Hash Tables

Modify your hash table implementation for following specifications

1. No duplicates allowed.

2. If the load factor exceeds the given threshold, then triple the number of buckets.

Assume that the hash table stores only integer valued keys.

Load factor = (Number of keys stored in the hash table) / (Number of buckets in the hash table)

Hash function to be used: Remainder after dividing the given key by the number of buckets.

Whenever we try to insert a duplicate key, display the status of the hash table.

Consider the following example input file

1.5

2

15

12

90

90

60

5

105

33

40

64

12

The first line of the input file specifies the load factor 1.5. Note that this is a real number.

The second line specifies initial number of buckets, two. You should begin with an empty hash table that has two buckets.

Your initial hash function will be remainder after dividing by the number of buckets.

From the third line onwards, the input file specifies various integer keys to be stored in the hash table. First three insert operations for keys: 15, 12, and 90 should work like the static hash table.

The fourth insert request is for key 90. It will cause a duplicate insertion. Do not store the duplicate. However display an error message, followed by the status of the hash table. Use the following format.

```
Key 90 already present in the hash table. Status of the hash table.
Number of keys: 3
Number of buckets: 2
Bucket 0:   12, 90
Bucket 1:  15
```

Next insert request is for key 60. This key is not present in the hash table. However, it cause the load factor to go beyond 1.5. Now we will triple the number of buckets to six. We will rehash all existing keys and the new key to be inserted. With six buckets, we can store up to nine keys in the hash table as allowed load factor is 1.5. Note that there is no restriction about number of keys per bucket. We are concerned only about the total number of keys in the hash table.

Next insertion requests for keys 5, 105, 33, 40, and 64 should work like the static hash table.

Next insertion request is for key 12 which is already present in the hash table. You should display the error and status messages.

```
Key 12 already present in the hash table. Status of the hash table.
Number of keys: 9
Number of buckets: 6
Bucket 0:   12, 90, 60
Bucket 1:
Bucket 2:
Bucket 3:  15, 105, 33
Bucket 4:  40, 64
Bucket 5:  5
```

How your code should compile.

```
> g++   main.cpp
```

How your code should run

```
> ./a.out <input.txt >output.txt
```