

113年專門職業及技術人員高等考試建築師、
32類科技師（含第二次食品技師）、大地工程
技師考試分階段考試（第二階段考試）
暨普通考試不動產經紀人、記帳士考試試題

代號：01360
頁次：9-1

等 別：高等考試

類 科：資訊技師

科 目：程式設計

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、一位程式設計師為解決某一問題找到了一個有效但仍然相當複雜的演算法，因為程式執行完成所花費的時間非常重要，他根據過去所學得知「在相同的執行環境下，針對同一演算法以 C++ 撰寫的程式其執行速度比以 Java 撰寫的程式來得快」，決定以 C++ 撰寫程式來實現所找到的演算法。請闡述為何會有上述引號內之說法。(7 分)

二、下面所給 Java 程式有一錯誤以致無法成功釋譯(interpreting)，此錯誤落在某一行程式碼中，為讓此程式可以順利釋譯執行產生輸出結果，請指出那一行程式碼敘述有誤，更正其錯誤，並詳述更正的理由。(7 分)

```
import java.util.*;
public class P1 {
    int anInt = 0;
    static void aMethod(int array[]) {
        for(int i=0; i<array.length-1; i++)
            for(int j=array.length-1; j>i; j--)
                if(array[j] < array[j-1]){
                    int temp = array[j];
                    array[j] = array[j-1];
                    array[j-1] =temp;
                }
        anInt++;
    }

    static void bMethod(int array[]) {
        for(int i=0; i<array.length-1; i++)
            for(int j=array.length-1; j>i; j--)
                if(array[j] > array[j-1]){
                    int temp = array[j];
                    array[j] = array[j-1];
                    array[j-1] =temp;
                }
        anInt++;
    }
}
```

```
public static void main(String[] args) {
    Scanner inputObj = new Scanner(System.in);
    System.out.println("Enter the array size: ");
    int size = inputObj.nextInt();
    int[] intArray = new int[size];
    System.out.println("Enter the elements of the array :");
    for(int i=0; i<size; i++)
        intArray[i] = inputObj.nextInt();
    P1 callerObj= new P1();
    callerObj.aMethod(intArray);
    System.out.println( anInt + ". Array content: \n  " + Arrays.toString(intArray));
    callerObj.bMethod(intArray);
    System.out.println(anInt + ". Array content: \n  " + Arrays.toString(intArray));
}
}
```

三、針對下面所給的 C++ 程式，若輸入是 11 5 -3 42 -20 -43 99 43 -4 11 -22 0 4 7 9 -1 皆不相同的數，輸出為何？並說明輸出的資料所代表的意義。(8 分)

```
#include <iostream>
using namespace std;
int main()
{
    int N;
    int nNumbers[1000];
    int numLarger[1000] = {0};
    cin >> N;
    for (int i=0; i<N; i++)
        cin >> nNumbers[i];
    for (int j=0; j<N; j++){
        for (int k=0; k<N; k++)
            if(nNumbers[k] > nNumbers[j])
                numLarger[j]++;
    }
    int m;
    cin >> m;
    while (m > 0){
        for(int i=0; i<N; i++ )
            if( m + numLarger[i] == N){
                cout << nNumbers[i] << endl;
                break;
            }
        cin >> m;
    }
    return 0;
}
```

四、針對下面所給的 C++類別 (class) 的定義，一程式設計師可以有(A) 與(B)兩種使用方式，若某一函數 (function) 均有使用它們來建立物件，就此程式在編譯時、執行當中以及執行完畢後，詳述這兩種使用方式所建立的物件在記憶體配置與使用上之區別。(8 分)

```
Class Person {  
    public:  
        void setFirstName(string fName);  
        void setLastName(string lName);  
        void setAge(int age);  
    private:  
        string firstName;  
        string lastName;  
        int age;  
};
```

使用方式(A):

```
Person* aPerson = new Person();  
aPerson->lastName = "ChangYang";  
aPerson->firstName = "WenShiang";
```

