph, n, color, v, c)) {
            color[v] = c;
            if (graphColouringUtil (graph, n, color, v+1)) {
                return true;
            }
            color[v] = -1;
        }
    }
    return false;
}

void graphColouring (int ** graph, int n) {
    int * color = new int[n];
    for (int i = 0; i < n; i++)
        color[i] = -1;
}

```

```
color[0] = 0;
```

```
if (graphColouringUtil(graph, n, color, 1)) {
```

```
    cout << "Vertex colour : ";
```

```
    for (int i = 0; i < n; i++) {
```

```
        if (color[i] == 0) cout << "Red ";
```

```
        else if (color[i] == 1) cout << "Blue ";
```

```
        else if (color[i] == 2) cout << "Green";
```

```
    }
```

```
}
```

```
else cout << "Not valid";
```

```
}
```

```
void alarm(int seconds) {
```

```
    cout << "Alarm set for " << seconds << "s ";
```

```
    Beep(500, 5000);
```

```
    cout << "Alarm Rang! " << endl;
```

```
}
```

```
int main() {
```

```
    int n;
```

```
    cin >> n;
```

```
    int *** graph = new int[n];
```

```
    readGraph(graph, n);
```

```
    alarm(5);
```

```
    graphColouring(graph, n);
```

```
    return 0;
```

```
}
```

## OUTPUT

6

0 1 0 1 1 0

1 0 1 1 0 1

0 1 0 1 1 1

1 1 1 0 1 0

1 0 1 1 0 0

0 1 1 0 0 0

Alarm set for 5 seconds

Alarm Rang!

Vertex 0 is coloured with Red

Vertex 1 is coloured with Blue

Vertex 2 is coloured with Red

Vertex 3 is coloured with Green

Vertex 4 is coloured with Blue

Vertex 5 is coloured with Green