MATH20029: Example Sheet 1

John ff

March 19, 2004

1. Using LEX and possibly associated C, write a lexical analyser for the following subset of a C-like language:

   - Comments starting with a `//` and lasting to the end of the line,
   - Variables made from upper, lower-case letters and digits only, but starting with a letter
   - Integers in decimal notation with an optional leading sign
   - Strings, enclosed in " characters, but no escaped characters (*i.e.* no `\n`)
   - The keywords `if else while int void return`
   - The operators and punctuation `( ) , ; = * / || &&`

   Ensure that when your lexer has recognised a token it prints the lexeme.

2. Write a function that takes a string and returns a positive integer that can be used as a hash code. Test it on some text files to show what the distribution of hash values is; do the same mod 1024

3. Write a hash table that can hold unique copies of strings, and provides the two functions

   - void add_string(char *);
   - char *lookup(char *);

   The second function should return the string stored in the hash system or NULL if it is not present.

4. Modify your LEX program to add the variables to the hash table you have written

5. Modify your symbol table to use a token structure rather than strings.

6. Modify your LEX program to create a token-type structure rather than just a recognition.