We will calculate h(i)h(i)h(i) for each key and place it in the appropriate position in the table.

1. **Key = 12**

h(12)=(2⋅12+5)mod 11=(24+5)mod 11=29mod 11=7h(12) = (2 \cdot 12 + 5) \mod 11 = (24 + 5) \mod 11 = 29 \mod 11 = 7h(12)=(2⋅12+5)mod11=(24+5)mod11=29mod11=7

Place **12** at index **7**.

2. **Key = 44**

h(44)=(2⋅44+5)mod 11=(88+5)mod 11=93mod 11=5h(44) = (2 \cdot 44 + 5) \mod 11 = (88 + 5) \mod 11 = 93 \mod 11 = 5h(44)=(2⋅44+5)mod11=(88+5)mod11=93mod11=5

Place **44** at index **5**.

3. **Key = 13**

h(13)=(2⋅13+5)mod 11=(26+5)mod 11=31mod 11=9h(13) = (2 \cdot 13 + 5) \mod 11 = (26 + 5) \mod 11 = 31 \mod 11 = 9h(13)=(2⋅13+5)mod11=(26+5)mod11=31mod11=9

Place **13** at index **9**.

4. **Key = 88**

h(88)=(2⋅88+5)mod 11=(176+5)mod 11=181mod 11=5h(88) = (2 \cdot 88 + 5) \mod 11 = (176 + 5) \mod 11 = 181 \mod 11 = 5h(88)=(2⋅88+5)mod11=(176+5)mod11=181mod11=5

Place **88** at index **5** (in a chain with **44**).

5. **Key = 23**

h(23)=(2⋅23+5)mod 11=(46+5)mod 11=51mod 11=7h(23) = (2 \cdot 23 + 5) \mod 11 = (46 + 5) \mod 11 = 51 \mod 11 = 7h(23)=(2⋅23+5)mod11=(46+5)mod11=51mod11=7

Place **23** at index **7** (in a chain with **12**).

6. **Key = 94**

h(94)=(2⋅94+5)mod 11=(188+5)mod 11=193mod 11=6h(94) = (2 \cdot 94 + 5) \mod 11 = (188 + 5) \mod 11 = 193 \mod 11 = 6h(94)=(2⋅94+5)mod11=(188+5)mod11=193mod11=6

Place **94** at index **6**.

7. **Key = 11**

h(11)=(2⋅11+5)mod 11=(22+5)mod 11=27mod 11=5h(11) = (2 \cdot 11 + 5) \mod 11 = (22 + 5) \mod 11 = 27 \mod 11 = 5h(11)=(2⋅11+5)mod11=(22+5)mod11=27mod11=5

Place **11** at index **5** (in a chain with **44** and **88**).

8. **Key = 39**

h(39)=(2⋅39+5)mod 11=(78+5)mod 11=83mod 11=6h(39) = (2 \cdot 39 + 5) \mod 11 = (78 + 5) \mod 11 = 83 \mod 11 = 6h(39)=(2⋅39+5)mod11=(78+5)mod11=83mod11=6

Place **39** at index **6** (in a chain with **94**).

9. **Key = 20**

h(20)=(2⋅20+5)mod 11=(40+5)mod 11=45mod 11=1h(20) = (2 \cdot 20 + 5) \mod 11 = (40 + 5) \mod 11 = 45 \mod 11 = 1h(20)=(2⋅20+5)mod11=(40+5)mod11=45mod11=1

Place **20** at index **1**.

10. **Key = 16**

h(16)=(2⋅16+5)mod 11=(32+5)mod 11=37mod 11=4h(16) = (2 \cdot 16 + 5) \mod 11 = (32 + 5) \mod 11 = 37 \mod 11 = 4h(16)=(2⋅16+5)mod11=(32+5)mod11=37mod11=4

Place **16** at index **4**.

11. **Key = 5**

h(5)=(2⋅5+5)mod 11=(10+5)mod 11=15mod 11=4h(5) = (2 \cdot 5 + 5) \mod 11 = (10 + 5) \mod 11 = 15 \mod 11 = 4h(5)=(2⋅5+5)mod11=(10+5)mod11=15mod11=4

Place **5** at index **4** (in a chain with **16**).

