

1. 你是運動 App 的程式設計師。你必須製作一個函式為跑者計算步速，所謂步速就是每公里所花的時間。輸出結果必須盡可能精準。要如何完成程式碼？請在回答區中選擇適當的程式碼片段。其中距離轉換為浮點數，分秒的輸入值都要轉換為整數。

#步速計算器

```
distance = (1)(input("Enter the distance traveled in meters"))
distance_kms = distance / 1000 # convert to kilometers
minute = (2)(input("Enter the time elapsed in minutes"))
time_sec = (3)(input("Enter the time elapsed in seconds"))
time = minute*60 + time_sec
pace = time / distance_kms
print("The average velocity is", str((pace//60)) + " minutes " + str((pace%60)) + " seconds per kilometer")
```

以上空格分別要填入的函式名為：

- (C)(1) A. int, B. string, C. float
- (A)(2) A. int, B. string, C. float
- (A)(3) A. int, B. string, C. float

答案:(1)C、(2)A、(3) A

解題分析：

(1)距離：使用者輸入的距離 `distance` 是以公尺為單位，並且在後續計算中會被 `distance / 1000` 轉換為公里。距離的測量通常需要精確，可能是小數（例如 1500.5 公尺），因此應該使用 `float` 函式將輸入轉換為浮點數，以保留小數點的精度。所以第一個空格應填 `float`。

(2)分鐘：題目明確要求「分秒的輸入值都要轉換為整數」。因此，使用者輸入的分鐘數 `minute` 應該使用 `int` 函式轉換為整數。

(3)秒數：同理，使用者輸入的秒數 `time_sec` 也應該使用 `int` 函式轉換為整數。

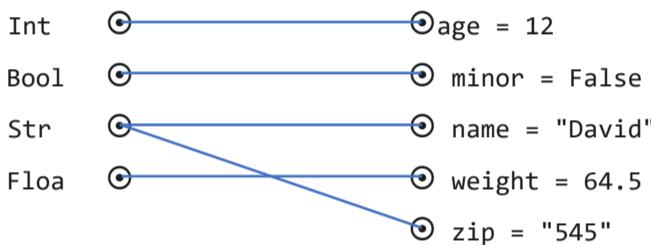
綜合以上分析，正確的函式依序是 `float, int, int`。

2. 你正在編寫一個 Python 程式用來記錄客戶資料並將其儲存在資料庫中。

這個程式處理各種各樣的資料。以下的變數宣告後它們的資料類別是？請將適合的程式碼片段連到正確的回答區。

程式碼片段

回答區



解題分析：

這個題目在測試對 Python 四種基本資料類型的理解。

1. Int (整數)：age = 12。12 是一個沒有小數點的整數，所以在 Python 中屬於 int 類型。
2. Bool (布林)：minor = False。True 和 False 是 Python 中僅有的兩個布林值，用於表示邏輯上的「真」與「假」，屬於 bool 類型。
3. Float (浮點數)：weight = 64.5。64.5 是一個帶有小數點的數字，這種數在 Python 中稱為浮點數，屬於 float 類型。
4. Str (字串)：
name = "David"。"David" 是由一串文字字元組成，並用雙引號包圍，這就是字串 (str)。
zip = "545"。雖然 545 是數字，但它被雙引號包圍，Python 會將其視為文字，而不是一個數值。因此，它也屬於 str 類型。在處理如郵遞區號、電話號碼等不需要進行數學運算的數字時，通常會用字串來儲存。

3. 你正在編寫一個計算使用者出生年份的程式。該程式詢問使用者的年齡和當前年份，然後輸出使用者的出生年份。你編寫以下程式碼。其中包含的行號只是做為參考。

```
01 age = input("Enter your age: ")
02 year = input("Enter the four digit year: ")
03 born = eval(year) - eval(age)
04 message = "You were born in " + str(born)
05 print(message)
```

請問下列何者是正確的？

- (V)A. 在 01 行中 year 的資料類型是 str
- ()B. 在 03 行中 born 的資料類型是 float
- ()C. 在 04 行中 message 的資料類型是 bool

答案:A

解題分析：

A：Python 的 `input()` 函式有一個重要特性：無論使用者輸入什麼內容（數字、文字、符號），它一律會以「字串 (str)」的形式回傳。因此，在第 02 行執行後，變數 `year` 儲存的會是使用者輸入的年份的「文字字串」，而不是一個數字。所以這個敘述是正確的。

B：在第 03 行，`eval()` 函式會嘗試將字串當作 Python 程式碼來執行。如果使用者輸入的是整數（例如 "2024" 和 "30"），`eval()` 會回傳 `int`。兩個 `int` 相減的結果 `born` 也會是 `int`(1994)。只有當使用者輸入的數字本身帶有小數點時，`born` 才可能是 `float`。因此，「`born` 的資料類型是 `float`」這個說法不一定成立，是錯誤的。

