Seat No.

King Mongkut's University of Technology Thonburi
Midterm Exam of Second Semester, Academic Year 2017

COURSE
CPE 113 Algorithms and Data Structures — **Computer Engineering 4th Yr.**
CPE 131 Programming with Data Structures — **Automation Engineering 1st Yr.**
Tuesday, March 6, 2018 — **13.00-16.00 h.**

---

kly 23/3/2018.

Instructions

1. This examination contains 9 questions, 8 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.

Name-Lastname_____ Student ID #_____

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.

Exam created by

Nuttanart Facundes
(Asst. Prof. Dr. Nuttanart Facundes)

This examination has been approved by the committee of Computer Engineering Department

(Assoc. Prof. Dr. Natasha Dejdumrong.
International Undergraduate Program Chairperson
Date 28 FEB 2018

Name_____ ID#_____

Total points: 25 points (25% of grading)

1. What would be the content of queue Q1 after the following code is executed and the following data are entered? (3 points)

```
1 Q1 = createQueue
2 S1 = createStack
3 loop (not end of file)
  1 read number
  2 if (number not 0)
    1 pushStack (S1, number)
  3 else
    1 popStack (S1, x)
    2 popStack (S1, x)
    3 loop (not empty S1)
      1 popStack (S1, x)
      2 enqueue (Q1, x)
    4 end loop
  4 end if
4 end loop
```

The data are: 5, 7, 12, 4, 0, 4, 6, 8, 67, 34, 23, 5, 0, 44, 33, 22, 6, 0

Name_____ ID#_____

2. (3 points)
Tracking the value of these variables:

(Note: be thoughtful about the meanings of *)
int a =3, b =4, c =5;
int* pa = &a, *pb = &b, *pc = &c;

	a	b	c	pa	pb	pc
a = b * c;						
a * = c;						
b = * pa;						
pc = pa;						
*pb = b * c						
c = (*pa) * (*pc);						
*pc = a * (*pb);						

3. (8 points)
Select from the following choices to fill in the blanks below:

A	Fibonacci Number	G	Larger instances of the same problem
B	Factorial of a number	H	AB*CD/+
C	Dynamic	I	ABCD+/*
D	Smaller instances of the same problem	J	Loop
E	Queue	K	Stack
F	Compile time	L	Base case

3.1 Recursion is a method in which the solution of a problem depends on _____
3.2 A problem that can be solved using recursion is_____
3.3 Recursion is similar to _____
3.4 In recursion, the condition for which the function will stop calling itself is _____
3.5 The postfix form of A*B+C/D is_____
3.6 A linear list of elements in which deletion can be done from one end and insertion can take place only at the other end is known as a_____
3.7 Linked list is considered as an example of _____ type of memory allocation
3.8 _____data structure is needed to convert infix notation to postfix notation.

4. Write down the C code used to create a new node in linked list (2 points)

_____

Name_____ ID#_____

5. Given that *data* variable is the array of structures of employee and the *data_count* variable is the number of item in array

```
typedef struct _employee {
    char name[50];
    char employee_type[30];     // {"full-time","part-time"}
    char gender[10];            // {"male", "female"}
    double salary;
} EMPLOYEE;
int data_count;
EMPLOYEE data[50];
```

Write down the C code for calculating the total salary for male employees (3 points)

```
double sumSalaryOfMaleEmployees (EMPLOYEE data[50], int data_count) {
    int i=0; double sum=0.0;









    return sum;
}
```

6. (4 points)

Given the structure of linked-list node, head variable and main function as below.

```
typedef struct _node
{
    int value;
    struct _node * next;
} Node;

Node head;

int main() {
    …
    printAll(head);
    return 0;
}
```

6.1) Write the *printAll()* function for displaying every value stored in the linked-list using **loop**.

```
void printAll (Node * current) {









}
```

6.2) Write the *printAll()* function (same purpose as in the previous question) using **recursion**.

```
void printAll (Node * current) {









}
```

Name_____ ID#_____

7. (1 point)

In the linked_list.c file, there are 4 linked-list operations which are

- append: insert the data as the last node of the list
- insertAt: insert the data as the node of the list at the given position
- delete: delete one node that contains the given value
- deleteAt: delete one node at the given position.

If the stack is implemented based on this linked_list.c file and stack's pop operation is done by calling deleteAt(0). Which of these code blocks is the most appropriate one?

A	`void push (int data) { append(data); }`	B	`void push (int data) { insertAt(data, 0); }`
C	`double top (int data) { return delete(data); }`	D	`double top (int data) { return delete(0); }`

Answer _____

Name_____ ID#_____

8. (1 point)

To complete the postfix evaluation function below, you are required to insert the code for (1).

```
float evaluatePostfix(char * postfix) {
    float result = 0.0;
    char * token = strtok(postfix, " ");
    while(token != NULL){
        if(isOperand(token)) {
            float operand = atof(token);
            pushFloat(operand);
        }
        else {
            float num1 = popFloat();
            float num2 = popFloat();
            float result = 0.0;
            char op = token[0];
            __________(1)__________
            pushFloat(result);
        }
        token = strtok(NULL, " ");
    }
    result = popFloat();
    return result;
}
```

Some of the codes below could make the postfix evaluation correct. What is it?

A	`if(op == '+')`
	`  result = num1 + num2;`
	`else if(op == '-')`
	`  result = num1 + num2;`
	`else if(op == '*')`
	`  result = num1 * num2;`
	`else if(op == '/')`
	`  result = num1 / num2;`
		B	`if(op == '+')`
			`  result = num1 + num2;`
			`else if(op == '-')`
			`  result = num1 - num2;`
			`else if(op == '*')`
			`  result = num2 * num1;`
			`else if(op == '/')`
			`  result = num2 / num1;`

C	`if(op == '+')`
	`  result = num2 + num1;`
	`else if(op == '-')`
	`  result = num2 - num1;`
	`else if(op == '*')`
	`  result = num1 * num2;`
	`else if(op == '/')`
	`  result = num1 / num2;`
		D	`if(op == '+')`
			`  result = num2 + num1;`
			`else if(op == '-')`
			`  result = num2 - num1;`
			`else if(op == '*')`
			`  result = num2 * num1;`
			`else if(op == '/')`
			`  result = num2 / num1;`

Answer _____

Name_____ ID#_____

9. Bonus points (up to *2 points)*

Write down one reading summary that your remember, or

Is there any challenge(s) that you did in labs? Write down one challenge that you did.