1. Intro. Supplementary clauses to speed up `sat-color-order queen`$nx n$`.gb` d: These clauses say that every k-clique must contain at least one relatively high color and at least one relative low color.

```
#include <stdio.h>
#include <stdlib.h>
  int n;      /* this many queens */
  int d;      /* this many colors */
  main(int argc, char *argv[])
  {
    register int i, j, k, l;
    ⟨ Process the command line 2 ⟩;
    ⟨ Generate the clauses 3 ⟩;
  }
```

2. ⟨ Process the command line 2 ⟩ ≡

```
  if (argc ≠ 3 ∨ sscanf(argv[1], "%d", &n) ≠ 1 ∨ sscanf(argv[2], "%d", &d) ≠ 1) {
    fprintf(stderr, "Usage: %s n d\n", argv[0]);
    exit(−1);
  }
  if (d < n) {
    fprintf(stderr, "The number of colors (%d) must be at least %d!\n", d, n);
    exit(−2);
  }
  printf("~ sat-queens-color-order-cliques %d %d\n", n, d);
```

This code is used in section 1.

3. ⟨ Generate the clauses 3 ⟩ ≡

```
  for (k = 0; k < n; k++) {
    ⟨ Generate cliques for row k 4 ⟩;
    ⟨ Generate cliques for column k 5 ⟩;
  }
  for (k = 1; k ≤ n + n − 3; k++) ⟨ Generate cliques for i + j = k 6 ⟩;
  for (k = 2 − n; k ≤ n − 2; k++) ⟨ Generate cliques for i − j = k 7 ⟩;
```

This code is used in section 1.

4. ⟨ Generate cliques for row k 4 ⟩ ≡

```
  {
    for (j = 0; j < n; j++) printf(" %d.%d<%d", k, j, d − n + 1);
    printf("\n");
    for (j = 0; j < n; j++) printf(" ~%d.%d<%d", k, j, n − 1);
    printf("\n");
  }
```

This code is used in section 3.

5. ⟨ Generate cliques for column k 5 ⟩ ≡

```
  {
    for (j = 0; j < n; j++) printf(" %d.%d<%d", j, k, d − n + 1);
    printf("\n");
    for (j = 0; j < n; j++) printf(" ~%d.%d<%d", j, k, n − 1);
    printf("\n");
  }
```

This code is used in section 3.

6. ⟨ Generate cliques for $i+j=k$ 6 ⟩ ≡

```
{
  if (k < n) {
    l = k + 1;
    for (i = 0; i ≤ k; i++) printf("␣%d.%d<%d", i, k − i, d − l + 1);
    printf("\n");
    for (i = 0; i ≤ k; i++) printf("␣~%d.%d<%d", i, k − i, l − 1);
    printf("\n");
  } else {
    l = n + n − 1 − k;
    for (i = n − l; i < n; i++) printf("␣%d.%d<%d", i, k − i, d − l + 1);
    printf("\n");
    for (i = n − l; i < n; i++) printf("␣~%d.%d<%d", i, k − i, l − 1);
    printf("\n");
  }
}
```

This code is used in section 3.

7. ⟨ Generate cliques for $i-j=k$ 7 ⟩ ≡

```
{
  if (k > 0) {
    l = n − k;
    for (i = k; i < n; i++) printf("␣%d.%d<%d", i, i − k, d − l + 1);
    printf("\n");
    for (i = k; i < n; i++) printf("␣~%d.%d<%d", i, i − k, l − 1);
    printf("\n");
  } else {
    l = n + k;
    for (i = 0; i < n + k; i++) printf("␣%d.%d<%d", i, i − k, d − l + 1);
    printf("\n");
    for (i = 0; i < n + k; i++) printf("␣~%d.%d<%d", i, i − k, l − 1);
    printf("\n");
  }
}
```

This code is used in section 3.

8. Index.

SAT-QUEENS-COLOR-ORDER-CLIQUES