C：在第 04 行，`message` 是由兩個字串 ("You were born in" 和 `str(born)` 的結果) 透過 `+` 運算子連接而成。字串串接的結果永遠是字串 (str)，不會是布林值 (bool)。因此這個敘述是錯誤的。

4. 在 Python 資料類型的課程中創建以下三個程式碼片段：

```
# Code segment 1  
x1 = "5"  
y1 = 4  
a = x1*y1  
  
# Code segment 2  
x2 = 10  
y2 = 4  
b = x2/y2  
  
# Code segment 3  
x3 = 5.5  
y3 = 1  
c = x3/y3
```

你需要評估程式碼片段。請問下列何者是正確的？（可多選）

- (V)A. 執行程式碼片段 1 後，變數 a 的資料類型為 str
(V)B. 執行程式碼片段 2 後，變數 b 的資料類型是 float。
()C. 執行程式碼片段 3 後，變數的資料類型為 int。

答案：A、B

解題分析：

A：在片段 1 中，x1 是一個字串 "5"，y1 是一個整數 4。在 Python 中，當 * 運算子用於一個字串和一個整數時，它執行的不是數學乘法，而是「字串重複」。所以 a 的值會是 "5" 這個字串重複 4 次，也就是 "5555"。因為結果是文字，所以 a 的類型是 str。此敘述正確。

B：在片段 2 中，x2 和 y2 都是整數。/ 運算子在 Python 3 中執行的是「真除法」，其結果「永遠」是浮點數 (float)，即使剛好可以整除。10 / 4 的結果是 2.5，所以 b 的類型是 float。此敘述正確。

C：在片段 3 中，x3 是一個浮點數 5.5。只要有浮點數參與的除法運算，結果也一定是浮點數。5.5 / 1 的結果是 5.5，所以 c 的類型是 float，而不是 int。此敘述錯誤。

5. 在 Python 程式中我們利用 type() 查詢每個值的資料類別，以下的程式執行後出現的資料類別分別是：

```
type (1E10)  
type (5.0)  
type("True")  
type (False)
```

- ()A. int, int, bool, bool
(V)B. float, float, str, bool
()C. int, float, str, bool
()D. float, int, str, str

答案：B

解題分析：這個題目考驗對 Python 資料類型以及特殊寫法的辨識能力。

1. type(1E10)：1E10 是「科學記號」表示法，它代表 1 乘以 10 的 10 次方 (1×10^{10})。在 Python 中，只要使用科學記號表示的數字，都會被視為浮點數 (float)。

2. type(5.0)：5.0 是一個帶有小數點的數字，即使小數部分是 0，Python 也會將其歸類為浮點數 (float)。

3. type("True")："True" 是用雙引號包圍起來的一串文字。雖然 True 本身是一個布林值，但一旦被引號包圍起來，它就變成了純粹的文字，因此它的類型是字串 (str)。

4. type(False)：False 是 Python 的一個關鍵字，代表布林值「假」。它沒有被引號包圍，所以它是一個布林 (bool) 類型的值。

因此，這四個值對應的類型依序是 float, float, str, bool。

6. 你加入了電子商務公司成為其程式開發部門的實習生。你的程式中有一個地方要讓使用者提供一個數值。

即使使用者輸入了小數，該值也必須轉換為整數來進行計算。你應該使用哪個程式碼片段？

- ()A. totalNums = input("How many items would you like?")
(V)B. totalNums = int(input("How many items would you like?"))
()C. totalNums = str(input("How many items would you like?"))
()D. totalNums = float(input("How many items would you like?"))

答案：B

解題分析：題目的核心需求是「將輸入值轉換為整數」。

A：只使用 input() 會得到一個字串 (str)，不符合需求。

B：input() 取得使用者的輸入（一個字串），然後 int() 函式會嘗試將這個字串轉換成一個整數。如果使用者輸入的是像 "12.5" 這樣的小數字串，int() 會直接報錯。但如果使用者輸入的是 "12"，它會成功轉成整數 12。題目的要求是「必須轉換為整數」，int() 是唯一能達成這個目標的函式。

C：str() 會將輸入的字串 (input()) 的結果本身就是字串) 再轉一次字串，沒有意義，結果仍然是字串。

D：float() 會將輸入轉換為浮點數，而不是題目要求的整數。

因此，int() 是最符合「轉換為整數」這個明確指令的選項。

7. 你設計了一個 Python 程式用來顯示每個員工每天工作到現在的小時數。

你需要計算工作時數並顯示訊息。程式碼如下：

```
01 start = input("What time do you start work today?")  
02 end = input("what time is it now?")  
03
```

如果要完成這個程式，在 03 行應該使用哪個程式碼？

- (V)A. print("You have worked for" + str(int(end) - int(start)) + "hours")
()B. print("You have worked for" + (int(end) - int(start)) + "hours")
()C. print("You have worked for" + str(end - start) + "hours")
()D. print("You have worked for" + int(end - start) + "hours")

