臺北市立大學

104 學年度研究所碩士班入學考試試題

班　　別：資訊科學系碩士班（資訊科學組）

科　　目：計算機概論（含程式設計）

不得使用計算機或任何儀具。

考試時間：90 分鐘【08：30－10：00】

總　　分：100 分

※ 注意：不必抄題，作答時請將試題題號及答案依照順序寫在答卷上；限用藍色或黑色筆作答，使用其他顏色或鉛筆作答者，所考科目以零分計算。(於本試題紙上作答者，不予計分。)

一、單選題（每題 3 分，共 18 分）

1. What name is given to the common microcomputer architecture that uses a single block of memory to **store both programs and data**?

   （A）RISC
   （B）von Neumann architecture
   （C）SISD
   （D）ABC

2. A **register** in a CPU can hold ___.

   （A）data
   （B）instructions
   （C）program counter values
   （D）all of the above

3. **Cache memory**

   （A）has greater capacity than RAM

   （B）is faster to access than CPU registers

   （C）is a permanent storage

   （D）is faster to access than DRAM

4. A process in the **running state** goes to the **waiting state** when _____.

   （A）it requests I/O
   （B）it gets access to the CPU
   （C）it finishes running
   （D）it enters memory

5. When a hashing algorithm produces an address for an insertion key and that **address is already occupied**, it is called a _____.

   （A）collision　（B）probe　（C）synonym　（D）linked list

6. As shown in the figure, PC2 is unable to access PC1. To troubleshoot this problem, the technician needs to confirm that the next hop interface is operational. Which **default gateway address** should the technician ping from PC2 to confirm this?

（A）10.0.0.254 （B）192.168.0.249

（C）10.0.1.254 （D）192.168.0.250
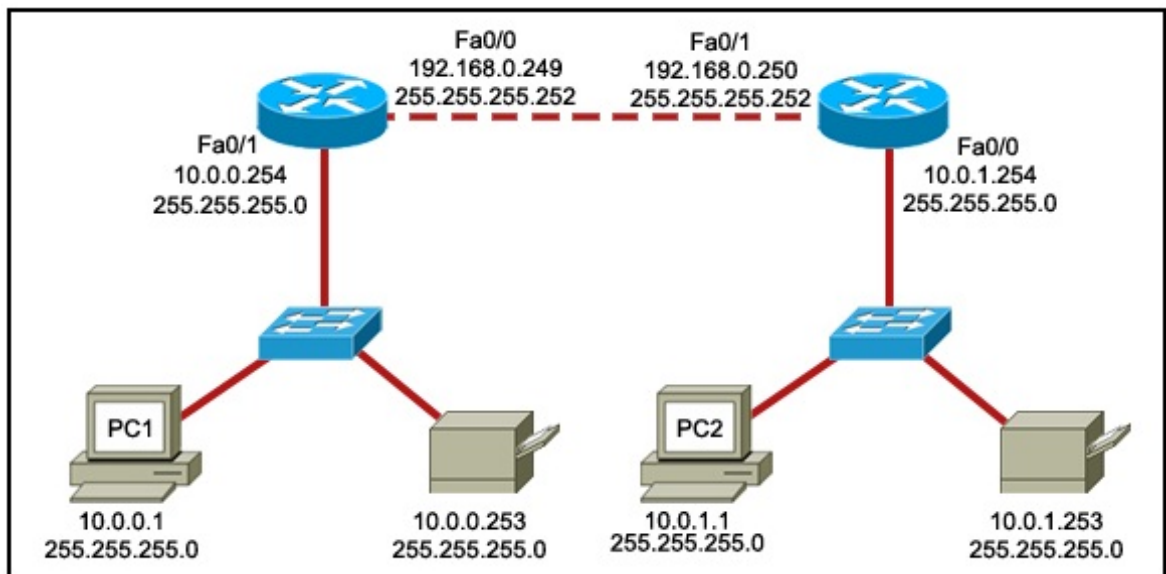
二、 單複選題（每題 4 分，全對才給分，共 16 分）

7. Generally, a programming language compiler

    （A）can detect syntax errors.

    （B）produces target language program.

