國立中興大學

109 學年度

碩士班考試入學招生

試　　題

學系：資訊科學與工程學系

乙組

科目名稱：資訊系統

109學年度碩士班招生考試試題

科目： 資訊系統　　系所： 資訊科學與工程學系 乙組

請於答案卷上作答，否則不予計分　**本科目不得使用計算機**　　本科目試題共 4 頁

PART I (50%)

1. (10%)

Suppose that the following declarations are in effect:

```
int   a[] = {5, 15, 34, 54, 14, 2, 52, 72};
int   *p = &a[1], *q= &a[5];
```

(1) What is the value of *(p+3)?

(2) What is the value of *(q-3)?

(3) What is the value of q-p?

(4) Is the condition p < q true or false?

(5) Is the condition *p < *q true of false?

2. (5%)

The following (rather confusing) function finds the media of three numbers. Please declare a new variable, add curly brackets, "{" or "}" and rewrite function, so that the function has just ONE return statement and each line contains exactly **one statement** or **one curly bracket**.

```
double median(double x, double y, double z)
{
   if   (x <= y)
      if   (y <= z)   return y;
      else if   (x <= z)   return z;
      else return x;
   if   (z <= y)   return y;
   if   (x <=z)   return x;
   return z;
}
```

3. (21%)

Suppose we have a finite sequence of numbers like A=(3, 1, 0, 2), and want to generate all its permutations. It turns out that a "good" approach to generating all the permutations is to start at the lowest permutation rank, (0, 1, 2, 3), and repeatedly compute the next lexicographical permutation in place. For example, the ***next*** lexicographical permutation of A is (3, 1, 2, 0), whilst the ***previous*** lexicographical permutation of A is (3, 0, 2, 1).

(1) (3%) Given a finite sequence of numbers B=(0, 1, 2, 5, 3, 3, 0), what is the **next** lexicographical permutation of B?
(2) (3%) What is the **previous** lexicographical permutation of B?
(3) (7%) Please describe your **next** lexicographical permutation algorithm **step by step** using the sequence B as an example.
(4) (8%) Please describe your **previous** lexicographic permutation algorithm **step by step** using the sequence B as an example.

4. (14%)

We can summarize the key properties of all the comparison-based sorting algorithms in the following table. Please complete this table, for example *O(n)*.

Sorting Algorithm	Strategy employed	Objects manipulated	Worst case complexity	Average case complexity
Bubble Sort	Exchange	Arrays	(1)	(2)
Selection Sort	Selection	Arrays	(3)	(4)
Insertion Sort	Insertion	Arrays/Lists	(5)	(6)
Treesort	Insertion	Trees/Lists	(7)	(8)
Heapsort	Insertion	Arrays	(9)	(10)
Quicksort	Divide & Conquer	Arrays	(11)	(12)
Mergesort	Divide & Conquer	Arrays/Lists	(13)	(14)

背面有題，請繼續作答。

PART II (50%)

5. (10%)

Single-Choice Questions

(a) The operating system is [ ]

i. Application software.
ii. Hardware.
iii. System software.
iv. A user program.

(b) Direct memory access is a technique that [ ]

i. Is most effective for slow character devices.
ii. Cannot be used for block devices.
iii. Enables the associated controller to read and write data directly from/to primary memory with no CPU intervention during data transfer.
iv. None of the above.

(c) In a time-sharing system, an interrupt for a user's program is handled by [ ]

i. The user's program directly.
ii. Operating system for the user program.
iii. Hardware.
iv. User's current shell program.

(d) In a system where round robin is used for CPU scheduling, the following is true when a process cannot finish its computation during its current time quantum [ ]

i. The process will terminate itself.
ii. The process will be terminated by the operating system.
iii. The process's state will be changed from running to ready.
iv. None of the above.

(e) The following algorithm is proposed to solve the critical section problem between two processes P1 and P2, where lock is a shared variable. Which of the following statements is true regarding the proposed algorithm? [ ]

P_1

```
do {
   while (lock) { NULL;}
   lock = TRUE;
   critical section;
   lock=FALSE;
   reminder section;
 } while(TRUE);
```

P_2

```
do {
   while(lock) { NULL;}
   lock = TRUE;
   critical section;
   lock=FALSE;
   reminder section;
 } while(TRUE);
```

i. Mutual exclusion to the critical section is guaranteed.
ii. Both processes can be in their critical section at the same time.
iii. Lock should be initialized to TRUE.
iv. None of the above.

6. (15%)

Multiple-Choice Questions

(a) Suppose that a process is executing "counter=counter+1" while another process is executing concurrently and independently "counter=counter-2", where the counter is a variable shared between the two processes and is accessed only by the two statements. Given that the value of counter is five before execution, the possible value(s) after both processes finish their statement are [ ]

i. Three.
ii. Four.
iii. Five
iv. Six.

(b) The following statements are true regarding a monitor where no condition variable is defined [ ]

i. Only one process can be running inside the monitor at any time.
ii. Processes can be blocked inside the monitor.
iii. No process can be blocked inside the monitor.
iv. Mutual exclusion is guaranteed among all the functions/procedures defined within the monitor.

(c) The following statement(s) are true regarding system calls [ ]

i. System calls are functions defined as part of the operating systems.
ii. System calls are functions that run in user mode in a dual mode system.
iii. System calls are implemented using a trap instruction which generates an interrupt.
iv. System calls are functions that run in system mode in a dual mode system.

(d) The following functions are necessary for an operating system that supports multiple users and multiple processes [ ]

i. Process management.
ii. Device management.
iii. Memory management.
iv. Non-preemptive scheduling algorithm.

(e) The following statements are true regarding a device that uses interrupt-driven I/O for I/O operations [ ]

i. The CPU must continuously check whether the current I/O operation is done.
ii. The CPU is informed through an interrupt request line when the current I/O operation is done.
iii. I/O on the device can be overlapped with computation on CPU by other processes.
iv. A device status table is necessary to save information about pending I/O requests.

7. (15%)

Pretty Good Privacy (PGP) can provide message confidentiality, sender authentication and message integrity. Please describe the operations of PGP.

8. (10%)

Please describe the idea of cipher block chaining (CBC).