

Hash Tables (Assignment 7 (CS142))

In this assignment we will build a rudimentary student registration system. At the time of admission of students in the institute, roll numbers are assigned to the registered students and the details of the students are stored in a database. The details for a student are maintained in the database as long as the student remains in the institute and may also need to be periodically updated, and when the student leaves the institute, this record may be archived or deleted.

We will build a hash table data structure that maintains this information. Let's assume the institute issues a 3-digit roll number to each student. Let us assume that there will be at most 20 students at any time in the database. For this purpose we will build a hash table of size 10. We will resolve collisions using chaining. Each of the buckets in the hash table will contain a pointer to a linked list. Each node of the linked list will contain the details for a particular student, say name and roll no. When a student joins the institute or leaves or some other information of the student changes, we will update the database.

We need to implement the following three operations for this purpose. They are as follows.

- Insertion of new student record: The details of the roll number and the name of the student will be provided as input.
- Deletion of a student record: The record for the student should be removed from the database.
- Query for a student, whether a student with a roll no exists in the database or not, and if it exists, return the name of the student.
- Number of students: Total number of students in the database.

You may assume that given key s , the hash function is $h(s) = s \bmod p$, where p is a prime number.

Expected input/output behaviour: Start with an initially empty hash table. Write functions for the above operations and based on the user input, provide outputs.

Submission: Name your submission as RollNo-Assign7.py and upload on Google classroom.