1. Intro. This program makes DLX data to find all ways to attack or occupy all cells of an $n \times n$ board with m queens.

```
#define maxn 16    /* hexadecimal limitation */
#include <stdio.h>
#include <stdlib.h>
  int m, n;    /* command-line parameters */
  main(int argc, char *argv[])
  {
    register int i, j, k;
    ⟨Process the command line 2⟩;
    ⟨Print the item-name line 3⟩;
    for (i = 0; i < n; i++)
      for (j = 0; j < n; j++) ⟨Print the option for a queen at position (i, j) 4⟩;
  }
```

2. ⟨Process the command line 2⟩ ≡

```
  if (argc ≠ 3 ∨ sscanf(argv[1], "%d", &n) ≠ 1 ∨ sscanf(argv[2], "%d", &m) ≠ 1) {
    fprintf(stderr, "Usage:␣%s␣n␣m\n", argv[0]);
    exit(−1);
  }
  if (n > maxn) {
    fprintf(stderr, "Sorry,␣I␣don't␣presently␣allow␣n>%d!\n", maxn);
    exit(−2);
  }
  printf("|␣%s␣%d␣%d\n", argv[0], n, m);
```

This code is used in section 1.

3. ⟨Print the item-name line 3⟩ ≡

```
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++) printf("1:%d|%x%x␣", m, i, j);
  printf("%d|Q\n", m);
```

This code is used in section 1.

4. ⟨Print the option for a queen at position (i, j) 4⟩ ≡

```
  {
    printf("Q␣%x%x", i, j);
    for (k = 0; k < n; k++)
      if (k ≠ i) printf("␣%x%x", k, j);
    for (k = 0; k < n; k++)
      if (k ≠ j) printf("␣%x%x", i, k);
    for (k = 1; i + k < n ∧ j + k < n; k++) printf("␣%x%x", i + k, j + k);
    for (k = 1; i − k ≥ 0 ∧ j − k ≥ 0; k++) printf("␣%x%x", i − k, j − k);
    for (k = 1; i + k < n ∧ j − k ≥ 0; k++) printf("␣%x%x", i + k, j − k);
    for (k = 1; i − k ≥ 0 ∧ j + k < n; k++) printf("␣%x%x", i − k, j + k);
    printf("\n");
  }
```

This code is used in section 1.

5. Index.

⟨ Print the item-name line 3 ⟩ Used in section 1.
⟨ Print the option for a queen at position (i,j) 4 ⟩ Used in section 1.
⟨ Process the command line 2 ⟩ Used in section 1.

QUEENDOM-DLX