

## Задача БРОЕНЕ

**Пояснение към решението** – файл `count_dp_opt_100p.cpp`

В `dp[i][j]` поддържа броя на пътищата по които може да стигнем до клетката с координати  $(i, j)$  последователно за стъпките  $k = 1, 2, \dots, ns$ . За  $k=1$  всичките стойности в масива са равни на 1.

Масивът `dpc[][]` използваме като спомагателен масив, за да пресметнем броя на пътищата при преминаване от стойност на  $k$  към следващата. Това се извършва в тялото на цикъла с индекс  $k$  чрез фрагмента:

```
for (int i = 0; i < m; i++)
for (int j = 0; j < n; j++)
{
    dpc[i][j]=0;
    for(int t=0; t<4; t++)
    {
        int it=i+di[t]; int jt=j+dj[t];
        if(0<=it && it<m && 0<=jt && jt<n)
            dpc[i][j] = (dpc[i][j]%MOD+dp[it][jt]%MOD)%MOD;
    }
}
```

В края на тялото на цикъла с индекс  $k$  пренасяме стойностите от масива `dpc[][]` в масива `dp[][]`. Накрая на програмата сумираме всички стойности на масива `dp[][]` и отпечатваме резултата. Всички пресмятания се извършват по модул `MOD`.

*Емил Келеведжиев*