答案：A

解題分析：這個程式的目標是計算兩個時間點之間的差值並印出。

1. 輸入與類型：在第 01 和 02 行，start 和 end 都是透過 input() 取得的，所以它們的類型都是字串 (str)。

2. 計算：要計算時間差，必須先將這兩個代表時間的字串轉換成數字。int() 函式可以做到這一點。所以，計算式應該是 int(end) - int(start)。

3. 輸出：print() 函式如果要用 + 來串接不同類型的資料，必須確保所有部分都是字串。

"You have worked for" 是字串。

int(end) - int(start) 的計算結果是一個整數 (int)。

"hours" 是字串。

直接用 + 連接字串和整數會導致 TypeError。因此，必須使用 str() 函式將計算出的整數結果再轉換回字串。

4. 選項分析：

- (A) : str(int(end) - int(start)) - 先用 int() 轉成數字做減法，再用 str() 將結果轉回字串。所有部分都是字串，可以成功串接。這是正確的。
(B) : (int(end) - int(start)) - 計算結果是整數，直接與前後的字串用 + 連接會出錯。
(C) : str(end - start) - end 和 start 都是字串，直接相減會出錯。
(D) : int(end - start) - end 和 start 都是字串，直接相減會出錯。

1. 高年級的老師要製作一份報表來顯示這次考試班上所有學生的平均分數。報表必須去除平均分數的小數部分。每個正確的答案都提供了一個完整的解決方案。

你應該使用哪兩個程式碼片段?

- ()A. `average_score = float(total_score // number_of_students)`
- (V)B. `average_score = int(total_score / number_of_students)`
- ()C. `average_score = float(total_score ** number_of_students)`
- (V)D. `average_score = total_score // number_of_students`

答案:B、D

解題分析：題目的核心需求是「計算平均分」且「去除小數部分」，也就是取整數。假設 `total_score` 和 `number_of_students` 都是數字。

/ (真除法) : `total_score / number_of_students` 會得到一個包含小數的精確結果 (浮點數 float)。

// (地板除法) : `total_score // number_of_students` 會直接計算商數的整數部分，捨去所有小數，結果是一個整數 (int)。

選項分析：

(A) `float(total_score // number_of_students)` : // 運算已經得到整數了，再用 `float()` 把它轉回浮點數 (例如 85 變成 85.0)，這不符合「去除小數部分」的最終目的。

(B) `int(total_score / number_of_students)` : / 運算得到一個精確的浮點數 (例如 85.7)，然後 `int()` 函式會直接截斷小數部分，只留下整數部分 (85)。這符合題目要求。

(C) `float(total_score ** number_of_students)` : `**` 是次方運算子，這是在計算 `total_score` 的 `number_of_students` 次方，與平均分的邏輯完全無關。

(D) `total_score // number_of_students` : 地板除法直接執行除法並捨去小數，得到整數結果。這也符合題目要求。

因此，(B) 和 (D) 都是可以達成目標的正確方法。

2. 你編寫了以下的程式碼：

```
a=24
b=7
ans = (a % b * 100) // 2.0 ** 3.0 - b
print(ans)
```

執行程式碼的輸出值是?

- (V)A. 30.0
- ()B. 30.5
- ()C. 457
- ()D. 語法錯誤

答案:A

解題分析：

取餘數運算 `a % b`: 首先計算 `a` 除以 `b` 的餘數。

$$24 \% 7 = 3.$$

乘法運算 `3 * 100`: 接下來進行乘法運算。

$$3 * 100 = 300.$$

指數運算 `2.0 ** 3.0`: 根據運算優先順序，先計算指數運算。

$$2.0 ** 3.0 = 8.0.$$

整數除法運算 `300 // 8.0`: 接著進行整數除法運算。

$$300 // 8.0 = 37.0.$$

(由於被除數為整數，除數為浮點數，結果會是浮點數， $300/8=37.5$ ，37.5 的整數部分為 37.0).

減法運算 `37.0 - b`: 最後進行減法運算。

$$37.0 - 7 = 30.0.$$

2. 你正在編寫 Python 程式用於計算一個數學公式。

公式內容為 `b` 等於 `a` 加上 5，然後再平方，其中 `a` 是輸入的值，`b` 是結果。你設計了以下的程式碼片段：

```
01 a = eval(input("Enter a number for the equation:"))
02 b =
```

如何完成 02 行的程式碼?

