

Name _____ ID _____ Seat # _____



**King Mongkut's University of Technology Thonburi
Midterm Exam of Second Semester,
Academic Year 2015**

COURSE CPE 113 Algorithms and Data Structures
Friday, February 26, 2016

Automation Engineering 1st Yr.
9.00-12.00 h.

Instructions

1. This examination contains 10 questions, 7 pages (including this cover page).
2. The answers must be written in the examination paper.
3. No books, notes, calculators or any other documents can be taken into the examination room.
4. Use your consideration and explain it if you have certain doubts about the exam questions.

Students must raise their hand to inform to the proctor upon their completion of the examination, to ask for permission to leave the examination room.
Students must not take the examination and the answers out of the examination room.

Students will be punished if they violate any examination rules. The highest punishment is dismissal.

This examination is created by

Asst. Prof. Dr. Nuttanart Muansuwan Facundes

Tel. 0-2470-9256

Name _____ ID _____ Seat # _____

Fill the answers in this examination paper. You may use the back of the sheet if you need more space. *Total points = 35 points*

1. Given the following infix expression:

$$2 + 1 - (4 - 3 * 1) * 3$$

Change the above infix expression into postfix format by using stack. (5 points)

Input	Stack	Output

Name _____ ID _____ Seat # _____

2. From the postfix expression result in (1) calculate the value of (i.e. evaluate) the expression using stack. (5 points)

Postfix	Stack	Operation

3. (6 points) In labs, you have experienced many structures of data (e.g. linked list, stack) and the way to programmatically implemented them in C language.

If the linked list was defined by the following code. Using the start variable as the head node of the list.

```
typedef struct _node {  
    int data;  
    struct _node * nextNode;  
} NODE;  
NODE *start = NULL;
```

What do the following codes do? Explain shortly.

3.1

```
NODE * new_node = (NODE *) malloc(sizeof(NODE));
```

3.2

```
int f1() {  
    NODE *temp = start;  
    int x = 0;  
    while(temp != NULL) {  
        x++;  
        temp = temp->nextNode;  
    }  
    return x;  
}
```

3.3

```
NODE *temp = start;  
while(temp != NULL) {  
    printf("->%d", temp->data);  
    temp = temp->nextNode;  
}
```

4. (6 points) The following is the code for Postfix Evaluation algorithm. Some lines of the codes are missing.

```

float evaluatePostfix(char * postfix) {
    (1)
    float result = 0.0;
    char * token = strtok(postfix, " "); // split a token from the postfix
    while(token != NULL)
    {
        if(isOperand(token))           // check if token is operand
        {
            float operand = atof(token); // convert string to number
            (2)
        }
        else
        {
            char anOperator = token[0]; // convert string {+,-,*,/} to a character
            (3)
            result = calculate(operand1, anOperator, operand2);
            (4)
        }
        token = strtok(NULL, " ");
    }
    (5)
    (6)
    return result;
}

```

Giving the blocks of codes.

- a) `STACK * stack = createStack();`
- b) `destroyStack(stack);`
- c) `pushFloat(stack, operand);`
- d) `result = popFloat(stack);`
- e) `float operand2 = popFloat(stack);`
`float operand1 = popFloat(stack);`
- f) `token = strtok(postfix, " ");`
`token = strtok(NULL, " ");`

Match the blanks with the given blocks of code to complete the postfix evaluation algorithm.

- | | | |
|-----------|-----------|-----------|
| (1) _____ | (2) _____ | (3) _____ |
| (4) _____ | (5) _____ | (6) _____ |

5. (4 points) Show the results of the following sequence of operations in stack and queue.

add(4), add(8), add(1), add(6), remove(), remove()

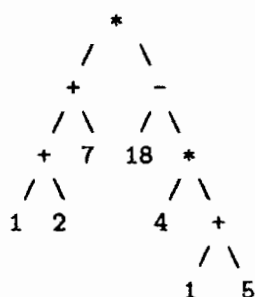
5.1 Stack

5.2 Queue

6. The **factorial** of a positive number is the product of the integral values from 1 to the number. For examples, $\text{Factorial}(3) = 3*2*1 = 6$, $\text{Factorial}(4) = 4*3*2*1 = 24$. Note that $\text{Factorial}(0) = 1$.

Write a recursive algorithm (with the function that calls itself) to calculate factorials (2 points).

7. What is the result of a) a preorder traversal, b) an inorder traversal, and c) a postorder traversal of the following expression tree (3 points)



a)

b)

c)

Questions 8-9: Select the choice that you think is correct (4 points)

8. A linked list is different from an array, because why?

- A. A linked list can handle more types of information than an array
- B. An array cannot be sorted but a linked list can be
- C. An array is fixed in size but a linked list is dynamically sizable
- D. An array is small in size but a linked list is big

9. What are two parts to recursion?

- A. (1) If the problem is easy, solve it immediately, and (2) If the problem can't be solved immediately, divide it into smaller problems.
- B. (1) Divide the problem into smaller problems, and (2) give immediate solutions for the hard problems.
- C. (1) Discard the hard cases, and (2) solve the easy cases.
- D. (1) Solve the problem by asking it to solve itself, (2) Solve the easy cases in one step.

10. Extra credit question (2 points):

What is the topic of your group's term project? What is/are the data structure type(s) that you use and what is the problem to solve?
