1. Intro. This program outputs unsatisfiable clauses based on a given 4-regular graph G. Let $\tilde{u} — \tilde{v}$ be the first edge of G; break this edge in half by inserting a new dummy vertex. The resulting graph G' has $2n+1$ edges. We construct clauses that essentially force G' to have equally many true edges as false edges at each of its vertices. (That can't happen, because it would imply an Eulerian trail of odd length in which true and false edges alternate.)

```
#include <stdio.h>
#include <stdlib.h>
#include "gb_graph.h"
#include "gb_save.h"
  main(int argc, char *argv[])
  {
    register int j, k;
    register Graph *g;
    register Vertex *u, *v;
    Vertex *utilde, *vtilde;
    Arc *a, *b;
    ⟨ Process the command line 2 ⟩;
    ⟨ Output the clauses 3 ⟩;
  }
```

2. ⟨ Process the command line 2 ⟩ ≡

```
  if (argc ≠ 2) {
    fprintf(stderr, "Usage:␣%s␣foo.gb\n", argv[0]);
    exit(−1);
  }
  g = restore_graph(argv[1]);
  if (¬g) {
    fprintf(stderr, "I␣couldn't␣reconstruct␣graph␣%s!\n", argv[1]);
    exit(−2);
  }
  for (v = g→vertices; v < g→vertices + g→n; v++) {
    for (j = 0, a = v→arcs; a; a = a→next) j++;
    if (j ≠ 4) {
      fprintf(stderr, "Vertex␣%s␣has␣degree␣%d,␣not␣4!\n", v→name, j);
      exit(−3);
    }
  }
  utilde = g→vertices;
  vtilde = utilde→arcs→tip;
  printf("~␣sat-eulerian-balance␣%s\n", argv[1]);
```

This code is used in section 1.

3. ⟨ Output the clauses 3 ⟩ ≡

```
for (u = g→vertices; u < g→vertices + g→n; u++) {
  for (a = u→arcs; a; a = a→next) {
    for (b = u→arcs; b; b = b→next)
      if (b ≠ a) {
        printf("␣%s%s.%s", ((u ≡ utilde) ∧ (b→tip ≡ vtilde)) ? "~" : "", u < b→tip ? u→name : b→tip→name,
            u < b→tip ? b→tip→name : u→name);
      }
    printf("\n");
    for (b = u→arcs; b; b = b→next)
      if (b ≠ a) {
        printf("␣%s%s.%s", ((u ≡ utilde) ∧ (b→tip ≡ vtilde)) ? "" : "~", u < b→tip ? u→name : b→tip→name,
            u < b→tip ? b→tip→name : u→name);
      }
    printf("\n");
  }
}
```

This code is used in section 1.

4. Index.

a: 1.
Arc: 1.
arcs: 2, 3.
argc: 1, 2.
argv: 1, 2.
b: 1.
exit: 2.
fprintf: 2.
g: 1.
Graph: 1.
j: 1.
k: 1.
main: 1.
name: 2, 3.
next: 2, 3.
printf: 2, 3.
restore_graph: 2.
stderr: 2.
tip: 2, 3.
u: 1.
utilde: 1, 2, 3.
v: 1.
Vertex: 1.
vertices: 2, 3.
vtilde: 1, 2, 3.

⟨ Output the clauses 3 ⟩ Used in section 1.
⟨ Process the command line 2 ⟩ Used in section 1.

SAT-EULERIAN-BALANCE

	Section	Page
Intro	1	1
Index	4	3