

```

1  const int MAXN = 10;
2
3  int N,M;
4  bool griglia[MAXN][MAXN];
5
6  // Controllo se posso mettere la x in griglia[r][c]:
7  bool Posso(int r, int c) {
8      // Tris orizzontale:
9      if (c-2>=0 && griglia[r][c-2] && griglia[r][c-1])
10         return false;
11      // Tris verticale:
12      if (r-2>=0 && griglia[r-2][c] && griglia[r-1][c])
13         return false;
14      // Tris diagonale in alto a sx:
15      if (r-2>=0 && c-2>=0 && griglia[r-2][c-2] && griglia[r-1][c-1])
16         return false;
17      if (r-2>=0 && c+2<M && griglia[r-2][c+2] && griglia[r-1][c+1])
18         return false;
19      return true;
20 }
21
22 int Riempi(int riga, int col) {
23     // Caso base: griglia finita
24     if (riga >= N) return 0;
25
26     int nuova_riga = (col + 1 == M) ? riga + 1 : riga;
27     int nuova_col = (col + 1) % M;
28
29     // Non metto la X:
30     int senza_x = Riempi(nuova_riga,nuova_col);
31     int con_x = 0;
32
33     // Se posso, metto la X:
34     if (Posso(riga,col)) {
35         griglia[riga][col]=true;
36         con_x = 1 + Riempi(nuova_riga,nuova_col);
37         // Reset della griglia[riga][col]:
38         griglia[riga][col]=false;
39     }
40     // Ritorno la soluzione migliore:
41     return max(senza_x,con_x);
42 }

```