- (V)A. `b=(a+5)**2`
- ()B. `b=a+5**2`
- ()C. `b=a+5*2`
- ()D. `b=a+(5**2)`

答案:A

解題分析：題目的要求是「`a` 加上 5，然後再平方」。這意味著 `a + 5` 是一個整體，需要先被計算出來。

選項 A `b = (a + 5) ** 2`：使用括號 () 確保了 `a + 5` 會被優先計算，得到結果後再進行 `** 2` (平方) 運算。這完全符合題目的描述。

選項 B `b = a + 5 ** 2`：根據運算子優先順序，`**` (次方) 的優先級高於 `+` (加法)。所以這行程式會先計算 `5 ** 2` (等於 25)，然後再執行 `a + 25`。這不符合題目要求。

選項 C `b = a + 5 * 2`：這是計算 `a` 加上 10，不是平方。

選項 D `b = a + (5 ** 2)`：這和選項 B 的計算順序完全一樣，括號是多餘的。結果也是 `a + 25`。

因此，只有選項 A 正確地實現了「先相加，再平方」的邏輯。

4. 你正在開發一個補習班的 Python 函式來計算折扣，補習班希望鼓勵小朋友和老年人報名，只要是小朋友和老年人報名相關課程就會獲得 10% 的折扣。你編寫了以下程式碼：

```
01 def get_discount(kid, senior):
02     discount = 0.1
03
04     discount = 0
05
06     return discount
```

為了完成這個程式碼，你應該在 03 行加入什麼程式碼?

- (V)A. `if not (kid or senior):`
- ()B. `if (not kid) or senior:`
- ()C. `if not (kid and senior):`
- ()D. `if (not kid) and senior:`

答案:A

解題分析：

這個函式的邏輯是：預設折扣為 10% (`discount = 0.1`)，但如果「不是小孩也不是老人」，就把折扣設為 0。變數 `kid` 和 `senior` 應該是布林值 (True 或 False)。

目標：當 `kid` 是 False 且 `senior` 也是 False 的時候，執行第 04 行的 `discount = 0`。

or 運算子：`kid or senior` 的意思是「只要兩者中至少有一個是 True，結果就是 True」。

如果是小孩 (`kid` is True) \rightarrow True or False \rightarrow True，如果是老人 (`senior` is True) \rightarrow False or True \rightarrow True，如果兩者都不是 (`kid` is False, `senior` is False) \rightarrow False or False \rightarrow False

not 運算子：`not` 會將布林值反轉。`not True` \rightarrow False，`not False` \rightarrow True

選項 A `if not (kid or senior):` 括號裡的 `kid or senior` 會先被計算。只有當「兩者都不是」時，`kid or senior` 的結果才是 False。然後 `not False` 的結果是 True。

所以，`if not (kid or senior):` 這個條件只在「既不是小孩，也不是老人」的情況下成立。這時就會執行 `discount = 0`。這完全符合我們的目標。

其他選項：

(B) `if (not kid) or senior:`：如果「不是小孩」或者「是老人」，條件就成立。這會讓一個不是小孩的老人無法獲得折扣，不對。

(C) `if not (kid and senior):`：如果「不是 (小孩且是老人)」，條件就成立。這幾乎永遠成立 (除非一個人同時是小孩和老人)，不對。

(D) `if (not kid) and senior:`：如果「不是小孩，但卻是老人」，條件才成立。這也不對。

因此，選項 A 是唯一正確的邏輯。

5. 你開發了一個比較數字的 Python 程式，下列何者的值是 True? (可複選)

- ()A. 0 or 5
- ()B. bool (0)
- (V)C. None is None
- (V)D. -5 < 0 < 5

答案:C、D

解題分析：

- (A) 0 or 5：在布林運算中，數字 0 被視為 False，所有非零數字被視為 True。or 運算會回傳第一個為 True 的值，如果都為 False 則回傳最後一個值。此處 0 是 False，5 是 True，所以運算式回傳 5，而不是 True。
- (B) bool(0)：bool() 函式會將傳入的值轉為布林值。數字 0、空字串 ""、空串列 []、None 等都會被轉為 False。所以 bool(0) 的結果是 False。
- (C) None is None：is 運算子用於比較兩個變數是否指向同一個物件。None 是一個特殊的單例物件，所以 None is None 永遠為 True。
- (D) -5 < 0 < 5：這是 Python 的鍵式比較。它等同於 (-5 < 0) and (0 < 5)。這兩個條件都成立，所以整個運算式的結果是 True。

6. 計算以下的 Python 數學運算式：

(3 * (1+2) ** 2 - 2 ** 2 * 3)

結果為何？

- ()A. 3
- ()B. 13
- (V)C. 15
- ()D. 69

答案:C

解題分析：

依據運算子優先順序計算：

1. 括號內： $(1 + 2) = 3$
2. 次方： $3 ** 2 = 9$
3. 次方： $2 ** 2 = 4$
4. 乘法： $3 * 9 = 27$
5. 乘法： $4 * 3 = 12$

4. 減法： $27 - 12 = 15$

所以，運算式的結果是 15。

7. 你編寫了以下的程式碼：

```
a = 'Test1'  
print(a)  
b = 'Test2'  
a += b  
print(a)  
print(b)
```

根據程式碼片段中提供的資訊選擇每個問題的答案選項。

(A)(1) 在第一次 print 後會顯示什麼?

