

淡江大學 99 學年度碩士班招生考試試題

系別：資訊管理學系

科目：資料結構

本試題共 2 頁，10 大題

本試題雙面印製

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. 觀察下列 java 程式碼，寫出它所印出的內容.

```

for(int i=1; i<=5; i++) {
    System.out.print(i); /* 印 i */
    System.out.print("["); /* 印左方括號 */
    for(int j=1; j<=i+2; j++) {
        System.out.print(i); /* 印 i */
        System.out.print(j); /* 印 j */
    }
    System.out.println("]"); /* 印右方括號並換行 */
}

```

- 2. C/C++/java 中的位元 (bit) 操作運算包括:

& (bitwise AND), | (bitwise OR),
 ^ (bitwise exclusive-OR), ~(bitwise NOT).
 例如對於整數變數 x，敘述 x&=~1; 可將 x 的最低位元 (least significant bit) 設成 0 而不改變其它位元.

- (a) 寫一個敘述，它將 x 的最低 bit 設為 1，而不改變其它位元.
- (b) 寫一個敘述，它翻轉 x 的最低 bit (0,1 互換)，而不改變其它位元.
- (c) 設整數變數 x 佔用 32 位元. 寫一段程式碼，負責印出 x (視為無號數) 的二進位表示法.

- 3. 設 a 是長度為 5 的整數陣列，內容為 10, 20, 30, 40, 50.
 b 是長度為 10 的整數陣列.

閱讀下列 java 程式碼，寫出它所印出的內容.

```

int j=0;
for(int i=4; i>=0; i--) {
    b[j]=a[i]+1; b[j+1]=b[j]+1; j+=2;
}
System.out.println(j); /* 印 j 後換行 */
for(j=0; j<10; j++) {
    System.out.print(b[j]+","); /* 印 b[j] 及逗點 */
}

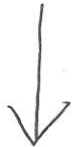
```

- 4. 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 */
    static Node[] create() {
        Node p1=new Node(1); Node p2=new Node(2);
        Node p3=new Node(3); Node p4=new Node(4);
        Node p5=new Node(5); Node p6=new Node(6);
        p1.link=p2; p2.link=p3; p3.link=p4;
        p4.link=p1; p5.link=p1; p6.link=p3;
        Node[] ans={p5, p6}; /* pointer array */
        return ans;
    }
}

```



◀ 注意背面尚有試題 ▶

系別：資訊管理學系

科目：資料結構

本試題共 2 頁，10 大題

```
class Mystery {
    public static void main(String[] _) {
        Node[] a=Node.create();
        Node p=a[0], q=a[1];
        for(int i=1; i<=5; i++) {
            p=p.link; q=q.link.link;
        }
        System.out.println(p.data);
        System.out.println(q.data);
    }
}
```

- (a) 若執行 `a=Node.create()`；畫出 `a` 所取得的資料結構。
(b) 寫出 `main` 所印出的內容。
5. 寫一個副程式，它的參數 `p` 是一個指標，指向 singly linked list 的首節點，它負責將首節點移置到最後，它的函數值是新的首節點的位址 (address)。若 `p` 為 `null`，或 `list` 中只含一個節點，則函數值為 `p`。
6. 觀察下列的 java 副程式
- ```
static int f(int n) {
 if(n<2) return n;
 return 3*f(n-1)-f(n-2);
}
```
- (a) 筆算求出 `f(0)`，`f(1)`，`f(2)`，`f(3)`，`f(4)`，`f(5)`=?  
(b) 分析這個副程式的複雜度。
7. 接上題，重寫這個副程式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，但若不用更好。
8. 設計一個二元樹 (binary-tree) 的資料結構，節點中可儲存整數資料。  
並撰寫一個副程式，負責以遞迴方式複製 binary tree。
9. 接上題，以 linked list 製作一個 queue 的資料結構，它所儲存的每一筆資料都是上題中二元樹節點的位址。
10. 接上題，寫一個副程式 `BTS`，它有兩個參數 `p`，`x`。  
`p` 是一個指標，指向二元樹的根節點，`x` 是一個整數。  
`BTS` 是要尋找二元樹中是否有某個節點的資料是 `x`。  
`BTS` 利用 queue 以廣度優先 (breadth-first) 的順序搜尋，  
函數值為最先遇到的內含 `x` 的節點的地址。  
若沒有這種節點則函數值為 `null`。