

110年公務、關務人員升官等考試、110年交通
事業公路、港務人員升資考試試題

等 級：薦任

類科(別)：資訊處理

科 目：程式語言

考試時間：2 小時

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

一、第 5 代行動通訊標準 (5G) 的網路架構包含終端設備、接取網路、邊緣運算、核心網路、外部服務等，其系統程式的開發需要許多軟體安全開發機制。

(一)為強化 5G 系統開發之安全性，可使用 SSDLC 和 DevSecOps 開發流程；請說明 DevSecOps 開發流程的要點。(15 分)

(二)軟體安全開發機制，包含建立威脅模型與分析、設計威脅模型的緩解措施。請說明若威脅模型中有會話劫持 (Session hijacking) 攻擊、機敏資料洩漏、Cookie 操控 (Manipulation)，其緩解措施分別如何設計。(10 分)

二、二元樹 (Binary tree) 可使用陣列 (Array) 和鏈結串列 (Linked list) 設計。

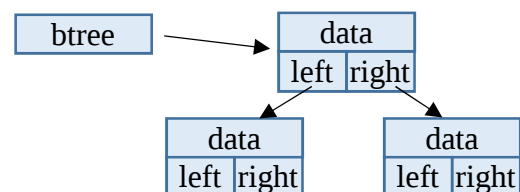
(一)以二元樹表達算術運算式 $c * (d - e) + f / g$ ，並寫出此運算式的後置表示法 (Postfix Expression)。(10 分)

(二)以一維陣列設計二元樹，節點循序編號，從陣列 1 開始表達根節點，節點編號規則為：左子樹是父節點編號乘以 2，右子樹是父節點編號乘以 2 加 1，-1 表示沒有子節點。請填寫以下陣列表達題(一)算術運算式之二元樹。(10 分)

索引	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
值		+	*	/												

(三)請根據右下圖完成下面 C 程式實作(I)~(V)之鏈結串列的表示，其中 btree 為指向一個二元樹的指標變數。(5 分)

```
____(I)____ node {  
    ____ (II) ____;  
    ____ (III) ____;  
    ____ (IV) ____;  
} treeNode_t;  
typedef ____ (V) ____ tree_t;  
tree_t btree;
```



三、給定正整數二維陣列，起點為其中數值最大的點。從起點開始移動，求經過點的數值之加總。移動規則：(1)從相鄰的點（上、下、左、右）選擇一個最大的值移動；(2)走過的點不能重複。

(一)請完成遞迴程式與非遞迴程式 (I~XI) 空格，使以下 C 程式均能執行出下列結果。(22 分)

path= 88, 42, 31, 18, 23, 21, 68, 36, 55, 77, 66, 63, 28, 33, 52, sum=701

(二)請比較遞迴與非遞迴程式記憶體空間使用狀況。(3 分)

```
#include <stdio.h>
#define maxN 30
const int NV = -1;
int max(int x, int y){int r = x>y ? (I) ; return r;}
int findR(int m[maxN][maxN], int x, int y, int sum){
    int mi = max(max(m[x+1][y],m[x-1][y]),max(m[x][y+1],m[x][y-1]));
    sum += m[x][y];
    printf("%d, ", m[x][y]);
    m[x][y] = NV;
    if (mi == NV) return (II) ;
    if (m[x+1][y] == mi) return findR(m, (III) , sum);
    if (m[x-1][y] == mi) return findR(m, (IV) , sum);
    if (m[x][y+1] == mi) return findR(m, (V) , sum);
    if (m[x][y-1] == mi) return findR(m, (VI) , sum);
}
int findI(int m[maxN][maxN], int x, int y, int sum){
    while(1){
        int mi = max(max(m[x+1][y],m[x-1][y]),max(m[x][y+1],m[x][y-1]));
        sum += m[x][y];
        printf("%d, ", m[x][y]);
        m[x][y] = NV;
        if (mi == NV) (VII) ;
        else if (m[x+1][y] == mi) (VIII) ;
        else if (m[x-1][y] == mi) (IX) ;
        else if (m[x][y+1] == mi) (X) ;
        else (XI) ;
    }
    return sum;
}
int main(){
    int x=0, y=0, mi=NV, n=5, m=6;
    int map[maxN][maxN]={ {NV, NV, NV, NV, NV, NV, NV},
                           {NV, 11, 15, 23, 18, 31, NV},
                           {NV, 31, 68, 21, 88, 42, NV},
                           {NV, 19, 36, 52, 33, 28, NV},
                           {NV, 12, 55, 77, 66, 63, NV},
                           {NV, NV, NV, NV, NV, NV, NV}};

    // 邊界都是 NV
    for (int i = 1 ; i <= n ; i++){
        for (int j = 1 ; j <= m ; j++){
            if (mi < map[i][j]){
                mi = map[i][j];
                x=i; y=j;
            }
        }
    }
    printf("path= ");
    printf("sum=%d\n", findR(map, x, y, 0));
    return 0;
}
```

