

# Practical 7

## 8-Queen Problem using Back Tracking

GAHAN M. SARAIYA, 18MCEC10

[18mcec10@nirmauni.ac.in](mailto:18mcec10@nirmauni.ac.in)

### I. INTRODUCTION

Aim of this practical is to perform chain matrix multiplication using Dynamic Programming.

- A 8 x 8 chess board and 8 queens where Queens can move horizontally, vertically and diagonally.
- A chessboard with N queens arranged on the board in such a way that no two queens can attack each other.
- use backtracking each time we got bounded by constraints i.e, when we have no more place valid to place the new queens in the board.
- Also, before placing a new queen to the board, at first the position in the board should be thoroughly checked and if the place is valid then we could place a new queen on the board.

### II. IMPLEMENTATION

#### I. Main Program - *8queen.c*

---

```
1  //  
2  ~~~~~  
3  // Author: Gahan Saraiya  
4  // GiT: http://github.com/gahan9/  
5  // StackOverflow: https://stackoverflow.com/users/story/7664524  
6  // Website: http://gahan9.github.io/  
7  //  
8  ~~~~~  
9  // Implementing 8-queen Problem  
10  
11 #include <stdio.h>  
12 #include <stdlib.h>  
13 #include <stdbool.h>  
14 #include <math.h>  
15  
16 int counter = 1;  
17  
18 int placeQueen(int *board, int row_no) {  
19     for (register int i = 0; i < row_no; i++) {  
20         if (board[i] == board[row_no])  
21             return false; // same column
```

```
20     if ((board[i] - board[row_no]) == (row_no - i))
21         return false; // same major diagonal
22     if ((board[row_no] - board[i]) == (row_no - i))
23         return false; // same minor diagonal
24 }
25 return true;
26 }
27
28 void pretty_print(int q[], int board_size) {
29     printf("\n");
30     for (int i = 0; i < board_size; i++) {
31         for (int j = 0; j < board_size; j++) {
32             if (q[i] == j)
33                 printf("Q ");
34             else
35                 printf("* ");
36         }
37         printf("\n");
38     }
39 }
40
41 void queenLogic(int *queens, int row_no, int board_size) {
42     if (row_no == board_size) {
43         pretty_print(queens, board_size);
44         if (counter % 10 == 1)
45             printf("%dst solution found...\n", counter);
46         else if (counter % 10 == 2)
47             printf("%dnd solution found...\n", counter);
48         else if (counter % 10 == 3)
49             printf("%drd solution found...\n", counter);
50         else
51             printf("%dth solution found...\n", counter);
52         counter++;
53     } else {
54         for (int i = 0; i < board_size; i++) {
55             queens[row_no] = i;
56             if (placeQueen(queens, row_no))
57                 queenLogic(queens, row_no + 1, board_size);
58         }
59     }
60 }
61
62 int main() {
63     int board_size = 8;
64     int *queens = malloc(board_size * sizeof(int));
65     printf("#####-----#\n"
66           "#----- %d queen Problem -----#\n"
67           "#-----#\n", board_size);
```

```

68     queenLogic(queens, 0, board_size);
69     return 1;
70 }

```

## I.1 Output

```

1 #####-----#####
2 #----- 08 queen Problem -----#
3 #-----#
4
5 Q * * * * *
6 * * * * Q * *
7 * * * * * * Q
8 * * * * * Q *
9 * * Q * * * *
10 * * * * * * Q
11 * Q * * * *
12 * * * Q * *
13 1st solution found...
14
15 Q * * * * *
16 * * * * * Q *
17 * * * * * * Q
18 * * Q * * * *
19 * * * * * Q *
20 * * * Q * * *
21 * Q * * * *
22 * * * * Q *
23 2nd solution found...
24
25 Q * * * * *
26 * * * * * Q *
27 * * * Q * * *
28 * * * * * Q *
29 * * * * * * Q
30 * Q * * * *
31 * * * * Q *
32 * * Q * * *
33 3rd solution found...
34
35 Q * * * * *
36 * * * * * Q *

