Seat No.

King Mongkut's University of Technology Thonburi
Midterm Exam of Second Semester, Academic Year 2016

COURSE CPE 131 Programming with Data Structures **Automation Engineering 1st Yr.**
Tuesday, February 28, 2017 **9.00-12.00 h.**

Instructions

1. This examination contains 12 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.

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. Peerapon Siripongwutikorn

International Undergraduate Program Chairperson

Date 21 ก.พ. 60

Name_____ ID#_____

Total points: 30 points (25% of grading)

1. In C programming, The array and pointer are declared as followed:

```
int A[] = {6,5,3,1,5};
int *p = A;
```

What are the output of the following expressions? Write your answer in the table. *(3 points)*

Expression	output
`printf("%d", *p);`	
`printf("%d", *p+2);`	
`printf("%d", *(p+2));`	
`printf("%d", A[1]);`	
`printf("%d", A[1+2]);`	
`printf("%d", A[1]+2);`	
`printf("%d", A-p);`	
`printf("%d", A[3]-*(p+3));`	
`printf("%d", &A[1]-(p+1));`	

The data structure code below is for the questions 2-4

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

2. In a linked-list, the *append* operation should insert the node at the end of the list. If the data structure is defined by the code above and some of the append function is implemented below.

```
if(head == null) {
    head = newNode;
}
else {
    Node * currNode = head;
    while(currNode->next != NULL){
        // (1)
    }
    // (2)
}
```

Now, you know that the variable `newNode` stores the new data already. You are going to continue the work to achieve the append function by filling in (1) and (2). *(2 points)*

Name_____ ID#_____

From the following, which are the best choices for (1) and (2)? Fill in the blanks below.

A	`currNode->next = currNode;`
B	`currNode = currNode->next;`
C	`newNode = currNode->next;`
D	`currNode->next = newNode;`
E	`head->next = currNode;`
F	`currNode->next = head;`

(1) _____ and (2) _____

3. To delete the first node of the list, which is the best choice? *(2 points)*

A	`Node* deleteNode = head;`
	`head->next = deleteNode;`
	`free(deleteNode);`

B	`Node* deleteNode = head;`
	`head = head->next;`
	`free(head);`

C	`Node* deleteNode = head;`
	`deleteNode->next = deleteNode;`
	`free(head);`

D	`Node* deleteNode = head;`
	`head = head->next;`
	`free(deleteNode);`

Answer _____

4. How many cases you have to handle separately when you implement the *deleteAt* function? What are they? *(3 points)*

5. In the given code that follows, each stack node would contain only a single character. Explain how the stack data structure gives you an advantage for reversing string. *(2 points)*

Name_____ ID#_____

```
typedef struct node {
    char data;
    struct node * next;
} STACK_NODE;
STACK_NODE * top = NULL;

void push(char data) {
    STACK_NODE * newNode = (STACK_NODE *)calloc(1,sizeof(STACK_NODE));
    newNode->data = data;
    newNode->next = top;
    top = newNode;
}

char pop() {
    char data = top->data;
    STACK_NODE * deleteNode = top;
    top = top->next;
    free(deleteNode);
    return data;
}
```

6. What are the similarity and difference between popStack and topStack? *(2 points)*

_____

_____

7. Convert these following infix to postfix notations. Write down your answer in the table. (3 points)

	infix notations	**postfix notations**
A	5+3-4	
B	5+(3-4)	
C	(5+4)*3	
D	(5*4)+3	
E	5*(4+3)	
F	(((a+b)*c-d)/(e+f))	
G	(a+b*c)-(d/(e+f))	

Name_____ ID#_____

8. Given that n is a positive integer. To solve the tower of Hanoi of level n (i.e. the number of disks), how many move are required? Explain the base case and general cases for solving the tower of Hanoi Problem. (3 points)

9. If you continue implementing infix to postfix transformation from the following code

```
void convertInToPostFix(char * infix, char * postfix)
{
    strcpy(postfix, "");
    beautifyInfix(infix); /* insert the space between tokens*/
    char * token = strtok(infix, " ");
    do {
        if (isOpenParenthesis(token)) {
            // (1)
        }
        else if (isCloseParenthesis(token)) {
            char * data = popString();
            while (!isOpenParenthesis(data))
            {
                // (2)
                // (3)
            }
        }
        else if (isOperator(token)) {
            char * data = (char*) topString();
            while (!stackEmpty()
                    && (priority (token) <= priority (data))) {
                // (4)
                // (5)
                // (6)
            }
            // (7)
        }
        else {
            // (8)
        }
    } while ((token = strtok(NULL, " ")) != NULL);
    while (stackSize() != 0) {
        char * data;
        // (9)
        // (10)
    }
}
```

Name_____ ID#_____

Given the following statements,

A	`pushString(token);`	D	`strcat(postfix, data);` `strcat(postfix, " ");`
B	`data = topString();`	E	`strcat(postfix, token);` `strcat(postfix, " ");`
C	`data = popString();`	F	`strcat(postfix, infix);` `strcat(postfix, " ");`

Fill in the appropriate statements below for infix to postfix evaluation. (5 points)

(1) _____ (2) _____ (3) _____ (4) _____ (5) _____
(6) _____ (7) _____ (8) _____ (9) _____ (10) _____

10. 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;
}
```

Name_____ ID#_____

Some of the codes below could make the postfix evaluation correct. What are they? *(2 points)*

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

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

Answer_____

11. Write the algorithm or pseudo-code for Fibonacci numbers as we learned in class.*(3 points)*

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

What is your group's term project topic? How many people are in your group and who are they?