

《程式語言》

一、請試述下列名詞之意涵：（每小題3分，共15分）

- (一)Context-Free Grammar
- (二)LR parser
- (三)Binding time
- (四)Turing Machine
- (五)Dynamic Programming

試題評析	本題為名詞解釋，相較於以往三等程式語言考試中，屬於較少見的題目。但名詞部分除LR parser外，其餘充分準備的考生應都不陌生，歷年也曾出現過許多相似題，如103檢事官、100檢事官、99高考、94地特……等。由於每題僅佔3分，因此僅需簡單說明每個概念的內涵即可，不需要太深入探討以免錯失其餘題目得分機會。
考點命中	<ol style="list-style-type: none"> 1.《高點高上程式語言講義》第六回，金乃傑編撰，頁1-2。 2.《高點高上程式語言講義》第三回，金乃傑編撰，頁56-57。 3.《高點高上程式語言講義》第一回，金乃傑編撰，頁115。 4.《高點高上程式語言講義》第二回，金乃傑編撰，頁29-30。

答：

- (一)上下文無關文法 (Context-Free Grammar)：若一個形式文法 $G = (N, \Sigma, P, S)$ 的產生規則都如： $V \rightarrow w$ ，則稱之，其中 $V \in N$ ， $w \in (N \cup \Sigma)^*$ 。因為 V 符號總可以被 w 字串自由替換，而無需考慮 V 符號出現的上下文故為上下文無關。由於上下文無關語法表達能力好，幾乎所有程式語言的語法都是用此語法定義。
- (二)LR剖析器 (LR parser)：為編譯器中語法剖析的程式，功能為建構語法樹。LR意指由左至右 (left to right) 處理輸入字串，並以最右邊優先衍生 (Right derivation) 的推導順序。由於LR剖析器嘗試由剖析樹的葉節點開始，向上一層層透過文法規則的化簡，最後規約回到樹的根部 (起始符號)，所以它是一種由下而上的剖析方法。
- (三)繫結時間 (Binding time)：為程式實體 (program entity) 與其屬性 (attributes) 發生關聯 (association) 的動作。依照時間的不同可分為定義 (Definition, 設計程式語言)、實作 (Implementation, 在機器上的運行規範)、編譯 (Compilation / Translation) 及執行 (Execute) 等四個時期。
- (四)圖靈機 (Turing Machine)：為英國數學家圖靈於1936年提出的一種抽象計算模型。圖靈的基本思想是用機器來模擬人用紙筆進行數學運算的過程，可分為兩種動作：在紙上寫上或擦除某個符號；把注意力從紙的一個位置移動到另一個位置。基於此動作，圖靈機的元件包括一條無限長的紙帶、讀寫頭、狀態暫存器及一套控制規則。
- (五)動態規劃 (Dynamic Programming)：是一種在數學、電腦科學中使用的計算方法，透過將原問題分解為相對簡單的子問題的方式求解複雜的問題。動態規劃的效益來自於將子問題的答案暫存，避免子問題被重新計算，以空間換取時間。例如：在計算費式級數時，將 $f(2)$ 、 $f(3)$ 的結果暫存，因此在計算 $f(4)$ 時就不需要重複計算 $f(1) \sim f(3)$ 了。

二、10 TBytes換算為多少bits? (5分)

試題評析	本題屬於計算機概論題目，只要細心應可輕鬆得分。
------	-------------------------

答：

$$1 \text{ TB} = 8 * 2^{40} \text{ bits} = 8,796,093,022,208 \text{ bits}$$

因為：

$$1 \text{ TB} = 1,024 \text{ GB} = 2^{10} \text{ GB}$$

$$1 \text{ GB} = 1,024 \text{ MB} = 2^{10} \text{ MB}$$

$$1 \text{ MB} = 1,024 \text{ KB} = 2^{10} \text{ KB}$$

【版權所有，重製必究！】

1KB = 1,024B = 2¹⁰B

1B = 8bits

三、關於副程式中參數的binding可分為shallow binding、deep binding和ad hoc binding等方法：

(一)請解釋上述三種binding的差別與優缺點。(9分)

