Assignment 8

Data Structure Lab

Date: November 10, 2021 **Time**: 2:30 P.M. – 5:30 P.M.

In this assignment, you maintain a hash table H of student records. Each student record contains a nine-letter roll number (to be stored as a string) and a floating-point CGPA. Hashing is done with respect to the roll numbers. Let s denote the initial size of the hash table, and n the number of records currently present in the table. The load factor $\lambda = n/s$ does not exceed 0.5. Let d denote the number of deleted cells in the table. In order to keep searches efficient, we will also demand that the fraction of deleted cells, that is, $\delta = d/s$, would never exceed 0.2.

H should be a hash table that uses quadratic probing as a collision resolution strategy. Since $\lambda < 0.5$, it is ensured that every insertion is always successful. The i-th hash of a roll number $R = d_0d_1D_2D_3d_4d_5d_6d_7d_8$ is the integer:

$$h(R, i) = ((d_0d_1d_4d_5d_6d_7d_8)_{10} + IDX(D_2D_3) + i^2)\%s$$

.

All the digits in the roll number are concatenated to form a seven-digit integer in the decimal notation. The contribution of the department equals the index (to be obtained by binary search) of D_2D_3 in the following string array:

char DEPT[23][3] = {"AE", "AG", "AR", "BT", "CE", "CH", "CS", "CY", "EC", "EE", "EX", "GG", "HS", "IE", "IM", "MA", "ME", "MF", "MI", "MT", "NA", "PH", "QD" };

The allocation size s of the hash table is always a prime and may change dynamically. When λ exceeds 0.5 (after an insertion) or δ exceeds 0.2 (after a deletion), rehashing is to be performed. Use special CGPA values to mark EMPTY and DELETED cells (like –1 and –2).

Write a program to do the following.

1. Read the initial number k of insertions from the user.
2. Create an empty hash table H of size s = nextprime(2k). You need to write a function for recognizing primes.
3. The user then enters k (Roll Number, CGPA) pairs. These are inserted one by one in H. After the k initial insertions, print H (printing format is given on the next page).
4. The program then enters a loop. In each iteration, the user does one of the following four operations:

   (a) Search(R): The user supplies a roll number R. Find out whether the roll number R is present in H. If so, the CGPA of the student with roll number R is (returned and) printed. Otherwise, print that the roll number does not exist in H.

(b) Insert(R,C): The user supplies a roll number R and a CGPA C. Insert the record (R,C) in H. If R is already present in H, then no change in made. If not, insertion is made in an empty or deleted cell following the usual insertion procedure. If insertion happens in a deleted cell, then the number d of deleted cells reduces by one. Moreover, if the load factor λ exceeds 0.5, then a rehashing is made. During a rehashing, a new empty hash table of size nextprime(2n) is created. The original hash table is scanned, and every record present in it is inserted one by one in the new hash table. Remove the old hash table, and rename the new hash table as H. After rehashing, the load factor is restored to a value less than 0.5. Moreover, the count d automatically goes to zero, since the new table has not experienced any deletion yet. Print the updated hash table (after rehashing, if done).

(c) Delete(R): The user supplies a roll number R. If R is not present in H, then no change is made. Otherwise, the standard deletion procedure is followed. Since a successful deletion increases the number of DELETED cells by one, it is necessary to check whether δ exceeds 0.2. If so, rehashing should be done. Use the same rehashing procedure as during insertions. The only difference is that the size of the new hash table will be kept the same as the size of the old hash table. Print the updated hash table (after rehashing, if done).

(d) Quit: Break the loop and exit the program.

Write the functions in the following order: nextprime(), createEmptyHashTable(), search(), insert(), delete(), rehash().

Sample Output

```
+++ Insert (08QD82110, 5.02)
+++ Insert (09AE52345, 7.61)
+++ Insert (11ME82353, 7.81)
s = 7, n = 3, d = 0
Entry 0: 11ME82353, 7.81
Entry 1: EMPTY
Entry 2: 09AE52345, 7.61
Entry 3: EMPTY
Entry 4: EMPTY
Entry 5: EMPTY
Entry 6: 08QD82110, 5.02

+++ Delete 08QD82110
s = 7, n = 2, d = 1
Entry 0: 11ME82353, 7.81
Entry 1: EMPTY
Entry 2: 09AE52345, 7.61
Entry 3: EMPTY
Entry 4: EMPTY
Entry 5: EMPTY
Entry 6: DELETED

+++ Insert (08MT61908, 6.17)
s = 7, n = 3, d = 1
```

```
Entry 0: 11ME82353, 7.81
Entry 1: EMPTY
Entry 2: 09AE52345, 7.61
Entry 3: 08MT61908, 6.17
Entry 4: EMPTY
Entry 5: EMPTY
Entry 6: DELETED

+++ Delete 09AE52345
+++ Rehashing... Please wait...
s = 7, n = 2, d = 0
Entry 0: EMPTY
Entry 1: EMPTY
Entry 2: EMPTY
Entry 3: 08MT61908, 6.17
Entry 4: EMPTY
Entry 5: EMPTY
Entry 6: 11ME82353, 7.81

+++ Insert (00BT10044, 5.17)
s = 7, n = 3, d = 0
Entry 0: EMPTY
Entry 1: EMPTY
Entry 2: 00BT10044, 5.17
Entry 3: 08MT61908, 6.17
Entry 4: EMPTY
Entry 5: EMPTY
Entry 6: 11ME82353, 7.81

+++ Insert (11BT30825, 5.98)
+++ Rehashing... Please wait...
s = 11, n = 4, d = 0
Entry 0: 08MT61908, 6.17
Entry 1: 11ME82353, 7.81
Entry 2: EMPTY
Entry 3: EMPTY
Entry 4: 00BT10044, 5.17
Entry 5: EMPTY
Entry 6: 11BT30825, 5.98
Entry 7: EMPTY
Entry 8: EMPTY
Entry 9: EMPTY
Entry 10: EMPTY

+++ Search(00BT10044)
The CGPA of 00BT10044 is 5.17

+++ Search(07IE66982)
Roll number 07IE66982 does not exist
```