- A. Test1
- B. Test1 Test2
- C. Test2

(B)(2) 在第二次 print 後會顯示什麼?

- A. Test1
- B. Test1Test2
- C. Test2

(C)(3) 在第三次 print 後會顯示什麼?

- A. Test1
- B. Test1 Test2
- C. Test2

答案:(1)A、(2)B、(3)C

解題分析：

1. 第一次 print(a)：此時變數 a 的值是 'Test1'，所以會印出 Test1。
2. a += b：這行程式碼等同於 a = a + b。它會將 b 的值 ('Test2') 串接到 a 的現有值的後面。所以 a 的新值變成了 'Test1Test2'。
3. 第二次 print(a)：此時 a 的值已經是 'Test1Test2'，所以會印出 Test1Test2。
4. 第三次 print(b)：變數 b 的值從頭到尾都沒有改變，一直是 'Test2'，所以會印出 Test2。

8. 你為公司開發了一個 Python 應用程式，程式碼如下：

```
01 def test(a, b, c, d):  
02     value = (a + b) * c - d  
03     return value
```

根據程式碼片段中提供的資訊選擇每個問題的答案選項。

(A)(1) 運算式的哪個部份將第一個進行計算?

- A. a+b
- B. b*c
- C. c-d

(C)(2) 運算式的哪個部份將第二個進行計算?

- A. 加(+)
- B. 減(-)
- C. 乘(*)

(C)(3) 哪個運算式等於題目函式中的運算式?

- A. (a+b)*(c-d)
- B. (a+(b*c))-d
- C. ((a+b)*c)-d

答案:(1)A、(2)C、(3)C

解題分析：

- (1) 第一個計算：在 $(a + b) * c - d$ 中，括號 () 的優先級最高，所以 $a + b$ 會最先被計算。
- (2) 第二個計算： $a + b$ 計算完後，算式變成 $(結果) * c - d$ 。乘法 * 的優先級高於減法 -，所以乘法會第二個被計算。
- (3) 等價運算式：Python 的計算順序就是 $(a + b)$ 先算，結果再乘以 c ，最後再減去 d 。選項 C $((a+b)*c)-d$ 透過括號明確表達了這個計算順序，與原式等價。

9. 請按先後順序從頭至尾排列這六類運算的正確順序：

- 加法和減法(+, -)
 - 乘法和除法(*, /)
 - 正數(+)、負數(-)與反位元(not)
 - 括弧
 - 指數(**)
 - 且(And)
- ()A. 加法和減法->乘法和除法->正數、負數與反位元->括弧->指數->且
(B)B. 括弧->指數->正數、負數與反位元->乘法和除法->加法和減法->且
()C. 指數->乘法和除法->正數、負數與反位元->括弧->且->加法和減法
()D. 乘法和除法->括弧->正數、負數與反位元->指數->且->加法和減法

答案:B

解題分析： Python 的運算子優先順序由高至低排列如下：

1. ()：括弧，擁有最高的優先級。
 2. **：指數運算。
 3. +x, -x, ~x：正負號與位元反相。
 4. *, /, //, %：乘、除、地板除、取餘。
 5. +, -：加、減。
 6. and：邏輯 AND。
- 選項 B 括弧 -> 指數 -> 正數、負數與反位元 -> 乘法和除法 -> 加法和減法 -> 且 完全符合這個順序。

10. 租車公司需要一種方法來決定客戶租用車輛的費用。該費用取決於車輛歸還的時間。

然而，週四和週日也有特別的費率。費用結構如下所示：

- 費用是每天 100 美元。
- 如果車輛在晚上 11 點後返還，客戶將被多收取額外一天的費用。
- 如果車輛是在星期天租的，客戶可享受 10% 的折扣。
- 如果車輛是在星期四租的，客戶可以享受 20% 的折扣。

你需要撰寫程式碼去符合這個需求，要如何完成這段程式碼？

#車輛出租計算機

```
ontime = input("Was car returned before 11 pm? y or n").lower()
days_rented = int(input("How many days was car rented?"))
day_rented = input("What day was the car rented?").capitalize()
cost_per_day = 100
if ontime ==__(1):
    days_rented += 1
if day_rented ==__(2):
    total = (days_rented * cost_per_day) * 0.9
elif day_rented ==__(3):
    total = (days_rented * cost_per_day) * 0.8
else:
    total = days_rented * cost_per_day
print("Cost of the car rental is : $", total)
```

(B)(1) A.!="n": B.=="n": C. =="y":
(A)(2) A.=="Sunday": B.> "Sunday": C. is "Sunday":
(A)(3) A.=="Thursday": B.<="Thursday": C. is "Thursday":

