Project #2: Due March 26th.

(Proactive Password checker): Make a program that checks a provided password using a bloom filter (See Bloom filter part in the textbook, Section 3.2).

A Bloom filter of order *k* consists of a set of *k* independent hash functions $H_1(x)$, $H_2(x)$, ..., $H_k(x)$, where each function maps a password into a hash value in the range 0 to N−1.

That is,

$H_i(X_j) = y, \quad 1 \le i \le k; \quad 1 \le j \le D; \quad 0 \le y \le N-1$

where X_j=jth word in a password dictionary

D = number of words in a password dictionary

The following procedure is then applied to the dictionary:

1. A hash table of *N* bits is defined, with all bits initially set to 0.
2. For each password, its *k* hash values are calculated, and the corresponding bits in the hash table are set to 1.

Thus, if for some (*i, j*), then the jth bit of the hash table is set to 1; if the bit already has the value 1, it remains at 1.

You can choose k hash functions by yourself. Dictionary D will be provided.
Set the parameters as follows. K = 3, N = 8192.
Generate a master table that has mapped all the words in the dictionary in the beginning. Accepts a new password and prints whether it is rejected or accepted on the screen. The input and its output pair are stored as a line in the file called 'Bloomcheked.txt' for each received input.