```

```

37 * * * * Q * * *
38 * * * * * * * Q
39 * Q * * * * *
40 * * * Q * * *
41 * * * * * Q *
42 * * Q * * * *
43 4th solution found...
44
45 * Q * * * * *
46 * * * Q * * *
47 * * * * * Q *
48 * * * * * * * Q
49 * * Q * * * *
50 Q * * * * *
51 * * * * * Q *
52 * * * * Q * *
53 5th solution found...
54
55 * Q * * * * *
56 * * * * Q * *
57 * * * * * Q *
58 Q * * * * *
59 * * Q * * * *
60 * * * * * * Q
61 * * * * * Q *
62 * * * Q * * *
63 6th solution found...
64
65 * Q * * * * *
66 * * * * Q * *
67 * * * * * Q *
68 * * * Q * * *
69 Q * * * * *
70 * * * * * * Q
71 * * * * * Q *
72 * * Q * * * *
73 7th solution found...
74
75 * Q * * * * *
76 * * * * * Q *
77 Q * * * * *

```

```

78 * * * * * Q *
79 * * * Q * * * *
80 * * * * * * * Q
81 * * Q * * * * *
82 * * * * Q * * *
83 8th solution found...
84
85 * Q * * * * * *
86 * * * * * Q * *
87 * * * * * * * Q
88 * * Q * * * * *
89 Q * * * * * * *
90 * * * Q * * * *
91 * * * * * * Q *
92 * * * * Q * * *
93 9th solution found...
94
95 * Q * * * * * *
96 * * * * * * Q *
97 * * Q * * * * *
98 * * * * * Q * *
99 * * * * * * * Q
100 * * * * Q * * *
101 Q * * * * * * *
102 * * * Q * * * *
103 10th solution found...
104
105 * Q * * * * * *
106 * * * * * * Q *
107 * * * * Q * * *
108 * * * * * * * Q
109 Q * * * * * * *
110 * * * Q * * * *
111 * * * * * Q * *
112 * * Q * * * * *
113 11st solution found...
114
115 * Q * * * * * *
116 * * * * * * * Q
117 * * * * * Q * *
118 Q * * * * * * *

```

```

119 * * Q * * * * *
120 * * * * Q * * *
121 * * * * * * Q *
122 * * * Q * * * *
123 12nd solution found...
124
125 * * Q * * * * *
126 Q * * * * * * *
127 * * * * * * Q *
128 * * * * Q * * *
129 * * * * * * * Q
130 * Q * * * * * *
131 * * * Q * * * *
132 * * * * * Q * *
133 13rd solution found...
134
135 * * Q * * * * *
136 * * * * Q * * *
137 * Q * * * * * *
138 * * * * * * * Q
139 Q * * * * * * *
140 * * * * * * Q *
141 * * * Q * * * *
142 * * * * * Q * *
143 14th solution found...
144
145 * * Q * * * * *
146 * * * * Q * * *
147 * Q * * * * * *
148 * * * * * * * Q
149 * * * * * Q * *
150 * * * Q * * * *
151 * * * * * * Q *
152 Q * * * * * * *
153 15th solution found...
154
155 * * Q * * * * *
156 * * * * Q * * *
157 * * * * * * Q *
158 Q * * * * * * *
159 * * * Q * * * *

```

```

160 * Q * * * * *
161 * * * * * Q
162 * * * * * Q * *
163 16th solution found...
164
165 * * Q * * * * *
166 * * * * * Q * *
167 * * * * * * * Q
168 * * * Q * * * *
169 Q * * * * * * *
170 * * * * * * Q *
171 * Q * * * * *
172 * * * * * Q * *
173 17th solution found...
174
175 * * Q * * * * *
176 * * * * * Q * *
177 * Q * * * * *
178 * * * * * Q * *
179 * * * * * * * Q
180 Q * * * * * * *
181 * * * * * * Q *
182 * * * Q * * * *
183 18th solution found...
184
185 * * Q * * * * *
186 * * * * * Q * *
187 * Q * * * * *
188 * * * * * * Q *
189 Q * * * * * * *
190 * * * Q * * * *
191 * * * * * * * Q
192 * * * * * Q * *
193 19th solution found...
194
195 * * Q * * * * *
196 * * * * * Q * *
197 * Q * * * * *
198 * * * * * * Q *
199 * * * * * Q * *
200 Q * * * * * * *

