**8 Queens POC program ideas**

My idea to model the 8 queens problem initially without backtracking is to create objects to represent the Queens with attributes of x and y ‘position’ and another variable of type int which represents the queen’s position in a custom grid.

I was trying to think of an easy way to work out if one queen was threatening another on the diagonals. So I decided that I would number all the spaces of the grid ( 8x8 in this case but larger grid can be used without modification) as shown below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |

As we will be working top to bottom left to right I have numbered the spaces accordingly.

I initially need an array of type queen object, and then a main which will keep calling the check/add method until the number of objects in the array is = total number of queens.

The check/add method will create an object of type queen with x=0 and y = the row number which will be taken in as a variable and pos = 1 + x \* 1 + y \* 8 together with the array of objects.

The method will firstly check if the array is empty in which case it will simple assign the object to the empty array and return.

Else it will advance the x variable by one and check against each object in the array using the position variable. It will work out the current position ( using pos = 1 + x \* 1 + y \* 8) and mod 9 this against the position and mod 7 it against the postion, if we get a zero returned we know that there is a queen in a threatening positon and advance x else we can append this object to the array and return true.

For the backtracking I understand that if x = size -1 and we still have not returned true that we will have to null the current array object and return to the previous object and do the same.