```cpp
#include<iostream>
using namespace std;
void afiseazaGrila(int mat[9][9], const char mesaj[])
{
	cout << mesaj << endl;
	for (int i = 0; i < 9; i++)
	{
		if (i % 3 == 0) cout << endl;
		for (int j = 0; j < 9; j++)
		{
			if (j % 3 == 0) cout << " ";
			int n = mat[i][j];
			if (n == 0) cout << "- ";
			else cout << n << " ";
		}
		cout << endl;
	}
	cout << endl;
}
bool esteAdmisibila(int mat[9][9])
{	//verificam valorile
	for (int i = 0; i < 9; i++) {
		for (int j = 0; j < 9; j++) {
			int n = mat[i][j];
			if (n < 0 || n>10) return false;
		}
	}
	//verificam liniile
	for (int i = 0; i < 9; i++) {
		bool estePrezent[10] = {};
		for (int j = 0; j < 9; j++) {
			int n = mat[i][j];
			if (n != 0 && estePrezent[n]) return false;
			estePrezent[n] = true;
		}
	}
	//verificam coloanele
	for (int j = 0; j < 9; j++) {
		bool estePrezent[10] = {};
		for (int i = 0; i < 9; i++) {
			int n = mat[i][j];
			if (n != 0 && estePrezent[n]) return false;
			estePrezent[n] = true;
		}
	}
	//verificam celulele
	for (int icel = 0; icel < 9; icel += 3) {
		for (int jcel = 0; jcel < 9; jcel += 3) {
			bool estePrezent[10] = {};
			for (int i = icel; i < icel + 3; i++) {
				for (int j = jcel; j < jcel + 3; j++) {
					int n = mat[i][j];
					if (n != 0 && estePrezent[n]) return false;
					estePrezent[n] = true;
				}
			}
		}
	}
	return true;
}
```

```
bool esteValid(int mat[9][9], int i, int j, int n)
{
	for (int h = 0; h < 9; h++)
	{
		if (h != i && mat[h][j] == n) return false;//mai este un n pe linie
		if (h != j && mat[i][h] == n) return false;//mai este un n pe coloana
	}
	int icel = i - i % 3;
	int jcel = j - j % 3;
	for (int hi = icel; hi < icel + 3; hi++)
	{
		for (int hj = jcel; hj < jcel + 3; hj++)
		{
			if (i != hi && j != hj && mat[hi][hj] == n) return false;
			                                                  //mai este un n in celula
		}
	}
	return true;
}


int formeazaTraseu(int mat[9][9], int itab[81], int jtab[81])
{   //formam traseul celulelor goale
	int k = -1;
	for (int i = 0; i < 9; i++)
	{
		for (int j = 0; j < 9; j++)
		{
			if (mat[i][j] == 0)
			{
				k++;
				itab[k] = i;
				jtab[k] = j;
			}
		}
	}
	return k;     //ultimul indice
}


void rezolvaGrila(int mat[9][9])
{
	//Backtracking

	int itab[81] = {};
	int jtab[81] = {};
	int kmax = formeazaTraseu(mat, itab, jtab);
	if (kmax < 0) return;
	int k = 0;
	//k==kmax inseamna ca am gasit o solutie
	int contorSolutii = 0;
	do
	{
		int ik = itab[k], jk = jtab[k], n = 0;
		//cautam urmatorul candidat valid pe locul [ik,jk]
		do
		{
			n = ++mat[ik][jk];
		} while (n <= 9 && !esteValid(mat, ik, jk, n));
```

```
		if (n > 9) //nu mai exista candidati, am epuizat locul
		{
			mat[ik][jk] = 0;    //presetam la zero
			k--;                //ne retragem un pas
		}
		else if (k == kmax) //avem o solutie, o afisam
		{
			afiseazaGrila(mat, "Solutie:");
			contorSolutii++;
			//break;
		}
		else k++;           //avem un candidat valid, avansam

	} while (k >= 0);
	cout << "Am gasit " << contorSolutii << " solutii. Salut!" << endl;
}
int main() {
	int mat[9][9] =
	{
		{ 1,2,0,6,0,0,0,8,0},
		{ 6,0,0,9,0,0,5,7,0},
		{ 0,5,0,0,0,0,0,0,0},
		{ 9,0,5,1,0,0,0,0,0},
		{ 7,1,2,8,0,0,0,0,3},
		{ 0,0,0,0,0,7,2,0,0},
		{ 0,0,0,0,7,0,3,6,5},
		{ 5,8,0,0,0,0,0,0,7},
		{ 0,7,0,2,5,0,0,9,4},
	};
	afiseazaGrila(mat, "Grila initiala");
	if (!esteAdmisibila(mat)) {
		cout << "Grila nu este admisibila!" << endl;
		return 0;
	}
	rezolvaGrila(mat);
	return 0;
}






/*
Grila initiala

 1 2 -  6 - -  - 8 -
 6 - -  9 - -  5 7 -
 - 5 -  - - -  - - -

 9 - 5  1 - -  - - -
 7 1 2  8 - -  - - 3
 - - -  - - 7  2 - -

 - - -  - 7 -  3 6 5
 5 8 -  - - -  - - 7
 - 7 -  2 5 -  - 9 4
```

```
Solutie:

 1 2 7  6 3 5  4 8 9
 6 3 8  9 1 4  5 7 2
 4 5 9  7 8 2  6 3 1

 9 6 5  1 2 3  7 4 8
 7 1 2  8 4 6  9 5 3
 8 4 3  5 9 7  2 1 6

 2 9 1  4 7 8  3 6 5
 5 8 4  3 6 9  1 2 7
 3 7 6  2 5 1  8 9 4

Solutie:

 1 2 7  6 3 5  4 8 9
 6 4 3  9 8 2  5 7 1
 8 5 9  7 1 4  6 3 2

 9 6 5  1 2 3  7 4 8
 7 1 2  8 4 6  9 5 3
 4 3 8  5 9 7  2 1 6

 2 9 1  4 7 8  3 6 5
 5 8 4  3 6 9  1 2 7
 3 7 6  2 5 1  8 9 4

Am gasit 2 solutii. Salut!

Press any key to close this window . . .

*/
```