```

```

201 * * * * * * * Q
202 * * * Q * * * *
203 20th solution found...
204
205 * * Q * * * * *
206 * * * * * Q * *
207 * * * Q * * * *
208 Q * * * * * * *
209 * * * * * * * Q
210 * * * * Q * * *
211 * * * * * * Q *
212 * Q * * * * * *
213 21st solution found...
214
215 * * Q * * * * *
216 * * * * * Q * *
217 * * * Q * * * *
218 * Q * * * * * *
219 * * * * * * * Q
220 * * * * Q * * *
221 * * * * * * Q *
222 Q * * * * * * *
223 22nd solution found...
224
225 * * Q * * * * *
226 * * * * * Q * *
227 * * * * * * * Q
228 Q * * * * * * *
229 * * * Q * * * *
230 * * * * * * Q *
231 * * * * Q * * *
232 * Q * * * * * *
233 23rd solution found...
234
235 * * Q * * * * *
236 * * * * * Q * *
237 * * * * * * * Q
238 Q * * * * * * *
239 * * * * Q * * *
240 * * * * * * Q *
241 * Q * * * * * *

```



```

242 * * * Q * * * *
243 24th solution found...
244
245 * * Q * * * *
246 * * * * * Q *
247 * * * * * * * Q
248 * Q * * * * *
249 * * * Q * * *
250 Q * * * * *
251 * * * * * Q *
252 * * * * Q * *
253 25th solution found...
254
255 * * Q * * * *
256 * * * * * Q *
257 * Q * * * * *
258 * * * * * * * Q
259 * * * * Q * *
260 Q * * * * *
261 * * * Q * * *
262 * * * * * Q *
263 26th solution found...
264
265 * * Q * * * *
266 * * * * * Q *
267 * Q * * * * *
268 * * * * * * * Q
269 * * * * * Q *
270 * * * Q * * *
271 Q * * * * *
272 * * * * Q * *
273 27th solution found...
274
275 * * Q * * * *
276 * * * * * * * Q
277 * * * Q * * *
278 * * * * * Q *
279 Q * * * * *
280 * * * * * Q *
281 * Q * * * * *
282 * * * * Q * *

```

283 28th solution found...

284

285 \* \* \* Q \* \* \* \*

286 Q \* \* \* \* \* \*

287 \* \* \* \* Q \* \* \*

288 \* \* \* \* \* \* Q

289 \* Q \* \* \* \* \* \*

290 \* \* \* \* \* Q \*

291 \* \* Q \* \* \* \* \*

292 \* \* \* \* \* Q \* \*

293 29th solution found...

294

295 \* \* \* Q \* \* \* \*

296 Q \* \* \* \* \* \*

297 \* \* \* \* Q \* \* \*

298 \* \* \* \* \* \* Q

299 \* \* \* \* \* Q \* \*

300 \* \* Q \* \* \* \* \*

301 \* \* \* \* \* Q \*

302 \* Q \* \* \* \* \* \*

303 30th solution found...

304

305 \* \* \* Q \* \* \* \*

306 \* Q \* \* \* \* \* \*

307 \* \* \* \* Q \* \* \*

308 \* \* \* \* \* \* Q

309 \* \* \* \* \* Q \* \*

310 Q \* \* \* \* \* \*

311 \* \* Q \* \* \* \* \*

312 \* \* \* \* \* Q \*

313 31st solution found...