Final Hash Table

Index	Keys
0	
1	20
2	
3	
4	16 → 5
5	44 → 88 → 11
6	94 → 39
7	12 → 23
8	
9	13
10	

//////R2.20

function as before: h(i)=(2i+5)mod 11h(i) = (2i + 5) \mod 11h(i)=(2i+5)mod11.

1. Key = 12
   h(12)=(2·12+5)mod 11=7h(12) = (2 \cdot 12 + 5) \mod 11 = 7h(12)=(2·12+5)mod11=7
   Place 12 at index 7.
2. Key = 44
   h(44)=(2·44+5)mod 11=5h(44) = (2 \cdot 44 + 5) \mod 11 = 5h(44)=(2·44+5)mod11=5
   Place 44 at index 5.
3. Key = 13
   h(13)=(2·13+5)mod 11=9h(13) = (2 \cdot 13 + 5) \mod 11 = 9h(13)=(2·13+5)mod11=9
   Place 13 at index 9.
4. Key = 88
   h(88)=(2·88+5)mod 11=5h(88) = (2 \cdot 88 + 5) \mod 11 = 5h(88)=(2·88+5)mod11=5
   Collision at index 5 (occupied by 44), so try 6.
   Place 88 at index 6.
5. Key = 23
   h(23)=(2·23+5)mod 11=7h(23) = (2 \cdot 23 + 5) \mod 11 = 7h(23)=(2·23+5)mod11=7
   Collision at index 7 (occupied by 12), so try 8.
   Place 23 at index 8.
6. Key = 94
   h(94)=(2·94+5)mod 11=6h(94) = (2 \cdot 94 + 5) \mod 11 = 6h(94)=(2·94+5)mod11=6
   Collision at index 6 (occupied by 88), so try 7, 8, 9, 10.
   Place 94 at index 10.
7. Key = 11
   h(11)=(2·11+5)mod 11=5h(11) = (2 \cdot 11 + 5) \mod 11 = 5h(11)=(2·11+5)mod11=5
   Collision at index 5, so try 6, 7, 8, 9, 10, 0.
   Place 11 at index 0.

8. Key = 39
   h(39)=(2·39+5)mod 11=6h(39) = (2 \cdot 39 + 5) \mod 11 =
   6h(39)=(2·39+5)mod11=6
   Collision at index 6, so try 7, 8, 9, 10, 0, 1.
   Place 39 at index 1.

9. Key = 20
   h(20)=(2·20+5)mod 11=1h(20) = (2 \cdot 20 + 5) \mod 11 =
   1h(20)=(2·20+5)mod11=1
   Collision at index 1, so try 2.
   Place 20 at index 2.

10. Key = 16
   h(16)=(2·16+5)mod 11=4h(16) = (2 \cdot 16 + 5) \mod 11 =
   4h(16)=(2·16+5)mod11=4
   Place 16 at index 4.

11. Key = 5
   h(5)=(2·5+5)mod 11=4h(5) = (2 \cdot 5 + 5) \mod 11 = 4h(5)=(2·5+5)mod11=4
   Collision at index 4, so try 5, 6, 7, 8, 9, 10, 0, 1, 2, 3.
   Place 5 at index 3.

//////////////////2.21: Quadratic Probing

For quadratic probing, when a collision occurs at index iii, we probe at i+12i + 1^2i+12, i+22i + 2^2i+22, etc., modulo the table size (11). If we encounter a collision, we continue with a different quadratic offset. Let's use the same hash function and keys:

1. Key = 12 at index 7.

2. Key = 44 at index 5.

3. Key = 13 at index 9.

4. Key = 88 at index 5 (collision). Try 5+12=65 + 1^2 = 65+12=6.

5. Key = 23 at index 7 (collision). Try 7+12=87 + 1^2 = 87+12=8.

6. Key = 94 at index 6 (collision). Try 6+12=76 + 1^2 = 76+12=7 (collision). Try 6+22=106 + 2^2 = 106+22=10.

7. Key = 11 at index 5 (collision).

///////////pseudocode for Removal with Linear Probing

Algorithm remove(key)

Input: key to be removed

Output: None (updates the hash table in place)

```
 index := hashFunction(key)
 while table[index] ≠ empty do
    if table[index] = key then
       table[index] := DELETED_MARKER
       return
    index := (index + 1) mod tableSize
end while
// If we reach here, key was not found in the table
return "Key not found"
```