Homework for Intrusion Detection lesson

Write a program in a language of your choice which uses the hashing method described in class to determine if a given password is in a dictionary of bad passwords. Your program should operate as follows:

1. Prompt for and read in from the standard input the number of passwords, $p$, in the dictionary.
2. Prompt for and read in $p$ passwords, one at a time, from the standard input.
3. Enter a query loop, terminated by the sentry “quit”, which:
   1. Prompts for and reads from the standard input a password.
   2. Accepts or rejects that password based on the hash table, printing “Accepted.” for accepted passwords or “REJECTED.” for rejected passwords.

Your program should set aside a boolean map of size $m$, indexed 0 to $(m – 1)$. As your program reads in each password in the bad-password dictionary, it should hash it using three separate hashing algorithms (two of which a provided below, the third should be of your own design), each of which returns a value between 0 and $(m – 1)$. Your program should then mark these three indices in the boolean map. When it comes time to determine if a given password is in the dictionary, your program should hash the queried password with each hashing algorithm and then determine if all three indices have been set, in which case the password should be rejected.

HASH 1
Add (mod $m$) the ASCII value of each letter in the password.

HASH 2
Multiply (mod $m$) the ASCII value of each letter in the password.

HASH 3
Create your own.

Keep in mind that some queries may generate “false positives,” and be rejected even though they are not in the dictionary. However, there should be no “false negatives,” that is, every word in the dictionary should be rejected.

Example Session

> How many passwords are in the dictionary?
< 3

> Enter the words in the dictionary, one at a time.
< 1234
< abcd
< password

> Query password? ('quit' quits)
< goodPassword
> Accepted.
< another
> Accepted.
< 1234
> REJECTED.
< quit