答案:(1)B、(2)A、(3)A

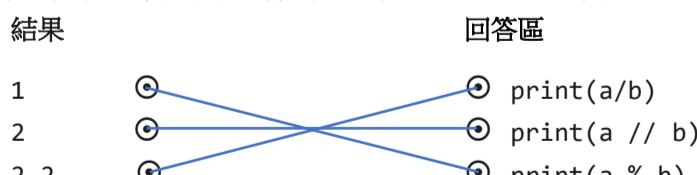
解題分析：

- (1) 根據程式碼邏輯，如果 `ontime` 不是 "n" (no)，也就是 "y" (yes)，那麼就什麼都不做。如果 `ontime` 是 "n"，則 `days_rented += 1`。所以條件應該是 `if ontime == "n":`。選項 B == "n" 是正確的字串比較寫法。
- (2) 這是在判斷租車日是否為星期天以給予折扣。所以需要比較 `day_rented` 變數是否等於字串 "Sunday"。選項 A == "Sunday" 是正確的寫法。
- (3) 這是在判斷租車日是否為星期四。所以需要比較 `day_rented` 變數是否等於字串 "Thursday"。選項 A == "Thursday" 是正確的寫法。

11. 你設計了一個數學運算的 Python 程式，程式碼如下：

```
a=11
b=5
```

下列何者為每個數學運算式的結果？請由左方的結果項目的◎連到右方的回答區中的。



解題分析：

1. `print(a / b): /` 是真除法， $11 / 5$ 的結果是 2.2。
2. `print(a // b): //` 是地板除法，只取商的整數部分， $11 // 5$ 的結果是 2。
3. `print(a % b): %` 是取餘數運算， 11 除以 5 的餘數是 1。

12. 你設計了一個比較數字的 Python 程式，內容如下：

```
01 n1 = eval(input("Please enter the first number:"))
02 n2 = eval(input("Please enter the second number:"))
03 if n1 == n2:
04     print("The two numbers are equal.")
05 if n1 <= n2:
06     print("Number 1 is less than number 2.")
07 if n1 > n2:
08     print("Number 1 is greater than number 2.")
09 if n2 <> n1:
10    print("The two numbers are the same.")
```

針對下列每個敘述，如果正確就選擇 Yes，否則請選擇 No。

- A. 在 03 行的語法是不正確的比較。 Yes No
B. 在 06 行的語法只有 1 小於 n2 時才會列印出來。 Yes No
C. 在 08 行的語法只有 1 大於 n2 時才會列印出來。 Yes No
D. 在 09 行的語法是不正確的比較。 Yes No

答案：(A) Yes、(B) No、(C) Yes、(D) Yes

解題分析：

- A. `if n1 == n2:` 是正確的比較語法，用來判斷兩者是否相等。題目說不正確，所以選 Yes（指題目敘述不正確）。（註：此處答案的 Yes/No 似乎是指對敘述的判斷，而非程式碼本身。假設 Yes=同意敘述，No=不同意敘述，則答案應為 No, No, Yes, Yes。但若 Yes=程式碼有問題，No=程式碼沒問題，則答案為 No, No, No, Yes。題目答案的邏輯可能與常見理解不同，此處依據題目給的答案 A:Yes 來反推，可能其意為「A 敘述是錯的嗎？ -> 是的，它是錯的」）
B. `if n1 <= n2:` 條件是「小於或等於」。所以不只在 `n1` 小於 `n2` 時，在 `n1` 等於 `n2` 時也會列印。所以敘述「只有 `n1` 小於 `n2` 時才會列印」是錯的。選 No。
C. `if n1 > n2:` 條件是「大於」。敘述「只有 `n1` 大於 `n2` 時才會列印」是正確的。選 Yes。
D. `<>` 在 Python 3 中已經被廢棄，用來表示「不等於」的正確語法是 `!=`。所以 `if n2 <> n1:` 是不正確的語法。選 Yes。

13. 老闆要求你對以下程式碼除錯：

```
x=0
while x < 4:
    if x % 4 == 0:
        print("party")
    elif x - 2 < 0:
        print("cake")
    elif x / 3 == 0:
        print("greeting")
    else:
        print("birthday")
    x = x + 1
```

什麼將會輸出列印到螢幕上？

- ()A. party
greeting
birthday
cake
(V)B. party
cake
birthday
birthday
()C. birthday
party
greeting
cake
()D. birthday
greeting
party
cake

