

```

1  const int MAXR = 10;
2  const int MAXC = 6;
3  int N, M;
4  int sol[MAXR][MAXC][1 << MAXC][1 << MAXC][1 << MAXC];
5
6  // Controllo se nella riga scelta è presente una X nella posizione pos:
7  bool Attivo(int pos, int bitmask_riga) {
8      // Casella non esiste:
9      if (pos < 0 || pos >= M) return false;
10     return (bitmask_riga & (1 << pos));
11 }
12 // Controllo se posso inserire nella casella [riga][col] una X senza creare tris
13 bool Posso(int riga, int col, int bp, int bm, int bo) {
14     // Tris orizzontale:
15     if (Attivo(col - 2, bo) && Attivo(col - 1, bo)) return false;
16     // Tris verticale:
17     if (Attivo(col, bm) && Attivo(col, bp)) return false;
18     // Tris in diagonale in alto a sx:
19     if (Attivo(col - 1, bm) && Attivo(col - 2, bp)) return false;
20     // Tris in diagonale in alto a dx:
21     if (Attivo(col + 1, bm) && Attivo(col + 2, bp)) return false;
22     return true;
23 }
24 /*
25  * bp = Bitmask_prima --> Riga prima della precedente
26  * bm = Bitmask_medio --> Riga precedente
27  * bo = Bitmask_ora --> Riga Attuale
28 */
29 int Riempi(int riga, int col, int bp, int bm, int bo) {
30     // Caso base: griglia finita
31     if (riga >= N) return 0;
32     // DP: ho già calcolato questa soluzione:
33     if (sol[riga][col][bp][bm][bo] != -1) return sol[riga][col][bp][bm][bo];
34
35     int nuova_riga = (col + 1 == M) ? riga + 1 : riga;
36     int nuova_col = (col + 1) % M;
37
38     // Aggiorno i bitmask delle righe: nel caso in cui cambio riga devo "mandare indietro" di 1
39     // tutti i bitmask poiché la riga attuale diventa la precedente e così via,
40     // altrimenti lascio quelli attuali:
41
42     int nuovo_bp = ((nuova_riga==riga) ? bp : bm);
43     int nuovo_bm = ((nuova_riga==riga) ? bm : bo);
44     int nuovo_bo = ((nuova_riga==riga) ? bo : 0);
45
46     // Non metto la X
47     int senza_x = Riempi(nuova_riga, nuova_col, nuovo_bp , nuovo_bm , nuovo_bo);
48     int con_x = 0;
49
50     // Se posso, metto la X
51     if (Posso(riga, col, bp, bm, bo)) {
52         // Ri-Aggiorno le righe in modo simile a prima, con la differenza
53         // che riempio la casella "col" della riga attuale,
54         // cioè di bo (bitmask riga attuale):
55         nuovo_bp = ((nuova_riga==riga) ? bp : bm);
56         nuovo_bm = ((nuova_riga==riga) ? bm : (bo | (1 << col)) );
57         nuovo_bo = ((nuova_riga==riga) ? (bo | (1 << col)) : 0 );
58         con_x = 1 + Riempi(nuova_riga, nuova_col, nuovo_bp , nuovo_bm , nuovo_bo);
59     }
60     // Ritorno e salvo il risultato migliore:
61     return sol[riga][col][bp][bm][bo] = max(senza_x, con_x);
62 }

```