# Lab 2

## Exercise 1:

**1) Indicate the advantages of using the hash function of fig. 5.4 of the course book, instead of the hash function of the program in test.cpp**

**Answer:**

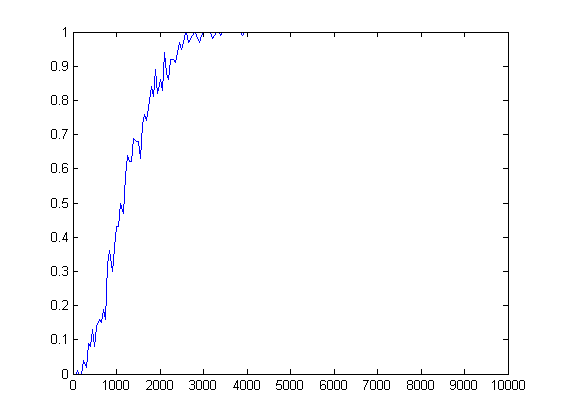
The first hash function gives smaller values when calculated compared to the hash function of fig. 5.4 of the course book which does multiplication with a prime number, and therefore gives higher values. That allows for a better distribution of the keys.

**2) Is it possible to display a frequency table, implemented as an instance of class HashTable, sorted by the key (as in the lab of TNG033 course)? Motivate your answer.**

**Answer:**

We don’t have our keys sorted because hash functions are not designed to sort alphabetically. So even if the words are similar, it doesn’t mean that they will be placed close to each other in the hash table.

## Exercise 2:



1) The probability of at least one collision when 2500 elements are inserted in the table is around 95%.

2) Number of elements that was needed to be inserted in the table for the probability of at least one collision to be at least 50% is 1100 elements.

3) Birthday paradox is a paradox that states if a room is filled with 23 people there is a probability of 50% that 2 people in the room has the same birthday. The probability reaches 99.9% that 2 people has the same birthday if the room is filled with just 70 people. This defies intuition. Because most people’s first intuition is to compare their birthday with everyone else in the room. The way to solve the birthday paradox is to compare everyone’s birthday with everyone, and multiply it.

4) The birthday paradox and hash tables have similar outcome. The probability is high for a hash collision already at a low number just like the birthday paradox.