Compiler Construction (List 1)

Hans de Nivelle

03.10.2013

Aim of this task is that you are able to run and compile a program using Ubuntu in room 137, and to get some feeling for tokenizing, and how unpleasant it can be. Tokenizing is the first step of compilation, in which one decides what are the separate words in a stream of characters.

1. Write a (preferably) C^{++} or C program that reads characters from $\mathtt{std}:: \mathtt{cin}$ and which counts occurrences of the word 'cookie', not distinguishing lower case from upper case in the input.

./cookiemonster abcdCookieedeefCookiebeteteteCooki -> There were 2 cookies!

The program should not store more than one character. In particular, solutions that store the last six characters are not acceptable. (Cookie monsters are not smart, and have only tiny brains.)

2. Since we all know that cookies are very bad for health, we will modify the cookie monster into a fruit eating monster, more precisely an ananas monster. Write the ananasmonster:

> ./ananasmonster ananasabcanananasAnotherAnanAs -> There were 3 ananases