314

315 \* \* \* Q \* \* \* \*

316 \* Q \* \* \* \* \* \*

317 \* \* \* \* \* Q \*

318 \* \* Q \* \* \* \* \*

319 \* \* \* \* \* Q \* \*

320 \* \* \* \* \* \* Q

321 Q \* \* \* \* \* \*

322 \* \* \* \* Q \* \* \*

323 32nd solution found...

```

324
325 * * * Q * * * *
326 * Q * * * * *
327 * * * * * Q *
328 * * Q * * * *
329 * * * * * Q *
330 * * * * * * Q
331 * * * * Q * *
332 Q * * * * *
333 33rd solution found...
334
335 * * * Q * * * *
336 * Q * * * * *
337 * * * * * Q *
338 * * * * Q * *
339 Q * * * * *
340 * * * * * * Q
341 * * * * * Q *
342 * * Q * * * *
343 34th solution found...
344
345 * * * Q * * * *
346 * Q * * * * *
347 * * * * * * Q
348 * * * * Q * *
349 * * * * * Q *
350 Q * * * * *
351 * * Q * * * *
352 * * * * * Q *
353 35th solution found...
354
355 * * * Q * * * *
356 * Q * * * * *
357 * * * * * * Q
358 * * * * * Q *
359 Q * * * * *
360 * * Q * * * *
361 * * * * Q * *
362 * * * * * Q *
363 36th solution found...
364

```

```

365 * * * Q * * * *
366 * * * * * Q * *
367 Q * * * * * * *
368 * * * * Q * * *
369 * Q * * * * * *
370 * * * * * * * Q
371 * * Q * * * * *
372 * * * * * * Q *
373 37th solution found...
374
375 * * * Q * * * *
376 * * * * * Q * *
377 * * * * * * * Q
378 * Q * * * * * *
379 * * * * * * Q *
380 Q * * * * * * *
381 * * Q * * * * *
382 * * * * Q * * *
383 38th solution found...
384
385 * * * Q * * * *
386 * * * * * Q * *
387 * * * * * * * Q
388 * * Q * * * * *
389 Q * * * * * * *
390 * * * * * * Q *
391 * * * * Q * * *
392 * Q * * * * * *
393 39th solution found...
394
395 * * * Q * * * *
396 * * * * * * Q *
397 Q * * * * * * *
398 * * * * * * * Q
399 * * * * Q * * *
400 * Q * * * * * *
401 * * * * * Q * *
402 * * Q * * * * *
403 40th solution found...
404
405 * * * Q * * * *

```

```

406 * * * * * Q *
407 * * Q * * * *
408 * * * * * Q
409 * Q * * * *
410 * * * * Q * *
411 Q * * * * *
412 * * * * Q *
413 41st solution found...
414
415 * * * Q * * *
416 * * * * * Q *
417 * * * * Q * *
418 * Q * * * *
419 * * * * Q *
420 Q * * * * *
421 * * Q * * *
422 * * * * * Q
423 42nd solution found...
424
425 * * * Q * * *
426 * * * * * Q *
427 * * * * Q * *
428 * * Q * * *
429 Q * * * * *
430 * * * * Q *
431 * * * * * Q
432 * Q * * * *
433 43rd solution found...
434
435 * * * Q * * *
436 * * * * * Q
437 Q * * * * *
438 * * Q * * *
439 * * * * Q *
440 * Q * * * *
441 * * * * * Q
442 * * * * Q *
443 44th solution found...
444
445 * * * Q * * *
446 * * * * * Q

```

```

447 Q * * * * *
448 * * * * Q * * *
449 * * * * * Q *
450 * Q * * * * *
451 * * * * * Q * *
452 * * Q * * * * *
453 45th solution found...
454
455 * * * * Q * * * *
456 * * * * * * * Q
457 * * * * Q * * *
458 * * Q * * * * *
459 Q * * * * * * *
460 * * * * * * Q *
461 * Q * * * * * *
462 * * * * * Q * *
463 46th solution found...
464
465 * * * * Q * * *
466 Q * * * * * * *
467 * * * * Q * * *
468 * * * * * Q * *
469 * * * * * * * Q
470 * Q * * * * * *
471 * * * * * * Q *
472 * * Q * * * * *
473 47th solution found...
474
475 * * * * Q * * *
476 Q * * * * * * *
477 * * * * * * * Q
478 * * * Q * * * *
479 * Q * * * * * *
480 * * * * * * Q *
481 * * Q * * * * *
482 * * * * * Q * *
483 48th solution found...
484
485 * * * * Q * * *
486 Q * * * * * * *
487 * * * * * * * Q