使用方式(B):

```
Person aPerson;  
aPerson.lastName = "ChangYang";  
aPerson.firstName = "WenShiang";
```

五、下面所給為一 Java 程式，它的輸入檔 rsData.txt 包含一些以 0 與 1 所形成的字串，檔案中每一行 (line) 只包含一字串。就此程式，若輸入檔的內容如下所示，其輸出為何？並說明輸出的資料所代表的意義。(20 分)

```
import java.io.*;  
import java.util.*;  
import java.util.concurrent.atomic.AtomicReference;  
class rCalculator extends Thread {  
    private List<String> inputLines;  
    private int numRs;  
    private int thdNumber;  
    private char firstOrLastChar;  
    public rCalculator(List<String> inputLines, int thdNumber) {  
        this.inputLines = inputLines;  
        this.thdNumber = thdNumber;  
        this.numRs = 0;  
        this.lastChar = 0;  
    }  
    public int getNumRs() {  
        return numRs;  
    }  
}
```

```
public char getFirstOrLastChar(){
    return firstOrLastChar;
}
@Override
public void run() {
    String wholeStr="";
    for (String ithStr: inputLines)
        wholeStr = wholeStr + ithStr;
    char prevChar= wholeStr.charAt(0);
    numRs++;
    for(int j=1;j<wholeStr.length(); j++){
        if(wholeStr.charAt(j) != prevChar){
            numRs++;
            prevChar = wholeStr.charAt(j);
        }
    }
    if(thdNumber==1)
        firstOrLastChar = wholeStr.charAt(wholeStr.length()-1);
    else firstOrLastChar = wholeStr.charAt(0);
    System.out.println("Number of Rs = " + numRs + " ---> Thread " + thdNumber);
}
}

public class rsOfString{
    public static void main(String[] args) {
        List<String> inputData = new ArrayList<>();
        int lineCount = 0;
        try (BufferedReader br = new BufferedReader(new FileReader("rsData.txt"))) {
            String line;
            while ((line = br.readLine()) != null) {
                inputData.add(line);
                lineCount++;
            }
        } catch (IOException e) {
            e.printStackTrace();
            return;
        }
        List<String> inputForThread0 = inputData.subList(0, lineCount/2);
        List<String> inputForThread1 = inputData.subList(lineCount/2, lineCount);
        rCalculator thread0 = new rCalculator(inputForThread0, 0);
        rCalculator thread1 = new rCalculator(inputForThread1, 1);
        thread0.start();
        thread1.start();
        try {
            thread0.join();
            thread1.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        int totalNumRs = thread0.getNumRs() + thread1.getNumRs();
        if(thread0.getFirstOrLastChar()==thread1.getFirstOrLastChar())
            totalNumRs--;
    }
}
```

```
    System.out.println("The total number of Rs = " + totalNumRs);  
}  
}
```

輸入檔(rsData.txt)內容：

```
010001010100000000111111111111  
110001010001000000000000001001  
1100010000000000000000111111111111  
10000001111111111111111111111110
```

六、下面所給為一以 C++ 定義的 Array class，請回答下列問題：

- (一)為什麼 operator<<運算子多載(operator overloading)不宣稱為 Array class 的一個成員函數 (member function)？請詳述之。(5 分)
- (二)為什麼 copy constructor 如 Array(const Array &)的參數(parameter)必須是 reference type? 請詳述之。(5 分)
- (三)請撰寫一段 C++ 程式來實現成員函數 Array &operator-(const Array &)，此函數將兩個整數陣列 F 與 G 合併成一陣列 R。假設陣列 G 是這函數的參數 (parameter)，合併方式如下：(10 分)
 - F 中的整數按原順序一一複製到 R 中。
 - 接著 G 中的整數按原順序一一複製到 R 中若此一整數不同時在 G 與 F 中。
 - 陣列 R 的大小 (即 entry 的數量) 需與所複製的整數數量一樣。

