Name_____ID_____S̶e̶a̶t̶#̶_____

ข้อเขียน

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

King Mongkut's University of Technology Thonburi   Inter
Midterm Exam of Second Semester,
Academic Year 2007

COURSE CPE 130 Algorithms and Data Structures   **Computer Engineering Department, 1st Yr.**
Wednesday, December 19, 2007   **9.00-12.00 h.**

Instructions
1. This examination contains 8 problems, 8 pages (including this cover page).
2. The answers must be written in the examination paper.
3. No books, notes, or any other documents can be taken into the examination room.
4. Calculators that conform to the university's regulations are allowed.
5. 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 designed by

Asst. Prof. Dr. Nuttanart Facundes

Tel. 0-2470-9256

This examination is approved by the Computer Engineering Department

(Asst. Prof. Suthep Madarasmi, Ph.D.)
Chair

Fill the answers in this examination paper. You may use the back of the sheet if you need more space. *Total points = 40 points*

1. (7 points)

Assume that you have a stack S, a queue Q, and the standard stack - queue operations: push, pop, enqueue and dequeue. Assume that print is a function that prints the value of its argument. Execute, in top-to-bottom order, the operations below and answer questions **a)**, **b)** and **c)** below.

```
enqueue(Q,5);
push(S,4);
enqueue(Q,8);
push(S, 6);
push(S, dequeue(Q));
print(pop(S));
push(S, 7);
enqueue(Q,3);
enqueue(Q, pop(S));
print(dequeue(Q));
push(S, dequeue(Q));
enqueue(Q, 2);
push(S,5);
enqueue(Q, pop(S));
print(dequeue(Q));
print(pop(S));
print(dequeue(Q));
```

a) Show the output from the print statements (1 point each):

first output	second output	third output	fourth output	fifth output

b) After the above operations are completed, how many items are left in stack S?

Answer (1 point):

c) After the above operations are completed, how many items are left in queue Q?

Answer (1 point):

2. (10 points)

Given the following infix expression:
(10 / 2 + 8 * 4) * ((3 + 5) + 9)

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

Input	Stack	Output

b) From the postfix expression in (a) calculate the value of the expression using stack. (5 points)

Postfix	Stack	Operation

3. (3 points)

For each application below, say what data structure(s) would be most appropriate for the task and explain why. An answer without the explanation is worth nothing.

    a. You need to implement a calculator that reads postfix expressions and returns their values.

    b. A CD writer is shared by multiple users. You want to schedule the requests from users for accessing it.

    c. A palindrome is a string that reads backwards the same as forwards, e.g., "radar" and "go dog". Note: Palindromes ignore spacing and punctuation. By comparing a string with its reverse, you can understand whether it is a palindrome or not.

4. (1 points)

Suppose cursor points to a node in a linked list. What Boolean expression must be true when cursor points to the tail node of the list?

> A. (cursor == NULL)
> B. (cursor->link == NULL)
> C. (cursor->data == NULL)
> D. (cursor->data == 0.0)
> E. None of the above

5. (2 points)

How long does it take to finish moving 100 discs in Tower of Hanoi game, at 100 moves per second? Answer approximately in terms of **years**.

6. (2 points)

Complete the following C statement for the algorithm of Fibonacci numbers.

```
int fibonacci (int n)  {



        }
```

7. (10 points)

Assume Head, Curr_Ptr, NewNode_Ptr have been declared as pointers; and InValue has been declared as int. Place the appropriate C code in each box.

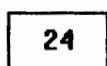|	
Head	**// Create a new node**
Head 12	**// Assign a data value to the new node**
Head 12 ↑ Curr_Ptr	**// Set up a current pointer for the head**
InValue 24	**// Get the data value for the next node**
	// while not end of input
NewNode_Ptr ↓	**// Create a new node**
	// Assign the input value to that node
InValue 24 >>> **NewNode_Ptr** ↓ 24	
Head 12 → **NewNode_Ptr** ↓ 24 ↑ Curr_Ptr	**// Set the link from the current node to the new node**
Head 12 → **NewNode_Ptr** ↓ 24 ↑ Curr_Ptr	**// Bump the current node pointer**
Head 12 → **NewNode_Ptr** ↓ 24 ↑ Curr_Ptr	**// Mark the tail of the list**
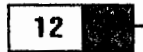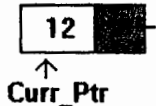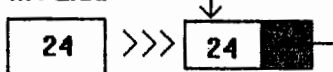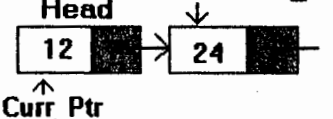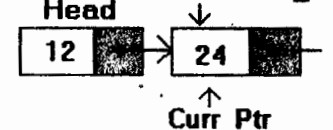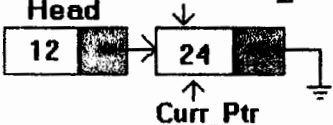
8. (5 points)

Suppose F is the pointer to the front of the queue and R is the pointer to the rear of the queue. The following queue stores A, B, and C

A	B	C				

F (points to A) R (points to C)

Write the algorithm of how to insert D into the queue, and how to delete from queue.