# Part 1

Construct a new method of probing where each probe sequence has a different sequence of jumps, as a function of h1(k), but still searches the entire hash table.

**New Method:** $h(k, i) = (h1(k) + f(i)) \bmod M$

where:

- k is the key,
- h1(k) is the initial hash value,
- f(i) is the jump sequence,
- M is the size of the hash table. (Looking ahead to part 2, we know M will be a million)
- $f(i) = i^2 + 7 * i$

**Motivation**

The proposed probing method combines the advantages of quadratic probing and a linear offset to minimize clustering and avoid long contiguous probes. Quadratic jumps with $i^2$ help distribute probe attempts across the table, reducing primary and secondary clustering.

Choosing 7 (which is co-prime to 1,000,000) ensures that probe sequences will not cycle prematurely. Ensuring M and 7 are co-prime, the entire has table is guaranteed to be searched.

Choosing 7 keeps the computations within a manageable range, avoiding the computational overhead associated with higher values. It remains efficient even for large hash tables like M=1,000,000 because the operations involved (squares and multiplications) are straightforward enough to be executed efficiently on modern processors.

When the hash table is partially filled, the probing sequence must eventually probe every slot to ensure that there is no "gap" where a key can’t be inserted. Since $i^2+7 \cdot i$ produces unique values modulo M for each i, every index will be visited at some point in the sequence, ensuring that the search will cover the entire table.