CS2104 – Programming Languages   September 2, 2012
School of Computing
National University of Singapore

Problem Set 2

Semester 1, 2012/13

Due: *September 16, 23:59*   **9 marks**

Submission: In IVLE, in the cs2104 workbin, you will find a folder called "Homework submissions". In that folder, there are currently *3 subfolders:* **PS2P01,…,PS2P03**. The last two digits of the folder name indicate the solution that is to be submitted into that folder: the solution to *Question 1* into **PS2P01**, and so on (that is, you need to submit 3 separate solutions to 3 questions). A solution should consist of a ***single text file*** that can be compiled, or loaded into the interpreter of interest and executed. You should provide as much supplementary information about your solution as you can, ***in the form of program comments***.

Problem 1 [B] [1 mark, submit to **PS2P01**]

Write a Prolog predicate definition that counts the number of occurrences of an operator inside an expression. For instance, the query:

```
?- count(a+b*c-(2+3*4)/(5*(2+a)+(b+c)^f((d-e)*(x-y))), *, C).
```

would count the number of occurrences of operator `*` in the expression given as the first argument. The return value should be bound to the variable passed as the third argument. For the query given above, the answer should be `C=4`.

Problem 2 [C] [2 marks, submit to PS2P02]

Write a Prolog program called `primes`, with two arguments `N` and `L`, that computes the list of the first `N` prime numbers and binds `L` to it. The following is an example of correct interaction with your predicate:

```
?- primes(5,L).
L = [2,3,5,7,11]
```

Problem 3 [C] [2 marks, submit to PS2P03]

Consider the language of arithmetic expressions with the operators `+`, `-`, `*`, and `/`, where the operands are either numeric constants, or Prolog atoms denoting mathematical variables. Thus, the Prolog term:

```
(x+2)*(a-x)
```

stands for the mathematical function $f(x)=(x+2)(a-x)$, where a would be symbolic constant. Write a Prolog program that computes the derivative of such an expression. For instance, the query:

```
?- derive((x+2)*(a-x),x,D).
```

should derive the expression given as the first argument with respect to the variable given as the second argument. The result should be placed in the variable given as the third argument. For the query above, the answer should be

```
D = (1+0)*(a-x)+(x+2)*(0-1)
```

Do not perform any arithmetic simplification in your solution to this problem. Simplification is the topic of the next problem.

Problem 4 [A] [4 marks, submit to PS2P04]

Write a Prolog program that performs the following arithmetic expression simplifications.

- Replaces *Expr*+0, *Expr*-0, 0+*Expr* with *Expr*.
- Replaces *Expr**1, *Expr*/1, 1*Expr with *Expr*.
- Replaces -1*Expr, *Expr**(-1), *Expr*/(-1), 0-*Expr* with –*Expr*.
- Replaces *Expr1*+(-*Expr2*) with *Expr* –*Expr2*.

In the rules given above, *Expr*, *Expr1*, and *Expr2* are generic expressions, of arbitrary complexity.

For instance, the expression

```
(1+0)*(a-x)+(x+2)*(0-1)
```

should be simplified into:

```
a-x-(x+2)
```