The best stable formation is to form a straight line.

Min Multiplicity

Case 1:

```
if(r >= (2 * f) + 1) mMin = 3;
```

				R	R	R
				R	R	
				R	R	

Number of Robbers, r, is more than the twice the smaller side, f, the min multiplicity of cops required to break the formation is 3.

			CCC	**R**	R	R
			CCC	**R**	R	
			CCC	**R**	R	

The robbers **R** bold and big are vulnerable.

Case 2:

```
else if((c < n && r >= n + c) || (c < k && r >= k + c)) mMin = 3; // limited cops
```

This is a very tricky situation.

In the last column number of robbers is equal to number of cops.

In case of multiplicity = 2, the thing look like the table below. The robbers in the last column will imitate the cop's moves and stay safe. So min Multiplicity required is 3, to break the formation.

				CC	R	R
				CC	R	R
				CC	R	R
					R	
				CC	R	R
					R	

Everyone is sitting safe.

Case 3:

```
else if(r >= f + 1) mMin = 2;
```

					R	R
					R	
					R	

Number of Robbers, r, is more than the smaller side, f, the min multiplicity of cops required to break the formation is 2.

				CC	**R**	R
				CC	**R**	
				CC	**R**	

The robbers **R** bold and big are vulnerable.

Case 4:

```
else if(r > 8) mMin = 2; // diagonal
```

In case of diagonal, too many pegs are needed. But when the board is too big diagonal is the best formation using 9 pegs, the corner is kept to move freely.

CC						
R	CC					
R	R	CC				
	R	R	CC			
R		R	**R**	CC		

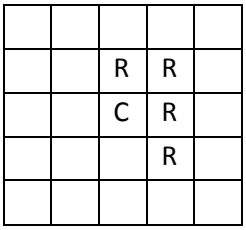
The robbers **R** bold and big are vulnerable, for multiplicity = 2.

When the board is small the best formation is STRAIGHT, and for large board DIAGONAL is best.

Max Multiplicity

```
int mMax = min(r, 8) + 1; // (max number of robber that can surround a cop) + 1 (to overpower the robbers)
```

To find the max Multiplicity we consider one Cop surrounded by all robbers, as shown below. The max number of robbers surrounding a cop will be 8, so the max multiplicity for extreme case will be (8+1)=9.



Conclusion

The absolute multiplicity lies between the max and the min values given above.