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

const int taille = 12 ;
int grille[taille][taille] = {{0,0,11,0,12,0,0,8,0,2,0,0},
                              {0,0,0,0,8,0,0,7,0,0,0,0},
                              {0,0,0,9,11,0,0,4,6,0,0,0},
                              {0,12,1,0,0,2,10,0,0,3,4,0},
                              {12,0,0,10,0,0,0,0,3,0,0,2},
                              {0,2,3,0,0,0,0,0,0,9,11,0},
                              {4,0,0,0,0,7,6,0,0,0,0,5},
                              {9,7,0,5,0,0,0,0,1,0,8,3},
                              {8,0,0,11,0,1,5,0,9,0,0,7},
                              {0,0,2,12,0,0,0,0,4,11,0,0},
                              {0,0,9,0,0,0,0,0,0,1,0,0},
                              {11,10,0,0,0,0,0,0,0,0,3,4}};;
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 ri, rj, mi, mj, l, m ;

  rj = j%12 ; if (rj < 3) mj = 0 ; else if (rj < 6) mj = 3 ; else if (rj < 9) mj
= 6 ; else mj = 9 ;
  ri = i%12 ; if (ri < 4) mi = 0 ; else if (ri < 8) mi = 4 ; else mi = 8  ;
  for (l = mi; l < mi+4 ; l++)
    for (m = mj; m < mj+3 ; m++)
      if (grille[l][m] == 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;
      estValide(position+1) ;
      grille[i][j] = garde ;
    }
}

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