```c
#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>
#include <iostream>
#include <cmath>

const int taille = 9 ;
int grille[taille][taille] =
  {{0,0,0,0,8,0,0,0,0},{0,6,0,0,0,9,0,0,5},{0,0,0,0,3,2,0,0,6},
   {0,4,0,5,0,0,0,0,0},{0,0,2,0,0,0,8,3,0},{0,0,0,0,0,0,0,0,0},
   {0,0,0,1,0,0,0,0,0},{0,0,3,0,0,0,2,1,0},{0,0,0,0,0,0,0,0,4}};
int numsol = 0 ;

void afficher() {int i, j ;

  for (i = 0; i < taille ; i++) {
    for (j = 0; j < taille ; j++)
      printf("%d ", grille[i][j]);
    std::cout << "\n" ;
  }
  std::cout << "----------\n" ;
}

bool absentSurLigne(int k, int i) {int j ;

  for (j = 0; j < taille ; j++)
    if (grille[i][j] == k)
      return false;
  return true;
}

bool absentSurColonne(int k, int j) {int i ;

  for (i = 0; i < taille ; i++)
    if (grille[i][j] == k)
      return false;
  return true;
}

bool absentSurBloc (int k, int i, int j)
{
  int mi = i-(i%3), mj = j-(j%3);  // ou encore : _i = 3*(i/3), _j = 3*(j/3);

  for (i = mi; i < mi+3; i++)
    for (j = mj; j < mj+3; j++)
      if (grille[i][j] == k)
      return false ;
  return true ;
}

bool estValide(int position) {int i, j, k ;

  if (position == taille * taille) {
    numsol++ ;
    std::cout << "Solution n°" << numsol << "\n" ;
    afficher() ;
  }
  i = position / taille ;
  j = position % taille ;
  if (grille[i][j] != 0) return estValide(position+1);
  for (k = 1; k <= taille ; k++)
    if (absentSurLigne(k,i) && absentSurColonne(k,j) && absentSurBloc(k,i,j)) {
      int garde = grille[i][j] ;
      grille[i][j] = k;
```

```c
      estValide(position+1) ;
      grille[i][j] = garde ;
    }
}

int main(void) {
  printf("Grille avant\n");
  afficher();
  estValide(0) ;
}
```