

淡江大學 97 學年度轉學生招生考試試題

系別：資訊管理學系三年級

科目：資料結構

可否使用計算機			
可		否	✓

本試題共 5 大題， 3 頁

注意:

每大題 20 分.

務必標明題號並依題序作答, 違者扣分.

作答的程式語言可任選 BASIC, C, Pascal, C++, 或 java.

1

(a) 閱讀下列的虛擬碼, 寫出它執行時所印的資料.

將 i 設為 1;

L1:

若 (i>5) 就跳到 L2

印 i;

將 i 設為 i+1;

跳到 L1;

L2:

結束;

(b) 閱讀下列的虛擬碼, 寫出它執行時所印的資料.

將 i 設為 1;

L1:

若 (i>5) 就跳到 L4

將 j 設為 1;

L2:

若 (j>i) 就跳到 L3;

印 i; 印 j;

將 j 設為 j+1;

跳到 L2;

L3:

輸出位置換行;

將 i 設為 i+1;

跳到 L1;

L4:

結束;

2. 寫一個完整的程式, 依序做下列的工作:

甲. 製造一個九格的陣列(array), 內容依序為 11,12,13,14,15,16,17,18,19.

乙. 用一個迴圈(loop)將這個陣列儲存的資料變成倒排.

丙. 最後再印出陣列內的資料.

3. 若規定 $A_0=1, A_1=1, A_{k+2}=A_k+A_{k+1}+1$

(a) 問 $A_6=?$ 作答時必需有計算過程.

(b) 寫一個完整的程式, 使用者輸入非負整數 n, 程式利用迴圈計算 A_n 的值. 並印出答案.

(c) 寫一個完整的程式, 使用者輸入非負整數 n, 程式利用遞迴函數(recursive function)計算 A_n 的值. 並印出答案.

本試題雙面印製

淡江大學 97 學年度轉學生招生考試試題

系列：資訊管理學系三年級

科目：資料結構

可否使用計算機			
可		否	✓

本試題共 5 大題，3 頁

4. 寫一個完整的程式，做下列的工作：

製造一種結構(struct 或 class)稱為 I3，內含整數欄位 x, y, z.

寫一個副程式稱為 SUM，它接收一個型態為 I3 的參數，並負責將 x+y 的值填入 z 欄位中

主程式依序做下列的工作：

甲. 造一個型態為 I3 的變數 p，將 x, y 欄位填入 10,20

乙. 以 p 為參數呼叫 SUM

丙. 印出 p 的 z 欄位的值.

5. (a) 閱讀下列程式並寫出執行結果

(b) 這個程式是用陣列(array)來模擬鍊結列(linked list)，用 index 來模擬地址(address)，請用現代的寫法(使用 struct/class, pointer)改寫一個等效的新程式.

```

/***** C 語言版本 *****/
#include "stdio.h"
#define BOUND 20
int DATA[BOUND]; int LINK[BOUND]; int AV=1;
int newNode(int D, int L) {
    int T=AV;
    if(AV>=BOUND) return 0;
    DATA[T]=D; LINK[T]=L;
    AV=AV+1; return T;
}
void print(int p) {
    int i;
    printf("[ ");
    for(i=p; i!=0; i=LINK[i]) {
        printf("%d",DATA[i]); printf(" ");
    }
    printf("\n");
}
void inc(int p) {
    int i;
    for(i=p; i!=0; i=LINK[i]) {
        DATA[i]=DATA[i]+1;
    }
}
int sample1() {
    int h=0; int i=10;
    for(i=10; i<=50; i=i+10) {
        h=newNode(i,h);
    }
    return h;
}

```

淡江大學 97 學年度轉學生招生考試試題

(8-)

系別：資訊管理學系三年級

科目：資料結構

可否使用計算機			
可		否	✓

本試題共 5 大題，3 頁

```

void main() {
    int p1=sample1();  print(p1);
    inc(p1);           print(p1);
}

/*****  java 版本  *****/
class LIA {
    static final int BOUND=20;
    static int[] DATA=new int[BOUND], LINK=new int[BOUND];
    static int AV=1;
    static int newNode(int D, int L) {
        if(AV>=BOUND) return 0;
        int T=AV;  DATA[T]=D; LINK[T]=L;
        AV=AV+1;  return T;
    }
    static void print(int p) {
        System.out.print("{ ");
        for(int i=p; i!=0; i=LINK[i]) {
            System.out.print(DATA[i]);
            System.out.print(' ');
        }
        System.out.println("}");
    }

    static void inc(int p) {
        for(int i=p; i!=0; i=LINK[i]) {
            DATA[i]=DATA[i]+1;
        }
    }

    static int sample1() {
        int h=0;
        for(int i=10; i<=50; i=i+10) {
            h=newNode(i,h);
        }
        return h;
    }

    public static void main(String[] noUse) {
        int p1=sample1();  print(p1);
        inc(p1);          print(p1);
    }
}
    
```