1. Consider a hash table consisting of $M = 11$ slots, and suppose nonnegative integer key values are hashed into the table using the hash function `h1()`:

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

a) [10 points] Suppose that collisions are resolved by using linear probing. The integer key values listed below are to be inserted, in the order given. Show the home slot (the slot to which the key hashes, before any probing), the probe sequence (if any) for each key, and the final contents of the hash table after the following key values have been inserted in the given order:

Key Value	Home Slot	Probe Sequence
43	1	
23	2	
1	5	
0	3	
15	1	2, 3, 4
31	0	
4	0	1, 2, 3, 4, 5, 6
7	8	
11	9	
3	9	10

Final Hash Table:

Slot	0	1	2	3	4	5	6	7	8	9	10
Contents	31	43	23	0	15	1	4		7	11	3

Detailed Solution Description

To solve the hash table problem using the given hash function and linear probing, we follow these steps:

1. Understand the Hash Function

The hash function h1h1h1 computes the hash value as follows:

1. Compute (key+7)×(key+7)(key + 7) \times (key + 7)(key+7)×(key+7).
2. Divide the result by 16.
3. Add the key to the result.
4. Take the result modulo 11 (the number of slots).

2. Compute the Home Slots

We calculate the home slot for each key using the hash function.

Key: 43

1. (43+7)×(43+7)=50×50=2500(43 + 7) \times (43 + 7) = 50 \times 50 = 2500(43+7)×(43+7)=50×50=2500
2. 2500/16=1562500 / 16 = 1562500/16=156
3. 156+43=199156 + 43 = 199156+43=199
4. 199%11=1199 \% 11 = 1199%11=1

Home Slot: 1

Key: 23

1. (23+7)×(23+7)=30×30=900(23 + 7) \times (23 + 7) = 30 \times 30 = 900(23+7)×(23+7)=30×30=900
2. 900/16=56900 / 16 = 56900/16=56
3. 56+23=7956 + 23 = 7956+23=79
4. 79%11=279 \% 11 = 279%11=2

Home Slot: 2

Key: 1

1. (1+7)×(1+7)=8×8=64(1 + 7) \times (1 + 7) = 8 \times 8 = 64(1+7)×(1+7)=8×8=64
2. 64/16=464 / 16 = 464/16=4
3. 4+1=54 + 1 = 54+1=5
4. 5%11=55 \% 11 = 55%11=5

Home Slot: 5

Key: 0

1. (0+7)×(0+7)=7×7=49(0 + 7) \times (0 + 7) = 7 \times 7 = 49(0+7)×(0+7)=7×7=49
2. 49/16=349 / 16 = 349/16=3
3. 3+0=33 + 0 = 33+0=3
4. 3%11=33 \% 11 = 33%11=3

Home Slot: 3

Key: 15

1. (15+7)×(15+7)=22×22=484(15 + 7) \times (15 + 7) = 22 \times 22 = 484(15+7)×(15+7)=22×22=484
2. 484/16=30484 / 16 = 30484/16=30
3. 30+15=4530 + 15 = 4530+15=45
4. 45%11=145 \% 11 = 145%11=1

Home Slot: 1 (Collision with 43)

Probe Sequence: 1, 2, 3, 4

Key: 31

1. (31+7)×(31+7)=38×38=1444(31 + 7) \times (31 + 7) = 38 \times 38 = 1444(31+7)×(31+7)=38×38=1444
2. 1444/16=901444 / 16 = 901444/16=90
3. 90+31=12190 + 31 = 12190+31=121
4. 121%11=0121 \% 11 = 0121%11=0

Home Slot: 0

Key: 4

1. (4+7)×(4+7)=11×11=121(4 + 7) \times (4 + 7) = 11 \times 11 = 121(4+7)×(4+7)=11×11=121
2. 121/16=7121 / 16 = 7121/16=7
3. 7+4=117 + 4 = 117+4=11
4. 11%11=011 \% 11 = 011%11=0

Home Slot: 0 (Collision with 31)

Probe Sequence: 0, 1, 2, 3, 4, 5, 6

Key: 7

1. (7+7)×(7+7)=14×14=196(7 + 7) \times (7 + 7) = 14 \times 14 = 196(7+7)×(7+7)=14×14=196
2. 196/16=12196 / 16 = 12196/16=12
3. 12+7=1912 + 7 = 1912+7=19
4. 19%11=819 \% 11 = 819%11=8

Home Slot: 8

Key: 11

1. (11+7)×(11+7)=18×18=324(11 + 7) \times (11 + 7) = 18 \times 18 = 324(11+7)×(11+7)=18×18=324
2. 324/16=20324 / 16 = 20324/16=20
3. 20+11=3120 + 11 = 3120+11=31
4. 31%11=931 \% 11 = 931%11=9

Home Slot: 9

Key: 3

1. (3+7)×(3+7)=10×10=100(3 + 7) \times (3 + 7) = 10 \times 10 = 100(3+7)×(3+7)=10×10=100
2. 100/16=6100 / 16 = 6100/16=6
3. 6+3=96 + 3 = 96+3=9
4. 9%11=99 \% 11 = 99%11=9

Home Slot: 9 (Collision with 11)

Probe Sequence: 9, 10

3. Resolve Collisions Using Linear Probing

When a collision occurs (i.e., the home slot is already occupied), we use linear probing to find the next available slot. The probe sequence for each key shows the steps taken to resolve collisions.

4. Final Hash Table

After inserting all keys and resolving collisions, the final hash table is:

Slot	0	1	2	3	4	5	6	7	8	9	10
	31	43	23	0	15	1	4	7	11	3	

This table shows the final positions of the keys after inserting them using the provided hash function and resolving any collisions with linear probing.