

## Coursework 2

This coursework is worth 3% and is due on 26 November at 16:00. You are asked to implement a tokeniser for the WHILE language, an evaluator for boolean and arithmetic expressions and a WHILE program for printing prime numbers.

You need to submit a document containing the answers for the questions below. You can do the implementation in any programming language you like, but you need to submit the source code with which you answered the questions. However, the coursework will *only* be judged according to the answers. You can submit your answers in a txt-file or pdf.

### Question 1 (marked with 1%)

Implement a tokeniser for the WHILE language. (1) Keywords in this language are

while, if, then, else, do, for, to, true, false and also, orelse, read, write

(2) Operators are

$+$ ,  $-$ ,  $*$ ,  $\%$ ,  $==$ ,  $!=$ ,  $>$ ,  $<$ ,  $:=$

(3) Strings are enclosed into "...", (4) you have parentheses (, {, }, and }, (5) there are semicolons ;, (6) whitespaces are either " " or  $\backslash n$ , (7) comments either start with  $\backslash \backslash$  and run to the end of the corresponding line ( $\backslash n$ ), comments can also be given by looking for  $/*$  as the beginning marker and  $*/$  as the end marker.

(8) Identifiers are letters followed by underscores \_, letters or digits. (9) There are also numbers, like 0, 1, ...

Once you have implemented all regular expressions for (1) - (9), then give the token sequence for the Fibonacci program shown below.

### Question 2 (marked with 1%)

Implement parser combinators and an evaluate function for arithmetic and boolean expressions. Arithmetic operations should include  $+$ ,  $-$ ,  $*$ ,  $\%$  (quotient). Boolean operations should include  $==$  (equal),  $!=$  (unequal),  $<$ ,  $>$ .

Using the parser and evaluation function, calculate the values for

- $17 < 3 * 3 * 3$
- $(29 - 20) * 3$
- $79 - 20 * 3$
- $2 * 2 != 12 \% 3$

### Question 3 (marked with 1%)

Write a program in the WHILE programming language that prints out all prime numbers between 0 and a fixed number (say 100). As a guidance have a look at the Fibonacci program and three nested loops program shown below.

```
1  /* Fibonacci sequence implemented in
2     the WHILE language */
3
4  write "Input a number ";
5  read n;
6  x := 0;    // start values
7  y := 1;
8  while n > 0 do {
9     temp := y;
10    y := x + y;
11    x := temp;
12    n := n - 1 // decrement counter
13 };
14 write "Result ";
15 write y
```

```
1  start := 1000;
2  x := start;
3  y := start;
4  z := start;
5  while 0 < x do {
6     while 0 < y do {
7         while 0 < z do { z := z - 1 };
8         z := start;
9         y := y - 1
10    };
11    y := start;
12    x := x - 1
13 };
```