```
#include <stdio.h>
#include <stdlib.h>

typedef struct _edges {
        int num_edges;
        int * edges;
} edges;

edges * read_graph(int * n) {
        edges * E;
        int i, j, ne;

        scanf("%d", n);
       E = (edges *) malloc(*n * sizeof(edges));
        for (i = 0; i < *n; ++i) {
                scanf("%d", &ne);
                E[i].num_edges = ne;
                E[i].edges = malloc(ne * sizeof(int));
                for (j=0; j < ne; ++j) {
                        scanf("%d", &(E[i].edges[j]));
                }
        }
        return E;
}

int dfs_bicolor_recursive(edges * E, int from, int * colors) {
        int i, dest;
        for (i=0; i < E[from].num_edges; ++i) {
                dest = E[from].edges[i];
                if (colors[dest] == 0) {
                        colors[dest] = -colors[from];
                        if (dfs_bicolor_recursive(E, dest, colors)
== -1) {
                                return -1;
                        }
                } else if (colors[dest] == colors[from]) {
                        return -1;
                }
        }
        return 0;
}

int dfs_bicolor(edges * E, int n) {
        int * colors = malloc(n * sizeof(int));
        int i;
        for (i = 0; i < n; ++i) {
                colors[i] = 0;
        }

        for (i = 0; i < n; ++i) {
                if (colors[i] == 0) {
                        colors[i] = 1;
                        if (dfs_bicolor_recursive(E, i, colors) ==
-1) {
```

```
                                    return -1;
                            }
                    }
            }
            return 0;
}

int main() {
            int n;
            edges * E = read_graph(&n);

            printf("%d", 1+dfs_bicolor(E, n));
            return 0;
}
```