```

```

488 * * * * * Q * *
489 * * Q * * * * *
490 * * * * * * Q *
491 * Q * * * * * *
492 * * * Q * * * *
493 49th solution found...
494
495 * * * * * Q * * *
496 * Q * * * * * *
497 * * * Q * * * *
498 * * * * * Q * *
499 * * * * * * * Q
500 * * Q * * * * *
501 Q * * * * * * *
502 * * * * * * Q *
503 50th solution found...
504
505 * * * * * Q * * *
506 * Q * * * * * *
507 * * * Q * * * *
508 * * * * * * Q *
509 * * Q * * * * *
510 * * * * * * * Q
511 * * * * * Q * *
512 Q * * * * * * *
513 51st solution found...
514
515 * * * * * Q * * *
516 * Q * * * * * *
517 * * * * * Q * *
518 Q * * * * * * *
519 * * * * * * Q *
520 * * * Q * * * *
521 * * * * * * * Q
522 * * Q * * * * *
523 52nd solution found...
524
525 * * * * * Q * * *
526 * Q * * * * * *
527 * * * * * * * Q
528 Q * * * * * * *

```

```

529 * * * Q * * * *
530 * * * * * * Q *
531 * * Q * * * * *
532 * * * * * Q * *
533 53rd solution found...
534
535 * * * * Q * * *
536 * * Q * * * * *
537 Q * * * * * * *
538 * * * * * Q * *
539 * * * * * * * Q
540 * Q * * * * * *
541 * * * Q * * * *
542 * * * * * * Q *
543 54th solution found...
544
545 * * * * Q * * *
546 * * Q * * * * *
547 Q * * * * * * *
548 * * * * * * Q *
549 * Q * * * * * *
550 * * * * * * * Q
551 * * * * * Q * *
552 * * * Q * * * *
553 55th solution found...
554
555 * * * * Q * * *
556 * * Q * * * * *
557 * * * * * * * Q
558 * * * Q * * * *
559 * * * * * * Q *
560 Q * * * * * * *
561 * * * * * Q * *
562 * Q * * * * * *
563 56th solution found...
564
565 * * * * Q * * *
566 * * * * * * Q *
567 Q * * * * * * *
568 * * Q * * * * *
569 * * * * * * * Q

```



```

570 * * * * * Q * *
571 * * * Q * * * *
572 * Q * * * * *
573 57th solution found...
574
575 * * * * * Q * * *
576 * * * * * * Q *
577 Q * * * * * *
578 * * * Q * * * *
579 * Q * * * * *
580 * * * * * * Q
581 * * * * * Q * *
582 * * Q * * * *
583 58th solution found...
584
585 * * * * * Q * * *
586 * * * * * * Q *
587 * Q * * * * *
588 * * * Q * * * *
589 * * * * * * Q
590 Q * * * * * *
591 * * Q * * * *
592 * * * * * Q * *
593 59th solution found...
594
595 * * * * * Q * * *
596 * * * * * * Q *
597 * Q * * * * *
598 * * * * * Q * *
599 * * Q * * * *
600 Q * * * * * *
601 * * * Q * * * *
602 * * * * * * Q
603 60th solution found...
604
605 * * * * * Q * * *
606 * * * * * * Q *
607 * Q * * * * *
608 * * * * * Q * *
609 * * Q * * * *
610 Q * * * * * *

```

```

611 * * * * * Q
612 * * * Q * * *
613 61st solution found...
614
615 * * * * Q * *
616 * * * * * Q *
617 * * * Q * * *
618 Q * * * * *
619 * * Q * * *
620 * * * * * Q
621 * * * * * Q *
622 * Q * * * *
623 62nd solution found...
624
625 * * * * Q * *
626 * * * * * Q
627 * * * Q * * *
628 Q * * * * *
629 * * Q * * *
630 * * * * * Q *
631 * Q * * * *
632 * * * * * Q *
633 63rd solution found...
634
635 * * * * Q * *
636 * * * * * Q
637 * * * Q * * *
638 Q * * * * *
639 * * * * * Q *
640 * Q * * * *
641 * * * * * Q *
642 * * Q * * *
643 64th solution found...
644
645 * * * * * Q *
646 Q * * * * *
647 * * * * Q * *
648 * Q * * * *
649 * * * * * Q
650 * * Q * * *
651 * * * * * Q *

