

ALGORITHME

Realisé par : Nada Ben Slimen

N reines

Le problème des n queen provient d'une relation de question aux échecs, le problème des n queens est joué sur une grille $n \times n$, chaque pièce occupant une cellule. Une reine est une pièce d'échecs qui, dans un mouvement donné, peut se déplacer à n'importe quelle distance verticalement, horizontalement ou en diagonale. Cependant la reine ne peut pas se déplacer dans plus d'une direction par tour.

Dans mon travail j'ai utilisé deux fonction principaux :

- ✓ La première fonction intitulée Tester qui prend comme paramètre i, j qui sont les indices ligne , colonne cette fonction cherche s'il existe une reine ou la case est interdite en diagonales , lignes et colonnes la fonction retourne valeur boolean false si la case est libre , true si la case est interdite ou occupée
- ✓ La deuxième fonction intitulée Placer_reine qui prend comme paramètre la taille du matrice tout d'abord on teste si la taille de matrice est null donc la fonction prend true sinon on parcourt chaque case et on appelle la fonction Tester pour tester les diagonales et la ligne et la colonne , et on teste la case si elle est occupée déjà par une reine ou pas , si tout est bon la case prend '1' comme valeur et on appelle la même fonction Placer_reine et on affecte comme paramètre $n-1$ et la teste si elle est true sinon la case prend '0' et la fonction prend false

Algorithme :

Fonction Tester(int i , int j)

Debut

Pour k de 0 à n faire

Si $\text{chess}[i][k] == 1 \parallel \text{chess}[k][j] == 1$

return True

Pour k de 0 à n faire

Pour l de 0 à n faire

Si $(k+l == i+j)$ or $(k-l == i-j)$

Si $\text{chess}[k][l] == 1$:

return True

return False

Fin

Fonction Placer_reine(n)

Debut

```
Si n==0
    Return True
Pour i de 0 à n faire
    Pour j de 0 à n faire
        Si (not(Tester(i,j))) and (chess[i][j]!=1)
            Debut
                chess[i][j] = 1
                Si Placer_reine(n-1)==True
                    return True
                chess[i][j] = 0
            Fin Si
    Fin pour
Fin pour
Return false
Fin
Ecrire('Donner la taille du matrice')
Lire(n)
Int chess[n][n]
Placer_reine(n)
Pour i de 0 à n faire
    Pour j de 0 à n faire
        Ecrire(chess[i][j])
    Fin pour
Fin pour
```