(二)下列為Java Script的syntax codes，請問執行sub1之後，最後x的輸出(在sub2)為多少？請根據上述三種binding分別作答。(6分)

```
function sub1() {
  var x;
  function sub2() {alert(x);} // 輸出x
  function sub3() { var x; x=3;
                    sub4(sub2);}
  function sub4(subx) { var x; x=4;
                       subx();}
  x=1; sub3();};
```

試題評析	本題為繫結的進階考題，考非區域變數取值的三種方法。雖以往繫結都以名詞解釋出現(如101年關務四等考試)，沒有較深入題目，但由於此題為講義中經典補充，若充分準備的考生應能得心應手，獲得不錯的分數。
考點命中	1.《高點高上程式語言講義》，第三回，金乃傑編撰，頁76-77。 2.《高點高上地方特考題神考前猜題程式語言重點整理》，金乃傑編撰，考點6再次命中！

答：

(一)三種繫結的差別與優缺點比較如下：

	淺繫結 (Shallow Binding)	深繫結 (Deep Binding)	特設繫結 (Ad-hoc Binding)
內容	為副程式變數取值的實作方式，指副程式中若無定義變數，則會參考執行副程式的敘述當時環境的變數值來取值。	為副程式變數取值的實作方式，指副程式中若無定義變數，則會參考副程式定義時結構的上層來取變數值。	為副程式變數取值的實作方式，指副程式中若無定義變數，則會參考呼叫副程式時的環境來取變數值。
優點	執行效率高：可直接參考活動紀錄堆疊上層決定變數值。	可讀性高：依照結構取值，容易理解推算變數值。	彈性高：可在執行時動態決定副程式中的變數值。
缺點	可讀性低：難以直觀判斷函數中的變數值，程式維護難。	彈性低：程式設計師無法根據需求自動替換函數中的變數值。	執行效能差：尋找變數定義必須沿動態鏈往上找到呼叫處才能決定。

(二)X的輸出結果如下：

1.淺繫結 (Shallow Binding)：印出4

sub1()中敘述呼叫sub3()，而sub3()將sub2()視為參數傳入sub4()，在sub4()中的subx()即為sub2()，且sub4()中區域變數x值為4，故sub2()將4印出。

2.深繫結 (Deep Binding)：印出1

sub1()中經過一連串呼叫到sub2()時，要取x的值即往sub2()的上層(sub1()中)取出區域變數x值為1，故將1印出。

3.特設繫結 (Ad-hoc Binding)：印出3

sub1()中敘述呼叫sub3()，而sub3()將sub2()視為參數傳入sub4()，在此同時，也是程式調用sub2()之處，故sub2()要取出x值的時候，就會參考到此時sub3()中的x的值，故印出3。

四、函數 $f(n)$ 定義如下： $f(1)=1, f(0)=0, f(n)=f(n-2)+2f(n-1)$ ，請問 $f(5)$ 等於多少？（5分）

試題評析	本題為費式數列計算，必須寫出計算過程，若僅寫答案分數恐不理想。由於題目較簡單，應細心作答，以免懊悔萬分。
考點命中	《高點高上程式語言講義》，第二回，金乃傑編撰，頁28-30。

答：

$$f(5) = 5$$

計算過程如下：

$$f(5) = f(3) + f(4) = 2 + 3 = 5$$

$$f(4) = f(2) + f(3) = 1 + 2 = 3$$

$$f(3) = f(1) + f(2) = 1 + 1 = 2$$

$$f(2) = f(0) + f(1) = 0 + 1 = 1$$

五、對於字串的長度，不同的語言有不同的設計方式，包括static length string、limited dynamic length string、dynamic length string：（每小題6分，共12分）

（一）請解釋上述三種不同設計方式。

（二）對於Java、C和C++這三種語言，它們對字串長度設計的方式各採取那種方式或混和？

試題評析	本題為字串長度之題目。以往雖有不同語言中字串的細部考題，如101年警察、100年普考、96關務四等，但今年的題型不僅較有結構，而且還須說明三種語言的實作，必須對每種語言的字串都有一定了解才能拿到高分。在答案撰寫面，亦必須注意題意，第一小題先條列寫出三種方法，並配合舉例說明三種方法之用法、背後可能的技術；第二小題也應切合問題指出「各採取哪種方式」，始能拿到高分。
考點命中	1.《高點高上程式語言講義》第三回，金乃傑編撰，頁34-37。 2.《高點高上程式語言講義》第四回，金乃傑編撰，頁2。

