

114年公務人員特種考試關務人員、身心障礙人員考試及  
114年國軍上校以上軍官轉任公務人員考試試題

考試別：關務人員考試

等別：四等考試

類科：資訊處理（選試英文）

科目：程式設計概要

考試時間：1 小時 30 分

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、下列 C 程式，輸入兩個字串，字串長度最長為 10 且字元可重複。程式判斷字串中的每一個字元，若兩個字串在同一個位置的字元都相同，則記錄一個 A，若兩個字串都有這個字元但位置不同，則記錄一個 B。程式最終輸出具 A 與 B 情況的字元與其數量。請寫出程式執行後輸出，以及計算 A,B, partA, partB 的程式碼之運作邏輯。(25 分)

```
01 #include <stdio.h>
02 void compare_strings(const char *X, const char *Y, int N) {
03     int A = 0, B = 0;
04     int countX[256] = {0}, countY[256] = {0};
05     char partA[11] = {0}, partB[11] = {0};
06     int indexA = 0, indexB = 0, partAN[11];
07     for (int i = 0; i < N; i++) {
08         if (X[i] == Y[i]) {
09             partA[indexA++] = X[i];
10             A++;
11         } else {
12             countX[(unsigned char)X[i]]++;
13             countY[(unsigned char)Y[i]]++;
14         }
15     }
16     for (int i = 0; i < 256; i++) {
17         int min_count = (countX[i] < countY[i]) ? countX[i] : countY[i];
18         if (min_count > 0) {
19             for (int j = 0; j < min_count; j++)
20                 partB[indexB++] = i;
21             B += min_count;
22         }
23     }
24     printf("%dA%dB; A: %s, B: %s", A, B, partA, partB);
25 }
26 int main() {
27     compare_strings("3A5@3", "35A63", 5);
28     compare_strings("f%09#2", "g5029#", 6);
29     return 0;
30 }
```

二、下列 Python 程式，可辨識輸入字串的不同類型。當註解掉 isC(s)函式程式碼，以及 Line 42, 43，請寫出程式執行後的輸出。若未註解上述程式碼，請完成 isC(s)函式(I)~(III)程式空格，辨識輸入 s 是否為浮點小數，使得 Line 42, 43 輸出"C,C"。(25分)

```

01 def isA(s):
02     parts = s.split('.')
03     if len(parts) != 4:         return False
04     for part in parts:
05         if not part.isdigit():return False
06         if int(part)<0 or int(part)>= 150:
07             return False
08     return True
09
10 def isB(s):
11     Elements = '0123456789abcdef'
12     newS = ""
13     parts = s.split(':')
14     if len(parts) != 3:     return False, newS
15     for part in parts:
16         if len(part) != 4: return False, newS
17         for c in part:
18             if not (c in Elements): return False, newS
19
20     for e in s:
21         if e in 'abcdef': newS += e.upper()
22         else: newS += e
23     return True, newS
24
25 def isC(s):
26     if s.count('.') (I) 1:           # (I)
27         return (II)                 # (II)
28     left, right = s.split('.')
29     return left.isdigit() (III) right.isdigit()   # (III)
30
31 def detect_type(symbol):
32     if isA(symbol): return 'A'
33     b, bs = isB(symbol)
34     if b : return 'B=>'+ bs
35     if isC(symbol): return 'C'
36     return 'E'
37
38 print(detect_type('140.100.100.80'))
39 print(detect_type('140.180.101.81'))
40 print(detect_type('f44d:30f8:1694'))
41 print(detect_type('TAIWAN'))
42 print(detect_type('123.52'), end=',')      #先註解
43 print(detect_type('0.987'))                #先註解

```

三、針對下列 Java 程式碼，請寫出此程式所有類別或介面之間的關係，以及執行後之輸出；並請說明 Line 34, 39 程式碼有問題的原因。（25 分）

```
01 interface Food { //食物
02     public abstract int getWeight();
03 }
04 class Deer implements Food { //鹿
05     public Deer(int w) { weight = w; }
06     public int getWeight() {return weight; }
07     private int weight;
08 }
09 interface Dragon { //龍
10     public abstract int eat(Food food);
11 };
12 class Dinosaur implements Dragon { //恐龍
13     public Dinosaur() { quantity = 0; }
14     public int eat(Food f) {
15         quantity += f.getWeight();
16         return quantity;
17     }
18     private int quantity;
19 };
20 class Tyrannosaurus extends Dinosaur { //暴龍
21     public Tyrannosaurus(Food f) { food = f; }
22     public String hunt(Food f) {
23         eat(food);
24         food = f;
25         int q = eat(f);
26         return food.getWeight() + ":" + q;
27     }
28     private Food food;
29 };
30 public class Hunt{
31     public static void main(String[] args) {
32         Food f1 = new Deer(3);
33         Food f2 = new Deer(5);
34         //Dragon d1 = new Dragon(); //程式問題
35         Dragon d2 = new Dinosaur();
36         Dragon t1 = new Tyrannosaurus(f1);
37         Tyrannosaurus t2 = new Tyrannosaurus(f1);
38         System.out.println("Dinosaur eat: "+d2.eat(f1));
39         //System.out.println("Tyrannosaurus eat: "+t1.hunt(f2)); //程式問題
40         System.out.println("Tyrannosaurus eat: "+t2.hunt(f2));
41     }
42 }
```

四、下列 Python 程式實作堆疊抽象資料型別，設定最大容量為 2。程式輸出是：

Push success=True

Push success=True

Push success=False

Pop success=True, data=8

Pop success=True, data=5

Pop success=False, data=-1

請填入 Line 01, 03, 06, 10, 18 程式碼空格(I)~(V)，使程式能正確執行。

(25 分)

```

01 MaxSize = (I) # (I)
02 def is_empty(top):
03     return top == (II) # (II)
04
05 def is_full(top):
06     return top == MaxSize (III) # (III)
07
08 def push(stack, top, n):
09     if not is_full(top):
10         top (IV) 1 # (IV)
11         stack[top] = n
12         return top, True
13     return top, False
14
15 def pop(stack, top):
16     if not is_empty(top):
17         data = stack[top]
18         top (V) 1 # (V)
19         return top, data, True
20     return top, -1, False
21
22 def main():
23     stack = [0] * MaxSize
24     top = -1
25     top, success = push(stack, top, 5)
26     print(f"Push success={success}")
27     top, success = push(stack, top, 8)
28     print(f"Push success={success}")
29     top, success = push(stack, top, 7)
30     print(f"Push success={success}")
31     top, data, success = pop(stack, top)
32     print(f"Pop success={success}, data={data}")
33     top, data, success = pop(stack, top)
34     print(f"Pop success={success}, data={data}")
35     top, data, success = pop(stack, top)
36     print(f"Pop success={success}, data={data}")
37
38 if __name__ == "__main__":
39     main()

```