

淡江大學 102 學年度碩士班招生考試試題

62-1

系別：資訊管理學系

科目：計算機概論

考試日期：3月10日(星期日) 第2節

本試題共 10 大題， 3 頁

注意事項：

- 一. 每題10分.
- 二. 務必依序作答!
- 三. 本試題以java描述. 但答題時可以自由選用java, C, C++, 或Pascal.
- 四. java的敘述:

```
Node p; p.data=5; int[] ans; Node.create();
```

分別相當於C++的敘述：

```
Node* p; p->data=5; int* ans; Node::create();
```

*****/

1. (a) 請解釋微米(micro-meter)及奈米(nano-meter).
(b) 請解釋 mega及 tera .

2. (本題須有計算過程)
(a) 請將八進位的(30715)₈轉成十六進位. (b) 請將十進位的(30715)₁₀轉成八進位.

3. (本題須有計算過程)
(a) 若使用8個bit來表示有號整數(signed integer), 則可表示什麼範圍的數?
(b) 接(a), 11011011 是代表什麼數值?

4. (a) 解釋C/C++/java中的4種位元(bit)操作運算, &, |, ^, ~.
(b) 寫一個敘述, 它將整數 x 的最低bit設為1, 而不改變其它位元.

系別：資訊管理學系

科目：計算機概論

考試日期：3 月 10 日(星期日) 第 2 節

本試題共 10 大題，3 頁

5. 觀察下列 java 程式碼，寫出它所印出的內容。

```
for(int i=1; i<=10; i=i+2) {  
    System.out.print(i);    /* 印 i */  
    System.out.print(" <!-- "); /* 印一串符號 */  
    for(int j=1; j<=i+2; j++) {  
        System.out.print(i); /* 印 i */  
        System.out.print(j); /* 印 j */  
    }  
    System.out.println(" --> "); /* 有印換行 */  
}
```

6. 觀察下列的 java 副程式

```
static int f(int n) {  
    if(n<2) return n;  
    return f(n-1)+2*f(n-2);  
}
```

- (a) 筆算求出 $f(0)$, $f(1)$, $f(2)$, $f(3)$, $f(4)$, $f(5)$ 的值。
(b) 分析這個副程式的複雜度。

7. 接上題，重寫這個副程式(可以用 C/C++ 作答)，以改進它的執行效能。

新副程式的複雜度必須是線性時間。

提示：可以使用陣列。若能不用，得分會較高。

淡江大學 102 學年度碩士班招生考試試題

62-3

系別：資訊管理學系

科目：計算機概論

考試日期：3月10日(星期日) 第2節

本試題共 10 大題， 3 頁

8. 閱讀下列 java 程式碼,

```
class Node { /* singly linked node */
    int data;
    Node link; /* a pointer */
    Node(int d) { data=d; } /* constructor */
}

class Mystery {
    public static void main(String[] _) {
        Node p1=new Node(1); Node p2=new Node(2);
        Node p3=new Node(3);
        p1.link=p2; p2.link=p3; p3.link=p2;
        Node p=p1;
        for(int i=1; i<=5; i++) {
            System.out.print(p.data);
            p=p.link;
        }
    }
}
```

(a) 若執行 `p=Node.create()`; 畫出 p 所取得的資料結構。

(b) 寫出 main 所印出的內容。

9. 利用上題的 Node 造一個 class Stack. 並撰寫適當的測試程式。

(如果用 C/C++ 回答, Node 必須作修正.)

10. (a) 舉例說明物件導向 (object-oriented) 技術中的繼承 (inheritance)

(b) 舉例說明動態繫結 (dynamic binding), 這又稱為虛擬呼叫 (virtual call)