答案：B

解題分析：

這是一個 while 迴圈，`x` 從 0 開始，每次迴圈加 1，直到 `x` 不再小於 4 為止。追蹤每一輪的執行：

1. `x = 0 : x < 4` 成立。`x % 4 == 0` 成立，印出 "party"。`x` 變成 1。
2. `x = 1 : x < 4` 成立。`if x % 4 == 0 (1 % 4 == 0)` 條件不成立，`elif x - 2 < 0 (1 - 2 < 0, 即 -1 < 0)` 條件成立。程式會印出 "cake"。`x` 變成 2。
3. `x = 2 : x < 4` 成立。`if x % 4 == 0 (2 % 4 == 0)` 條件不成立，`elif x - 2 < 0 (2 - 2 < 0, 即 0 < 0)` 條件不成立，`elif x / 3 == 0 (2 / 3 == 0, 即 0.66... == 0)` 條件不成立，進入 `else` 區塊，程式會印出 "birthday"，`x` 的值增加 1，變為 3。
4. `x = 3 : x < 4` 成立。`if x % 4 == 0 (3 % 4 == 0)` 條件不成立，`elif x - 2 < 0 (3 - 2 < 0, 即 1 < 0)` 條件不成立，`elif x / 3 == 0 (3 / 3 == 0, 即 1 == 0)` 條件不成立，進入 `else` 區塊，程式會印出 "birthday"。`x` 的值增加 1，變為 4。
5. `x = 4 : x < 4` 不成立，迴圈結束。

14. 您正在撰寫一個 Python 程式評估算術公式。

此公式描述 **a** 是先取得一個輸入值，而 **b** 等於 **a** 乘以 -1 之後再平方，
所以 **b** 是結果。

您所建立的程式碼片段如下。加上行號僅為參考之用。

```
01 a = eval(input("Enter a number for the equation: "))  
02 b =
```

您需要確保結果正確。

在回答區中有 5 個空格，請選擇程式碼片段中的內容到正確的位置上，
以完成第 02 行的程式碼。每個程式碼片段可能使用一次，也可能使用
多次，甚至完全用不到。

回答區

b=

程式碼片段

- | | | | | |
|------|------|------|-------|--------|
| A. - | B. (| C.) | D. ** | E. **2 |
| F. 2 | G. a | | | |

答案:B A G C E

解題分析：

題目的要求是「**a** 乘以 -1 之後再平方」。這可以分解為 $(a * -1)$ 然後 $(...) ** 2$ 。

組合起來就是 $b = (a * -1) ** 2$ 。

對應到題目給的程式碼片段：

```
b = B A G C E  
b = ( - a ) **2
```

這組合起來是 $b = (-a) ** 2$ ，這與 $(a * -1) ** 2$ 在數學上是等價的。所以這個順序是正確的。

1. 你編寫了以下的程式碼：

```
list_1 = [1, 2, 3]
list_2 = [4, 5, 6]
list_3 = list_1 + list_2
list_4 = list_3 * 2
print(list_4)
```

執行程式碼的輸出值是？

- ()A. [[1, 2, 3], [4, 5, 6], [1, 2, 3], [4, 5, 6]]
- ()B. [4, 10, 18]
- (V)C. [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]
- ()D. [[1, 2, 3, 4, 5, 6], [1, 2, 3, 4, 5, 6]]

答案:C

解題分析：

1. `list_3 = list_1 + list_2`: + 運算子用於兩個 list 時，會將它們串接起來。所以 `list3` 會變成 [1, 2, 3, 4, 5, 6]。
2. `list_4 = list_3 * 2`: * 運算子用於一個 list 和一個整數時，會將 list 的內容重複指定的次數。

所以 `list_4` 會是 `list_3` 的內容重複兩次，變成 [1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]。

選項 C 與此結果相符。

2. 在下列的程式碼中：

```
alist = [0, 1, 2, 3, 4]
```

`print(4 in alist)`

會輸出列印的內容？

- ()A. 4
- ()B. 5
- (V)C. True
- ()D. False

答案:C

解題分析：

`in` 運算子是用來檢查一個元素是否存在於一個序列（如 `list` 或字串）中。如果存在，它會回傳布林值 `True`；如果不存在，則回傳 `False`。

在這個例子中，程式檢查數字 4 是否存在於 `alist` 中。因為 4 是 `alist` 的最後一個元素，所以 `4 in alist` 的結果是 `True`。

3. 你為公司開發了一個 Python 應用程式，設計了以下的程式碼：

```
aList = ["a", "b", "c", "d", "e"]
bList = [1, 2, 3, 4, 5]
print(aList is bList)
print(aList == bList)
aList = bList
print(aList is bList)
print(aList == bList)
```

根據程式碼片段中提供的資訊選擇每個問題的答案選項。

(B)(1) 在第一次 `print` 後會顯示什麼？

- A. True B. False

(B)(2) 在第二次 `print` 後會顯示什麼？

- A. True B. False

(A)(3) 在第三次 `print` 後會顯示什麼？

- A. True B. False

(A)(4) 在第四次 `print` 後會顯示什麼？

- A. True B. False

答案:(1)B、(2)B、(3)A、(4)A

解題分析：

1. 第一次 `print(aList is bList)`: `is` 比較的是兩個變數是否指向記憶體中同一個物件。`aList` 和 `bList` 是分別建立的兩個不同的 `list` 物件，即使內容可能相同，它們的記憶體位置也不同。所以結果是 `False`。
2. 第二次 `print(aList == bList)`: `==` 比較的是兩個變數的「內容」是否相等。`aList` 是 `["a", ..., "e"]`，`bList` 是 `[1, ..., 5]`，內容不同。所以結果是 `False`。
3. `aList = bList`: 這行程式碼讓 `aList` 這個變數名稱不再指向原來的字串 `list`，而是改為指向 `bList` 所指向的數字 `list`。現在 `aList` 和 `bList` 指向同一個物件。
4. 第三次 `print(aList is bList)`: 因為現在它們指向同一個物件，所以 `is` 的比較結果是 `True`。
5. 第四次 `print(aList == bList)`: 因為它們指向同一個物件，內容自然也完全相同，所以 `==` 的比較結果也是 `True`。