答：

（一）三種不同的設計方式說明如下：

1.靜態長度字串（static length string）：字串在宣告時長度已決定，且不能再修改。

例如：在C語言中以char *s = "hello world"宣告，則此時s之長度就已經固定為12個char了（包含最後一個結束符號0）。

2.有限可變長度字串（limited dynamic length string）：字串在宣告時長度已決定，但可能會預留一些空間以在未來存入更多字串。

例如：在C語言中以char s[50]宣告，則s最大長度是50，但程式設計師可透過strcpy(s, "hello world")將較短的字串存入。

3.動態長度字串（dynamic length string）：字串在使用時，可以依照需求動態改變長度，通常在實作時必須將字串型態定義為類別，並透過宣告字串物件來建立。因為字串為一個物件，故內容可透過鏈結串列或參考到另一實際儲存字串的空間來儲存字元，達到長度可動態變化。

（二）對於Java、C與C++三種語言所採取的方式說明如下：

1.Java：提供動態長度字串、靜態長度字串。

在Java中字串為一類別，宣告方法如String s1 = new String("hello world")；但另外也可以透過String s2 = "hello world"將字串定義為一常數。若字串為常數時，則字串為靜態長度（儲存在code segment）。但因s2為一個參考，仍然可以透過s2 = new String("Jack")等語法參考到其他字串物件，引此長度的限制不會對程式造成影響。

2.C語言：提供靜態長度字串、有線靜態長度字串。

在C語言中字串是使用字元陣列模擬而來，因此若預先沒有宣告長度，如char *s1 = "hello world"就為固定長度；若預先宣告長度char s2[50]則為有限可變長度。

3.C++：提供靜態長度字串、有線可變長度字串、動態長度字串三種。

在C++標準函式庫提供了string類別，使用前要先include <string>。C++中建立String物件可透過string

sl(“hello world”)來建立。

六、給定下列的文法 (Grammar)：

$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

$\langle \text{id} \rangle = A | B | C$

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{id} \rangle | \langle \text{expr} \rangle * \langle \text{id} \rangle | (\langle \text{expr} \rangle) | \langle \text{id} \rangle$

請畫出右列字串： $A = ((A*B)+C*A)$ ，所對應right-most derivation sequence與對應的分析樹 (parse tree)。(5分)

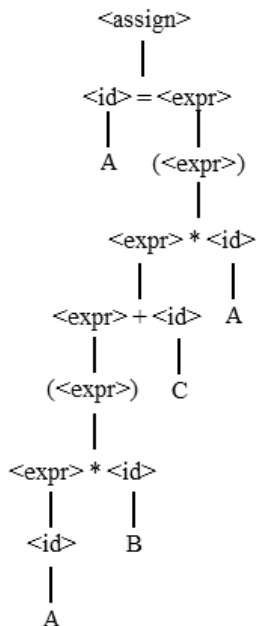
試題評析	本題為104身心障礙、104關務4等、102升官、101地特、101鐵路……等的相似題，若充分練習，細心作答，得到滿分絕非難事。
考點命中	1.《高點高上程式語言講義》第六回，金乃傑編撰，頁56-57。 2.《高點高上地方特考題神考前猜題程式語言重點整理》，金乃傑編撰，考點15再次命中！

答：

(一)最右推倒過程如下：

1. $\langle \text{assign} \rangle$
2. $\langle \text{id} \rangle = \langle \text{expr} \rangle$
3. $\langle \text{id} \rangle = (\langle \text{expr} \rangle)$
4. $\langle \text{id} \rangle = (\langle \text{expr} \rangle * \langle \text{id} \rangle)$
5. $\langle \text{id} \rangle = (\langle \text{expr} \rangle * A)$
6. $\langle \text{id} \rangle = (\langle \text{expr} \rangle + \langle \text{id} \rangle * A)$
7. $\langle \text{id} \rangle = (\langle \text{expr} \rangle + C * A)$
8. $\langle \text{id} \rangle = ((\langle \text{expr} \rangle) + C * A)$
9. $\langle \text{id} \rangle = (((\langle \text{expr} \rangle * \langle \text{id} \rangle) + C * A)$
10. $\langle \text{id} \rangle = (((\langle \text{expr} \rangle * B) + C * A)$
11. $\langle \text{id} \rangle = (((\langle \text{id} \rangle * B) + C * A)$
12. $\langle \text{id} \rangle = ((A * B) + C * A)$
13. $A = ((A * B) + C * A)$

