

### 3) Implement N Queen's problem using Back Tracking.

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

int x[30], count = 0;

int place(int k, int i) {
    int j;
    for (j = 1; j <= k - 1; j++) {
        if ((x[j] == i) || (abs(x[j] - i) == abs(j - k)))
            return 0;
    }
    return 1;
}

void print_sol(int n) {
    int i, j;
    count++;
    printf("\n\nSolution # %d:\n", count);
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= n; j++) {
            if (x[i] == j)
                printf("Q\t");
            else
                printf("*\t");
        }
        printf("\n");
    }
}

void queen(int k, int n) {
    int i, val;
    for (i = 1; i <= n; i++) {
        if (place(k, i)) {
            x[k] = i;
            if (k == n)
                print_sol(n);
            else
                queen(k + 1, n);
        }
    }
}
```

```

    }
}

void main() {
    int i, n;
    printf("Enter the number of Queens\n");
    scanf("%d", &n);
    queen(1, n);
    printf("\nTotal solutions=%d", count);
}

```

```

Enter the number of Queens
4

```

```

Solution #1:

```

```

*      Q      *      *
*      *      *      Q
Q      *      *      *
*      *      Q      *

```

```

Solution #2:

```

```

*      *      Q      *
Q      *      *      *
*      *      *      Q
*      Q      *      *

```

```

Total solutions=2%

```