

```

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

typedef struct _edges {
    int grado;
    int * adiacenti;
} edges;

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

int dfs(edges * E, int s, int * colore) {
    int i, v;
    for(i = 0; i<E[s].grado; i++) {
        v = E[s].adiacenti[i];
        if(colore[v] == 0) {
            colore[v] = -colore[s];
            if(dfs(E, v, colore) == 0) return 0;
        }
        else if (colore[s] == colore[v]) return 0;
    }
    return 1;
}

int bipartito(edges * E, int n) {
    int i;
    int * colore = malloc(n*sizeof(int));
    for(i=0; i<n; i++) colore[i] = 0;
    for(i=0; i<n; i++) {
        if(colore[i] == 0) {
            colore[i] = 1;
            if(dfs(E, i, colore) == 0) {
                free(colore);
                return 0;
            }
        }
    }
    return 1;
}

int main() {
    int n, i;

```

```
scanf("%d",&n);
edges * E = read_graph(n);
printf("%d", bipartito(E,n));
for (i = 0; i < n; i++) free(E[i].adiacenti);
free(E);
return 0;
}
```