(二)分析樹如下：



【版權所有，重製必究！】

七、請計算下列式子最後的y值：（每小題3分，共15分）

- (一) `int x=3, y=2; y*= ++x + 3;`
 (二) `int x=3, y=2; y /= x++;`
 (三) `int x=10, y=1; y= x++ + --y;`
 (四) `int x=10, y=1; y /= ++x + y--;`
 (五) `int y=0;`
`for(int k=0; k < 10; y+=k)`
`{ if(++k ==6) continue;`
`k++;}`

試題評析	本題一到四小題為前置加（減）與後繼加（減）的題目，作答時需掌握「後繼」會再敘述完成「後」再改變變數之值。另外也需注意*=等指定運算子也優先順序比加減乘除還低，因此必須等右手邊算式做完後才會展開帶入。若能掌握這些原則搭配細心作答，應能得分。第五小題為迴圈求值的題目，其考點還是在其中的if()每次進迴圈都會判斷，故if()中的++k也會每次進迴圈都做，故k的值都會被先加上1。
考點命中	1.《高點高上程式語言講義》第一回，金乃傑編撰，頁37-40。 2.《高點高上地方特考題神考前猜題程式語言重點整理》，金乃傑編撰，考點3再次命中！

答：

(一)14

```
y *= ++x + 3;
→y *= 4 + 3
→y *= 7
→y = y * 7
→y = 2 * 7
→y = 14
```

(二)0

```
y /= x++
→y /= 3
→y = y/3
→y = 2/3 = 0.6666...但取整數部分（無條件捨去），故為0。
```

(三)10

```
y = x++ + --y
→y = 10 + 0
→y = 10
```

(四)-1

```
y /= ++x + y--
→ y /= 11 + 1
→y /= 12
→y = y/12
→y = 1/12
→y = 0 在此敘述執行後，因前有y--，會再將答案減1，故為-1。
```

(五)30

原式可改寫為：

```
for(k = 0~9){
  ++k;
  if(k == 6) continue;
  k++;
  y += k;
```

【版權所有，重製必究！】

}
每一圈k與y的值为：

圈數	0(初始狀態)	1	2	3	4	5
k	0	2	4	6	8	10(下次會跳出迴圈)
y	0	2	6	12	20	30

八、若採取二種不同參數傳遞的方法：pass by reference、pass by value result，執行下列程式，則x與y的值各為多少？（8分）

```
int x=1, y=3;
void fun(int a, int b);
void main()
{ int x=2;
  fun(x, y);
  printf("x=%d, y=%d", x, y);
}
void fun(int a, int b) {a=b+x; b=a+y;}
```

試題評析	本題為參數傳遞考題，與103高考、102高考、99關務、94地特……相似，但因有全域變數，因此在作答時建議使用老師上課所教之「解題線」，較不會因為粗心而丟失分數。另外值得注意的是今年考出多年未考的pass by value result，需掌握此法為使用call by value「模擬call by reference」，先做call by value，等副程式執行完後再將結果寫回，故若副程式中沒有副作用，則答案與call by reference相同。
考點命中	1.《高點高上程式語言講義》第二回，金乃傑編撰，頁6-14。 2.《高點高上地方特考題神考前猜題程式語言重點整理》，金乃傑編撰，考點7再次命中！

答：

(一)pass by reference結果為：x = 4, y = 7

global		main	fun	
x	y	x	a	b
1	3 7	2 4	參考到main的x 運算式：a = b + x →global: y + global: x →3 + 1 = 4(與main的x同步)	參考到global的y 運算式：b = a + y →4 + global: y →4 + 3 = 7(與global的y同步)

