This question paper consists of 3 printed pages, each of which is identified by the Code Number COMP332101.

This is a closed book examination.
No material is permitted.

School of Computing

May/June 2018

COMP332101

Programming Languages and Compilation

Answer ALL FOUR questions

Time allowed: 2 hours

Question 1

This question deals with the assembly language from `nand2tetris`.

(a) Let all the memory registers contain 0. Which values will be contained in M[0], M[1], M[2] and M[3] after the following commands have been executed? **[6 marks]**

```
@0
AMD=A+1
AMD=M+1
AMD=D+1
```

(b) Assume the registers M[13] and M[17] contain numbers of absolut value less than 2^{12}. Provide assembly code to swap the content of these registers. All other memory registers should not change, but the content of the A- and D-registers can be overwritten. **[6 marks]**

[question 1 total: 12 marks]

Question 2

This is a question about FORTRAN 1.

(a) What are the type and the value of the following expressions?

(i) `1+2/3` **[2 marks]**

(ii) `1/2+3` **[2 marks]**

(iii) `1.0+2/3` **[2 marks]**

(iv) `1.0/2+3` **[2 marks]**

(b) For real numbers x with $|x| \leq 1$ the sum $\sum_{n=0}^{99} \frac{x^n}{n!}$ approximates e^x. How would you compute this approximation in FORTRAN? **[5 marks]**

[question 2 total: 13 marks]

Question 3

This is a question about the function `append` that concatenates two lists in Haskell.

(a) Declare `append` using pattern matching. **[5 marks]**

(b) Declare `append` using `foldl` or `foldr`. **[5 marks]**

[question 3 total: 10 marks]

CONTINUED ON FACING PAGE

Question 4

Consider the context-free grammar $G = (\{A, B\}, \Sigma, P, A)$ with two terminals `(` and `)` in Σ, where

$$A \to AB \mid B \qquad\qquad B \to \texttt{(}A\texttt{)} \mid \texttt{()}$$

are the productions of G. Construct an LR(0) parser for G by completing the following tasks.

(a) Give all items of the augmented grammar G'. **[5 marks]**

(b) Construct the NFA that accepts viable prefixes of G'. **[10 marks]**

(c) Convert the NFA into a DFA. **[10 marks]**

(d) Construct the parsing table. **[10 marks]**

[question 4 total: 35 marks]

[grand total: 70 marks]