若所撰寫之程式碼中之陣列名稱不是用 F、G、R，請於答案中註明以何變數名稱分別來稱呼 F、G、R 陣列，未加以註明者，本子題不予計分。此外，請就下面要點標示出所對應的程式碼，未加以標示者，本子題不予計分：

- 決定陣列 R 的大小之程式碼。
- 將陣列 F 的內容複製至陣列 R 的程式碼。
- 將陣列 G 的內容複製至陣列 R 的程式碼。

```
#include <iostream>
using namespace std;
class Array {
    friend ostream &operator<<( ostream &, const Array & ); // output an array
    friend istream &operator>>( istream &, Array & ); //input an array
public:
    Array( int = 10 ); // default constructor
    Array( const Array & ); // copy constructor
    ~Array(); // destructor
    int getSize() const; // return array size
    const Array &operator=( const Array & ); // assignment operator
    bool operator==( const Array & ) const; // equality operator
    bool operator!=( const Array &right ) const; // inequality operator
    int &operator[]( int ); // subscript operator
    int operator[]( int ) const; // subscript operator
    Array &operator+(const Array &); // merge two arrays
    const Array &operator>>(int i);
    const Array &operator-();
    Array &operator-(const Array &);
private:
    int size; // pointer-based array size
    int *ptr; // pointer to first element of pointer-based array
};
```

七、一位程式設計師設計了一個 C++類別 Pt 與類別 Shape，並使用它們來開發一些幾何形狀的類別，如三角形、平行四邊形、圓形、長方形、橢圓形等類別。程式設計師利用 Pt 類別來建立用以描述幾何形狀的點，並利用繼承(inherit)Shape 類別將這些幾何形狀的類別建構在 Shape 類別上，若主程式碼與執行主程式碼所產出結果如下所示，請回答下列問題：(每小題 5 分，共 10 分)

(一)為何用這樣的語法”virtual double area() const = 0;”來敘述計算一個幾何形狀面積的成員函數樣板 (function prototype)？請詳述之。

(二)請說明輸出結果的第四行”Point(0, 0) is created”是如何產生的。

```
class Pt{
    friend class Shape;
    friend class Triangle;
public:
    Pt(double =0, double =0);
    double getX() const {return x;};
    double getY() const {return y;};
private:
    double x;
    double y;
};
Pt::Pt(double c1, double c2) {
    x = c1; y = c2;
```

```
        cout << "Point(" <<x<<" , "<<y<<" ) created." << endl;
    }
class Shape {
public:
    Shape( const int &ctColor, const int &fdColor) : contourColor(ctColor), filledColor(fdColor){};
    virtual double area() const = 0; //計算一個形狀(Shape)的面積
    virtual double perimeter() const = 0; //計算一個形狀的周長
    virtual bool outside(const Pt &) const = 0; //決定所給的點是否在一個形狀外部
    virtual void print() const = 0; //把構成一個形狀的資訊列印出來
    virtual void draw() const = 0; //把一個形狀畫出來
    int getContourColor(){return contourColor;} //取出一個形狀的輪廓顏色
    int getFilledColor(){return filledColor;} //取出一個形狀內部填滿的顏色
private:
    int contourColor = -1; //一個形狀的輪廓顏色
    int filledColor = -1; //一個形狀內部填滿的顏色
};

int main() {
    Pt p1(0, 0);
    Pt p2(2, 0);
    Pt p3(1, 1);
    Triangle aTrgle(p1, p2, p3, 20, 10);
    aTrgle.area();
    aTrgle.outside(p1);
    aTrgle.perimeter();
    aTrgle.print();
    aTrgle.draw();
    cout << "Contour color: " << aTrgle.getContourColor() << endl;
    cout << "Filled color: " << aTrgle.getFilledColor() << endl;
    return 0;
}
```

