Computer Science 303
Algorithms
Spring 2023
Assignment 4

Due: 11:59 p.m., Wednesday, 4/4/2023

For this project, you will write a program in C++ that implements hashing functions using two collision resolution techniques.

Your code must be modular and use vectors and classes. Additional information concerning these constructs will be given in class. Your solution should read strings from the terminal (terminated by a value of `"!"`) and output lists representing the hash tables produced using linear probing and perfect hashing. If for some reason, any item(s) cannot be inserted, hashing should continue with the next item in the list and a single line should be printed before the table for each item that could not be inserted.

The hash function for both schemes is given in the slides for string hashing, using the Weiss value for **R**. Each collision resolution scheme should be implemented as discussed in class/textbook and should return a Boolean indicating the success of the insertion. Additionally, each resolution scheme should be contained within its own class. For perfect hashing, each element of the hash table is a pointer to a secondary hash table. Note that you should write both a header file (**.hpp**) and a code source file (**.cpp**) for each, as well as **hashmain.cpp** to drive the program.

The **HashLin** class should declare the following private items: a vector hash table, an **init** method, which initializes all elements in the table to `""`, and a **hash** method that returns the index. For the public part, the class should include a constructor, which takes a **size** parameter, an **insert** method with a string parameter, and a **print** method to print out the elements of the table to the screen. Before passing the table size to the **HashLin** constructor, the size should be computed to be the closest prime number greater than the number of strings to insert. The constructor can call **init** to initialize the hash table.

The **HashPerfect** class should declare similar private items: a hash table array, an **init** method which initializes all elements in the table to **NULL**, and a **hash** function. The public parts of the class include: a constructor that simply calls **init**, an **insert** method, and a **print** method to print out the elements in the table and secondary tables to the screen. The **insert** method takes an entire vector of strings as a parameter, determines the number of strings that hash to each location, dynamically allocates memory for each location, then inserts all of the strings into the secondary hash tables. Note that the hash table should be an array of 10 pointers, each of which may point to a secondary hash table of type **HashLin**.

Your program should include files with the names below (along with corresponding **hpp** files for the hash classes). It can be compiled with the command:

```
g++ hashmain.cpp HashLin.cpp HashPerfect.cpp –o hash
```

..and run with the command:

```
./hash < keys.txt
```