    （C）can detect logic errors.

8. Which of the following task will require the largest amount of time?

    （A）fetch instructions from memory.

    （B）access data in memory.

    （C）execute the instructions.

9. A RAM chip of 1024 bit capacity, how many chips are needed to provide a memory capacity of 4K bytes?

    （A）4 （B）32 （C）1 （D）1024

10. The following bytes were originally encoded using odd parity. In which of them do you know that an error has occurred?

（A）10000111　（B）10101101　（C）11100110　（D）11111111

三、問答題（共 66 分）

1. Decode the bit pattern 10101010 in（7 分）

   (a) **sign-and-magnitude** representation

   (b) **2's complement** representation

   (c) **excess-127** representation

2. What will happen when you attempt to compile and run the following Java code?（5 分）

```
public class ThreadTest extends Thread {
    public static void main(String arg[]){
        new ThreadTest(){
            public void run() {
            System.out.println("University of Taipei");
            }
        }.start();
        System.out.println("Computer Science");   }}
```

3. (a) Convert the infix expression $A * (G + C) / T$ to the postfix expression.（4 分）

   (b) Convert the postfix expression $10\ 2 * 100 + 30 - 70\ 7 / +$ to the infix expression.（4 分）

4. Build a Huffman tree according to the following table, in which each letter has its own frequency.（10分）

letter	A	R	N	D	C	E
frequency	45	13	12	16	9	5

5. The Towers of Hanoi is a puzzle consisting of three rods A, B, and C and n disks all of different sizes. The puzzle begins with all n disks in ascending order of size (as pictured below) on the rod A, the smallest on the top, thus making a conical shape. The objective of the puzzle is to move all n disks to the rod C in ascending order, obeying the following simple rules:

    i.   Only move 1 disk at a time from one rod to another.

    ii. Only take the top (uppermost) disk from a rod.

    iii. At no time may a disk be placed on top of a smaller disk.

    Write a recursive algorithm to solve the Tower of Hanoi with the minimum number of moves.（7分）
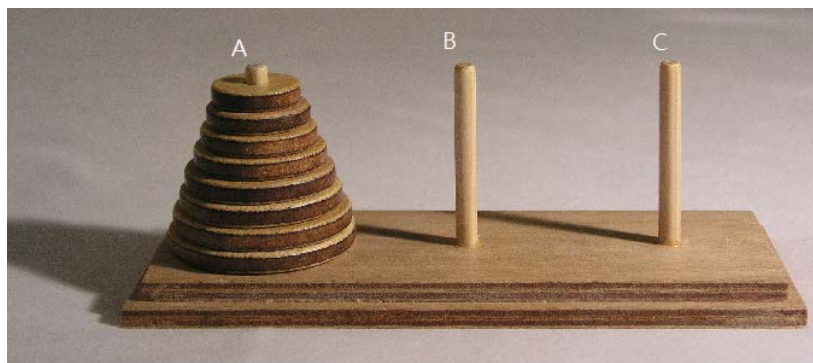
6. Explain the differences between stack and queue.（4 分）

7. What is Model–View–Controller (MVC)?（10 分）

8. Write down the results of the following programs.（15 分）

    (a)

```
int main(){
    int i;
    for ( i = 1 ; i < 25; i++ ) {
        if ( i % 6 == 0 && i > 6 ) {
            printf("%d\n", i);
        }
    }
    return 0;
}
```

(b)

```
#include <stdio.h>
void f1( int number ){
    number = number * 20;
}
void f2( int *nPtr ) {
    *nPtr = *nPtr * *nPtr ;
}
int main( void ) {
    int number = 3;
    f1( number );
    printf( "%d\n", number );
    f2( &number );
    printf( "%d\n", number );
    f1( number );
    printf( "%d\n", number );
    f2( &number );
    printf( "%d\n", number );
    return 0;
}
```

(c)

```
#include <stdio.h>
int f(int n) {
    if ( n == 1 || n == 2 ) {
        return 1;
    }
    else {
        return f(n-1) + f(n-2);
    }
}
int main(){
    printf ("ans: %d\n", f(6) );
    return 0;
}
```