輸出結果：

```
Point(0, 0) is created.
Point(2, 0) is created.
Point(1, 1) is created.
Point(0, 0) is created.
Point(0, 0) is created.
Point(0, 0) is created.
A triangle is created! 20 10
Triangle area is calculated:
Point outside a triangle is decided:
Triangle perimeter is calculated:
Points of a triangle are printed:
A triangle is drawn
Contour color: 20
Filled color: 10
```

八、請使用 HTML 與 JavaScript，將本題所提供之程式碼所缺部分（方框 1, 2, 3）撰寫完整，設計出可達成下列功能的網頁：（方框 1 占 4 分、方框 2 占 8 分、方框 3 占 8 分，共 20 分）

- 產出如下圖(A)(B)(C)之表單 (Form)，表單內主要包含兩個輸入框、一個按鈕、一個 result 區塊及一個 Canvas 元素。輸入框分別為「長度」和「寬度」，讓用戶在「長度」和「寬度」的輸入框中可分別輸入長方形的長和寬，單位為像素，「長度」和「寬度」的數值皆大於 0 且小於 500。
- 按下「計算面積」按鈕後，JavaScript 函式 calculateArea() 會讀取輸入的數值，檢查其有效性，並計算長方形的面積。
- 計算結果顯示在 result 區塊，並使用 Canvas 元素在網頁上繪製出一個長方形。Canvas 中的長方形會根據輸入的長寬居中顯示，且設定為藍色填充樣式。

```
<!DOCTYPE html>
<html>
<head>
  <title>長方形面積計算</title>
</head>
<body>
  <h1>長方形面積計算</h1>
  <label for="length">長度:</label>
  <input type="number" id="length" placeholder="輸入長度">
  <br>
  <label for="width">寬度:</label>
  <input type="number" id="width" placeholder="輸入寬度">
  <br>


方框1


  <p id="result"></p>
  <canvas id="rectangleCanvas" width="500" height="500" style="border:1px solid #000;"></canvas>

  <script>
    function calculateArea() {
      const length = parseFloat(document.getElementById('length').value);
      const width = parseFloat(document.getElementById('width').value);


方框2


      document.getElementById('result').textContent = '長方形的面積為: ${area} 平方單位';
      const canvas = document.getElementById('rectangleCanvas');
      const context = canvas.getContext('2d');

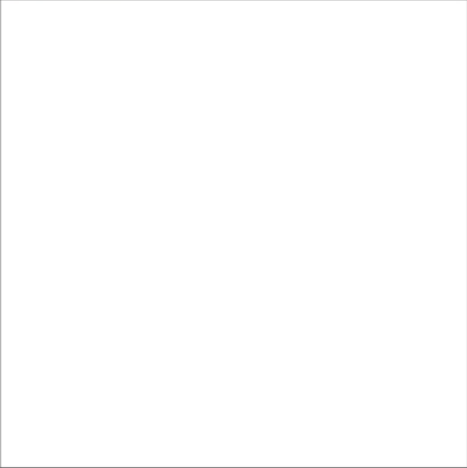

方框3


    }
  </script>
</body>
</html>
```

長方形面積計算

長度:

寬度:



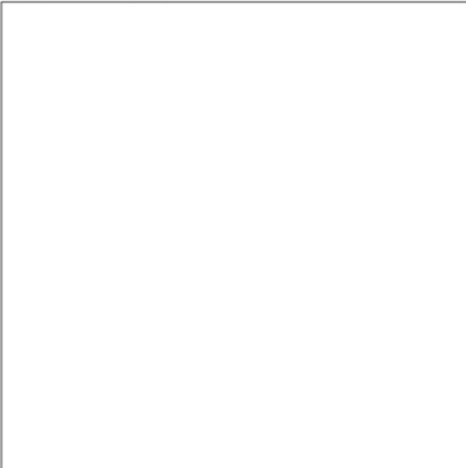
(A).表單初始畫面

長方形面積計算

長度:

寬度:

請輸入有效的長度和寬度!



(B).輸入無效數值狀況

長方形面積計算

長度:

寬度:

長方形的面積為: 50000 平方單位



(C).輸入有效數值狀況