```

```

652 * * * Q * * * *
653 65th solution found...
654
655 * * * * * Q * *
656 * Q * * * * *
657 * * * * * Q *
658 Q * * * * *
659 * * Q * * * *
660 * * * * Q * *
661 * * * * * Q
662 * * * Q * * *
663 66th solution found...
664
665 * * * * * Q * *
666 * Q * * * * *
667 * * * * * Q *
668 Q * * * * *
669 * * * Q * * *
670 * * * * * Q
671 * * * * Q * *
672 * * Q * * * *
673 67th solution found...
674
675 * * * * * Q * *
676 * * Q * * * *
677 Q * * * * *
678 * * * * * Q *
679 * * * * Q * *
680 * * * * * Q
681 * Q * * * * *
682 * * * Q * * *
683 68th solution found...
684
685 * * * * * Q * *
686 * * Q * * * *
687 Q * * * * *
688 * * * * * Q
689 * * * Q * * *
690 * Q * * * * *
691 * * * * * Q *
692 * * * * Q * *

```

693 69th solution found...

694

695 \* \* \* \* \* Q \* \*

696 \* \* Q \* \* \* \* \*

697 Q \* \* \* \* \* \*

698 \* \* \* \* \* \* Q

699 \* \* \* \* Q \* \* \*

700 \* Q \* \* \* \* \*

701 \* \* \* Q \* \* \* \*

702 \* \* \* \* \* Q \*

703 70th solution found...

704

705 \* \* \* \* \* Q \* \*

706 \* \* Q \* \* \* \* \*

707 \* \* \* \* Q \* \* \*

708 \* \* \* \* \* Q \*

709 Q \* \* \* \* \* \*

710 \* \* \* Q \* \* \* \*

711 \* Q \* \* \* \* \*

712 \* \* \* \* \* \* Q

713 71st solution found...

714

715 \* \* \* \* \* Q \* \*

716 \* \* Q \* \* \* \* \*

717 \* \* \* \* Q \* \* \*

718 \* \* \* \* \* \* Q

719 Q \* \* \* \* \* \*

720 \* \* \* Q \* \* \* \*

721 \* Q \* \* \* \* \*

722 \* \* \* \* \* Q \*

723 72nd solution found...

724

725 \* \* \* \* \* Q \* \*

726 \* \* Q \* \* \* \* \*

727 \* \* \* \* \* Q \*

728 \* Q \* \* \* \* \*

729 \* \* \* Q \* \* \* \*

730 \* \* \* \* \* \* Q

731 Q \* \* \* \* \* \*

732 \* \* \* \* Q \* \* \*

733 73rd solution found...

```

734
735 * * * * * Q * *
736 * * Q * * * * *
737 * * * * * * Q *
738 * Q * * * * * *
739 * * * * * * * Q
740 * * * * Q * * *
741 Q * * * * * * *
742 * * * Q * * * *
743 74th solution found...
744
745 * * * * * Q * *
746 * * Q * * * * *
747 * * * * * * Q *
748 * * * Q * * * *
749 Q * * * * * * *
750 * * * * * * * Q
751 * Q * * * * * *
752 * * * * Q * * *
753 75th solution found...
754
755 * * * * * Q * *
756 * * * Q * * * *
757 Q * * * * * * *
758 * * * * Q * * *
759 * * * * * * * Q
760 * Q * * * * * *
761 * * * * * * Q *
762 * * Q * * * * *
763 76th solution found...
764
765 * * * * * Q * *
766 * * * Q * * * *
767 * Q * * * * * *
768 * * * * * * * Q
769 * * * * Q * * *
770 * * * * * * Q *
771 Q * * * * * * *
772 * * Q * * * * *
773 77th solution found...
774

