Page No. ______
Date

## Homework 5.1 .

Ans ⇒ Some desirable properties of hash function are -

- It should map the key to same value every time. There should not be any form of "randomness". Because if the same key could be mapped to more than one value, searching would become $O(n)$ instead of $O(\alpha)$ $[\alpha = \frac{n}{m}]$

- Calculating hash value should be fast.

- Mapping should be as uniform as possible.

- It should [illegible] use [illegible] (ideally) all parameters available (any not some parameter like "last 2 digits").

- Considering the function given by Rijurekha ma'am.

- It does not map all entries to different values. "Rahul Garg" (sir) and "Yogish Sabharwal" (sir) are mapped to same value.

- Calculating hash value is fast. (Assuming length of name is not very large).

- It maps the key to same value every time.

- It uses all the parameters available (name, age & gender)

Page No.
Date

# Homework 5.2

Ans =>

```
counter = 0
function H(x):
        counter = counter + 1
        return  (x + counter) mod 13
```

- Calculation of hash value is fast.

- It does relatively less number of "juggling" (as prof Naveen Garg says).
  for example MAD technique (multiply, add and divide) ~~[illegible]~~ is more juggled than this.

- It may not map the key to same value every time because value of counter is not constant.

- ~~[illegible]~~ Mapping is uniform. (assuming we want values to be between 0 and 12 only)

Make

Page No. ______
Date

## Homework 5.3

In chaining, lookup misses ~~take~~ take time proportional only to load factor $O(\alpha)$ as we have to search in the linked list corresponding to the key.

But in linear probing, searching also depends on how ~~[illegible]~~ many elements are filled. If filled elements are more, searching is more costly. It is ~~inverted~~ inversely proportional to empty space.

$$\text{time taken by lookup misses} \propto \frac{1}{\text{amount of emptyness}}$$

As load factor approaches 1, the array is ~~[illegible]~~ no more empty and time ~~taken~~ is very large.

## Homework 5.4

- Better value of n can be 5.

Reason

- As said by Dr. Naveen, we get into infinite loop when the incrementing (output of probing function) number is a factor of the size of array.

- Here since 5 and 9 are coprime, we will not run in a loop.

Note:- This is the reason why choosing a prime N is recommended.

In general, if n ~~[illegible]~~ and a are co-prime, then cycle problem would not occur (argument follows from Dr. Naveen's statement and the fact that GCD of two-coprime numbers is 1).