

یک جدول درهم تشکیل شده از  $m = 11$  خانه (*slot*) را در نظر بگیرید و فرض کنید که کلیدها (که مقادیری صحیح و نامنفی‌اند) به وسیله تابع درهم ساز (*hash function*) زیر به خانه های جدول *map* شده اند:

```
int h1(int key)
{
    int x = (key + 7) * (key + 7);
    x = x / 16;
    x = x + key;
    x = x % 11;
    return x;
}
```

یک راه مقابله با مشکل *collision* در جداول درهم پیاده سازی *open addressing* است. فرض کنید برای پیاده سازی *open addressing* بخواهیم از کاوش درجه دو استفاده کنیم و از تابع کاوش (*probe function*) زیر استفاده کنیم:

$$\frac{x^2 + x}{2}$$

در این صورت مکان هر کلید با تابع زیر مشخص می‌شود:

$$h(k, i) = (h1(k) + \frac{i^2 + i}{2}) \bmod m$$

کلیدهای داده شده به ترتیب داده شده وارد جدول درهم کنید و اطلاعات خواسته شده در جدول زیر را کامل کنید و در آخر مشخص کنید در جدول نهایی داخل هر *slot* چه کلیدی قرار گرفته است.

Key	Home slot	Probe sequence
43	1	0
23	2	0
1	5	0
0	3	0
15	4	2
31	0	0
4	6	3
7	8	0
11	9	0
3	10	1

Final Hash-Map >> 31 | 43 | 23 | 0 | 15 | 1 | 4 | \_ | 7 | 11 | 3