```

```

775 * * * * * Q * *
776 * * * Q * * * *
777 * * * * * * Q *
778 Q * * * * * * *
779 * * Q * * * * *
780 * * * * Q * * *
781 * Q * * * * * *
782 * * * * * * * Q
783 78th solution found...
784
785 * * * * * Q * *
786 * * * Q * * * *
787 * * * * * * Q *
788 Q * * * * * * *
789 * * * * * * * Q
790 * Q * * * * * *
791 * * * * Q * * *
792 * * Q * * * * *
793 79th solution found...
794
795 * * * * * Q * *
796 * * * * * * * Q
797 * Q * * * * * *
798 * * * Q * * * *
799 Q * * * * * * *
800 * * * * * * Q *
801 * * * * Q * * *
802 * * Q * * * * *
803 80th solution found...
804
805 * * * * * * Q *
806 Q * * * * * * *
807 * * Q * * * * *
808 * * * * * * * Q
809 * * * * * Q * *
810 * * * Q * * * *
811 * Q * * * * * *
812 * * * * Q * * *
813 81st solution found...
814
815 * * * * * * Q *

```

```

816 * Q * * * * *
817 * * * Q * * *
818 Q * * * * *
819 * * * * * Q
820 * * * * Q * *
821 * * Q * * * *
822 * * * * * Q *
823 82nd solution found...
824
825 * * * * * Q *
826 * Q * * * * *
827 * * * * * Q *
828 * * Q * * * *
829 Q * * * * *
830 * * * Q * * *
831 * * * * * Q
832 * * * * Q * *
833 83rd solution found...
834
835 * * * * * Q *
836 * * Q * * * *
837 Q * * * * *
838 * * * * * Q *
839 * * * * * Q
840 * * * * Q * *
841 * Q * * * * *
842 * * * Q * * *
843 84th solution found...
844
845 * * * * * Q *
846 * * Q * * * *
847 * * * * * Q
848 * Q * * * * *
849 * * * * Q * *
850 Q * * * * *
851 * * * * * Q *
852 * * * Q * * *
853 85th solution found...
854
855 * * * * * Q *
856 * * * Q * * *

```

```

857 * Q * * * * *
858 * * * * Q * * *
859 * * * * * * Q
860 Q * * * * * *
861 * * Q * * * * *
862 * * * * * Q * *
863 86th solution found...
864
865 * * * * * * Q *
866 * * * Q * * * *
867 * Q * * * * * *
868 * * * * * * Q
869 * * * * * Q * *
870 Q * * * * * *
871 * * Q * * * * *
872 * * * * Q * * *
873 87th solution found...
874
875 * * * * * * Q *
876 * * * * Q * * *
877 * * Q * * * * *
878 Q * * * * * *
879 * * * * * Q * *
880 * * * * * * Q
881 * Q * * * * * *
882 * * * Q * * * *
883 88th solution found...
884
885 * * * * * * Q
886 * Q * * * * * *
887 * * * Q * * * *
888 Q * * * * * *
889 * * * * * Q *
890 * * * * Q * * *
891 * * Q * * * * *
892 * * * * * Q * *
893 89th solution found...
894
895 * * * * * * Q
896 * Q * * * * * *
897 * * * * Q * * *

```



```

898 * * Q * * * * *
899 Q * * * * *
900 * * * * * Q *
901 * * * Q * * *
902 * * * * * Q *
903 90th solution found...
904
905 * * * * * * Q
906 * * Q * * * *
907 Q * * * * *
908 * * * * * Q *
909 * Q * * * * *
910 * * * * Q * *
911 * * * * * Q *
912 * * * Q * * *
913 91st solution found...
914
915 * * * * * * Q
916 * * * Q * * *
917 Q * * * * *
918 * * Q * * * *
919 * * * * * Q *
920 * Q * * * * *
921 * * * * * Q *
922 * * * * Q * *
923 92nd solution found...

```