

Xi'an Jiaotong-Liverpool University
西交利物浦大学

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CSE104		Computer Science and Software Engineering	

2nd SEMESTER 2016/17 RESIT EXAMINATIONS

BACHELOR DEGREE – Year 2

DATA STRUCTURES AND ALGORITHMS

TIME ALLOWED: 2 Hours

INSTRUCTIONS TO CANDIDATES

- 1、 Answer all questions.
- 2、 Answers should be written in the answer booklet(s) provided.
- 3、 No calculator is allowed during the examination.
- 4、 Only answers in English are accepted.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAM HALL.

Part II. (25 marks) Answer the following Question

31. Consider prefix expressions with binary operators only in the following. Assume there exists no error in the prefix expressions to be evaluated.

Given an infix expression $5+4*9/(6-3)$, the equivalent prefix expression is $+ 5 / * 4 9 - 6 3$. To evaluate the expression we will check if the first three tokens (a token is an operand or an operator) are an operator followed by two operands (or 1 operand for unary operators). For example, with the expression $+ 5 / * 4 9 - 6 3$, the first three tokens are $+$, 5 , and $/$. The addition cannot be performed because the third token is another operator. To evaluate the expression we rotate it (move tokens from the front to the back) until the first three tokens are an operator and the operands it needs. Then we will perform the operation and store the result somehow so it can be processed again later if needed. 10

- (a) Identify the data structure(s) and method(s) used for the evaluation of prefix expressions in your solutions. You must explain **briefly why** these are used. 15
- (b) Write down the pseudo code for the abovementioned algorithm so that it will return the correct result after evaluating the given input expression.

You have the following as input:

T is a data structure holding the prefix expression to be evaluated.

Your algorithm (in pseudo code) will make use of the input given and some data structure to evaluate the given prefix expression. You will also need to specify what methods (or functions) for your algorithm to use upon the data structure.

Do not write your algorithm as a complete Java program.

Use the following template to fill in your solutions:

```
// T holds a prefix expression as input
eval (T) {
    while (T.size() is greater than 1) {

    }
}
```

Your solutions should not exceed 9 lines of pseudo code.

END OF PAPER