

系所別:

資訊工程學系

科目:

計算機概論

網路學習科技研究所

一、對於下列元素:

2, 7, 1, 8, 4, 5, 9, 0, 3, 6

依序加入 (insert) 到原來為空 (empty) 的度數為 6 的 B 樹 (B tree of order 6)。請畫出結果的 B 樹。(10 分)

二、某校資訊系課程可視為  $N$  個資料元素 (data elements) 構成的一個集合，裏面每個資料元素即一門課，含(1)課號(四位數字)(4 digits), (2)名稱(四個中文字)等資料欄位 (data fields)，以”課號”為鍵 (key)，假設依序加入 (insert) 下面四門課於該集合:

1234 資料結構

2345 軟體工程

3456 網際網路

2333 離散結構

分別用下面三種資料結構來表示該集合，請分別畫圖表示之:

(一) 陣列 (array)

(二) 雙鏈結環狀串列 (double linked circular list) (要有頭節點 (head node))

(三) 二元搜尋樹 (binary search tree) (15 分)

三、下面陣列 (array) 表示一個 min-heap:

1 3 4 5 8 9 6

(一) 請畫出此 heap 圖。

(二) 請畫出加入 (insert) 2 後，此 heap 變化後之圖。(10 分)

參考用

系所別:

資訊工程學系

科目:

計算機概論

網路學習科技研究所

四、(15%) Please draw a binary tree that you could use to store the letters A through M in order.

五、(20%) The eXtensible Markup Language (XML) is a standardized style for designing notational systems for representing data as text file. Following the XML standard, many other markup languages have been developed for different purpose of representation, for example SMIL is for multimedia representation, and MathML is for mathematic representation. Please list five more markup languages that are developed by following XML standard.

六、(30%) 請將以下程式執行後的輸出寫出來。

```
#include <stdio.h>
void a( void ); /* function prototype */
void b( void ); /* function prototype */
void c( void ); /* function prototype */
int x = 1; /* global variable */
int main()
{
    int x = 5; /* local variable to main */
    printf("local x in outer scope of main is %d\n", x);
    {
        /* start new scope */
        int x = 7;
        printf("local x in inner scope of main is %d\n", x);
    }
    /* end new scope */
    printf("local x in outer scope of main is %d\n", x);
    a(); /* a has automatic local x */
    b(); /* b has static local x */
    c(); /* c uses global x */
    a(); /* a reinitializes automatic local x */
    b(); /* static local x retains its previous value */
    c(); /* global x also retains its value */
    printf("local x in main is %d\n", x);
    return 0;
}
void a( void )
{
    int x = 25; /* initialized each time a is called */
    printf("\nlocal x in a is %d after entering a\n", x);
    ++x;
    printf("local x in a is %d before exiting a\n", x);
}
void b( void )
{
    static int x = 50; /* static initialization only */
    /* first time b is called */
    printf("\nlocal static x is %d on entering b\n", x);
    ++x;
    printf("local static x is %d on exiting b\n", x);
}
void c( void )
{
    printf("\nglobal x is %d on entering c\n", x);
    x *= 10;
    printf("global x is %d on exiting c\n", x);
}
```