四、敘述統計學上中位數和平均數均為數據資料的集中趨勢，中位數是將一組數值資料由小到大排列，最中間的數值為中位數。若有奇數個資料，則取最中間的數值為中位數，例如 1, 2, 3, 3, 4, 6, 7, 7, 21 的中位數是 4；若有偶數個資料，則取最中間兩個數值的平均為中位數，例如 1, 2, 3, 3, 4, 6, 7, 7, 8, 21 的中位數是 $(4+6)/2=5$ 。算術平均數是將一組數值加總，除以這組數值的個數，例如 1, 2, 3, 3, 4, 6, 7, 7, 21 的算術平均數 $=54/9=6$ 。以下 C++ 程式輸出：4, 6,

```

01 #include <iostream>
02 #include <algorithm>
03 using namespace std;
04 class Compute{
05 public:
06     void setData(int *, int);
07     (I) =0;
08 protected:
09     int *data, d_size;
10 };
11 class ComputeMedian: public Compute{
12 public:
13     double getNum();
14 };
15 class ComputeMean: public Compute{
16 public:
17     double getNum();
18 };
19 void Compute::setData(int *d, int s){
20     data = d;
21     d_size = s;
22     sort(data, data+d_size);
23 }
24 double ComputeMedian::getNum(){
25     if ( (II) ==1) return data[d_size/2];
26     else return (data[(III)]+data[1+d_size/2])/2.0;
27 }
28 double ComputeMean::getNum(){
29     double sum=0, avg=0;
30     for (int i=0; i<d_size; i++)
31         sum = sum + data[i];
32     return ((IV));
33 }
34 class Report {
35 public:
36     Report(int *d, int s){
37         data = d;
38         d_size = s;
39     }
40     void setCompute(Compute *c) { cp = c; }
41     void report(){
42         cp->(V);
43         cout<<cp->getNum()<<" ";
44     }
45 private:
46     int *data, d_size;
47     Compute *cp;
48 };
49 int main(){
50     int data[10] = {6, 7, 1, 21, 2, 3, 4, 3, 7};
51     Report r(data, 9);
52     Compute *cp1 = new ComputeMedian();
53     Compute *cp2 = new ComputeMean();
54     r.setCompute(cp1);
55     r.report();
56     r.setCompute(cp2);
57     r.report();
58     return 0;
59 }

```

- (一)請完成程式碼 (I~V) 使程式正常運作。(15 分)
- (二)請根據程式碼完成下面 UML 類別圖的關係連線，並說明此設計對模組耦合性 (Coupling) 的影響。(10 分)

Report
-data -d_size
+setCompute() +report()

{abstract} Compute
-data -d_size
+setData() +getNum(){abstract}

ComputeMean
+getNum()

ComputeMedian
+getNum()