(二)pass by value result結果為：x = 4, y = 7

global		main	fun	
x	y	x	a	b
1	3 7	2 4	數值2 運算式：a = b + x →3 + global: x →3 + 1 = 4(寫回main)	數值3 運算式：b = a + y →4 + global: y →4 + 3 = 7(寫回global)

九、(一)如果有兩個整數x, y，請寫出相對應的副程式碼，使得這兩數可以做交換。（5分）

(二)T c[10]; int m=3, n=2; (T為某種type，可能為int、float、double等)，請寫出相對應的副程式碼swap，當呼叫形式為swap(c, m, n)，可讓c[m]跟c[n]的值做交換，即使T的型態不同，此程式一樣可以正確處理。（5分）

試題評析

本題為久未出現的C語言「函數樣板」題目，與97年高考類似。若能掌握C語言函數樣板的語法，並能熟悉C語言中陣列傳入函數的寫法，此題應能迎刃而解。

考點命中

《高點高上程式語言講義》第四回，金乃傑編撰，頁115-117。

答：

(一)以C++撰寫如下：

```

6 void swap(int &x, int &y){
7     int temp = x;
8     x = y;
9     y = temp;
10 }

```

註：此處使用C++的傳參考呼叫，寫起來最簡潔乾淨。

(二)以C++撰寫如下：

```

6     template <class T>                //宣告樣板，將未知型態以T表示
7 void swap(T *c, int m, int n){
8     T temp = c[m];
9     c[m] = c[n];
10    c[n] = temp;
11 }
12
13 int main(){
14     int a[] = { 2, 3, 5, 7, 11 };
15     for (int i = 0; i < 5; i++){      //輸出原整數陣列
16         printf("a[%d] = %d ", i, a[i]);
17     }
18     swap(a, 1, 3);                    //置換
19     printf("\n");
20     for (int i = 0; i < 5; i++){      //輸出置換過後整數陣列
21         printf("a[%d] = %d ", i, a[i]);
22     }
23     printf("\n");
24     double b[] = { 2.54, 3.14, 4.25 };
25     for (int i = 0; i < 3; i++){      //輸出原實數陣列
26         printf("b[%d] = %.2f ", i, b[i]);
27     }
28     swap(b, 0, 2);                    //置換
29     printf("\n");
30     for (int i = 0; i < 3; i++){      //輸出置換過後實數陣列
31         printf("b[%d] = %.2f ", i, b[i]);
32     }
33 }

```

執行結果：

```

a[0] = 2 a[1] = 3 a[2] = 5 a[3] = 7 a[4] = 11
a[0] = 2 a[1] = 7 a[2] = 5 a[3] = 3 a[4] = 11
b[0] = 2.54 b[1] = 3.14 b[2] = 4.25
b[0] = 4.25 b[1] = 3.14 b[2] = 2.54

```

究！】

十、利用template的概念，寫出一個函數power(x, n)可以計算 x^n ，不管x為實數、整數或自然數，

但假設 n 為整數。(10分)

試題評析	本題配10分，考點除了template外，還需實作次方。雖C++中有Math函式庫中的pow()可用，但若使用了會使答案程式碼沒有內容（pow()已經很完整，有7種多載，也可以接受整數、浮點數、雙倍精度浮點數、長整數……等）。因此若要得到滿分，除了必須自己用迴圈處理次方，還必須要注意題目中提到的「 n 為整數」，故 n 可能為負。因此要考慮到 n 次方 n 為負數、0、或1的情形。
考點命中	《高點高上程式語言講義》第四回，金乃傑編撰，頁115-117。

答：

以C++撰寫如下：

```

6  template<class T>
7  T power(T x, int n){
8      T xBase = x;                //紀錄連續乘法每次要乘的數值
9      bool negative = false;      //預設n為正數
10     if (n < 0){                  //若n為負數
11         negative = true;
12         n = -n;                  //先將n變為正數以記錄次方
13     }
14     else if (n == 0) return 1;   //任何數的0次方都是1
15     for (int i = 1; i < n; i++) x *= xBase;
16                                     //用迴圈建立連續乘法
17     if (negative) x = 1.0 / x;    //若為負數次方，變為...分之一
18     return x;
19 }
20
21 int main(){
22     printf("%d", power(3, 5));
23 }

```

【版權所有，重製必究！】