Problem Set 4—Due Friday, October 25

Problem 4–1 Construct a fast-to-compute hash function h which is perfect (no collision) on the set of C++ keywords:

`asm`	`continue`	`float`	`new`	`signed`	`try`
`auto`	`default`	`for`	`operator`	`sizeof`	`typedef`
`break`	`delete`	`friend`	`private`	`static`	`union`
`case`	`do`	`goto`	`protected`	`struct`	`unsigned`
`catch`	`double`	`if`	`public`	`switch`	`virtual`
`char`	`else`	`inline`	`register`	`template`	`void`
`class`	`enum`	`int`	`return`	`this`	`volatile`
`const`	`extern`	`long`	`short`	`throw`	`while`

Regard the input to h as a number t and a string $s = s[0] \cdots s[t-1]$ of t ASCII characters. Make the output to h a number in $[0..m-1]$ for some m which you may choose. Make h as fast-to-compute as possible without making m excessively large. As you may want to experiment on your computer to help find a nice h, the above list of reserved words has been dropped to our Web page.

Problem 4–2. Illustrate the behavior of Forward Radix Sort by sorting the following octal numbers:

472 035 144 256 257 653 364 265 377 177 622 541 717 273 321 350

Let the buckets correspond to octal digits. Show all your work.

Problem 4–3. Problem 12-5 (p. 242).

Problem 4–4. Problem 18-2 (p. 376).