

```
// Vertex Cover
```

```
// Wiki: Vertex Cover:
```

```
// In the mathematical discipline of graph theory, a vertex cover (sometimes node cover)
```

```
// of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex of the set
```

```
// Wiki: Edge Cover:
```

```
// In graph theory, an edge cover of a graph is a set of edges such that every vertex of the graph
```

```
// is incident to at least one edge of the set
```

```
// Min Edge Cover = TotalNodes - MinVertexCover
```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
bitset<210>vis;
```

```
int lft[210], rht[210];
```

```
vector<int>G[210];
```

```
bool VertexCover(int u) { //Min Vertex Cover
```

```
    vis[u] = 1;
```

```
    for(int i = 0; i < (int)G[u].size(); ++i) {
```

```
        int v = G[u][i];
```

```
        if(vis[v])
```

```
            continue;
```

```
        vis[v] = 1;
```

```
        if(lft[v] == -1) {
```

```
            lft[v] = u;
```

```
            rht[u] = v;
```

```
            return 1;
```

```
        }
```

```
        else if(VertexCover(lft[v])) {
```

```
            lft[v] = u;
```

```
            rht[u] = v;
```

```
            return 1;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

```
int main() {
```

```
    memset(lft, -1, sizeof lft);
```

```
    memset(rht, -1, sizeof rht);
```

```
    for(int i = 1; i <= n; ++i) {
```

```
        vis.reset();
```

```
        MinVertexCover += VertexCover(i);
```

```
    }
```

```
    return 0;
```

```
}
```

```
// n is the number of left-side nodes
```

```
// if left-side nodes are unspecified then do bi-coloring
```

```
// MinVertexCover is the answer
```