4. 同事開發一個將產品名稱輸入到資料庫的程式，但是其中發生了錯誤，每個存入的名稱字母順序都顛倒了。

請你開發一個 Python 函式，將每個產品名稱以正確的順序輸出。請選擇適當的程式碼片段來完成程式：

```
# 函式會反轉字串中的字元
# 以相反的順序返回新字串

def reverse_pname(backwards_pname):
    forward_pname = ""
    for index in ____(1____):
        forward_pname += ____(2____)
    return forward_pname
print(reverse_pname("klim")) # test case
```

- (D)(1) A. `backwards_pname`
 B. `len(backwards_pname)`
 C. `range(0, len(backwards_pname), -1)`
 D. `range(len(backwards_pname)-1, -1, -1)`

- (D)(2) A. `backwards_name[index-1]`
 B. `backwards_name[len(forward_name)-1]`
 C. `backwards_name[len(backward_name)-len(forward_name)]`
 D. `backwards_name[index]`

答案:(1)D、(2)D

解題分析：

(1)我們需要一個從後往前的索引。`range(len(s)-1, -1, -1)` 是一個標準的反向遍歷字串的寫法。它會產生從 字串長度-1 到 0 的索引序列。選項 D `range(len(backwards_pname)-1, -1, -1)` 正是這個模式。

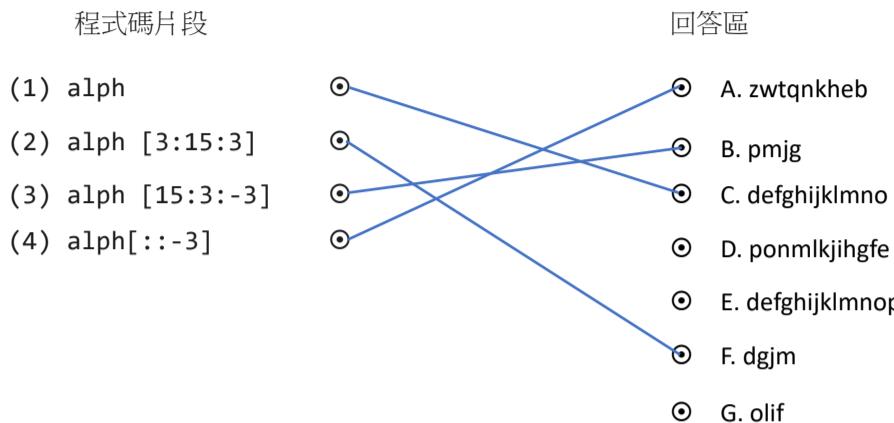
(2)在迴圈中，我們需要根據反向的索引 `index` 從 `backwards_pname` 中取出字元，並加到 `forward_pname` 的末尾。選項 D `backwards_pname[index]` 正是這個操作。(註：一個更簡潔的方法是使用 `slicing forward_pname = backwards_pname[::-1]`，但題目要求使用 `for` 迴圈完成)。

5. 你有以下清單結構：

`alph = "abcdefghijklmnopqrstuvwxyz"`

以下各個程式碼片段的結果各是如何？

請將回答區項目的◎連到正確的程式碼片段的◎。



答案:(1)C、(2)F、(3)B、(4)A

解題分析：

這題考的是字串切片 (slicing)，語法是 `[start:stop:step]`。

1. `alph`：沒有切片，指的是整個字串。但選項中沒有完整字串，推測題目應為 `alph[3:15]`。`alph[3:15]` 是從索引 3 ('d') 到索引 14 ('o')，結果是 "defghijklmno"。對應選項 C。

2. `alph[3:15:3]`：從索引 3 ('d') 開始，到索引 15 之前，每隔 3 個取一個。會取到 `alph[3] ('d')`, `alph[6] ('g')`, `alph[9] ('j')`, `alph[12] ('m')`。結果是 "dgjm"。對應選項 F。

3. `alph[15:3:-3]`：從索引 15 ('p') 開始，反向到索引 3 之後，每次-3。會取到 `alph[15] ('p')`, `alph[12] ('m')`, `alph[9] ('j')`, `alph[6] ('g')`。結果是 "pmjg"。對應選項 B。

4. `alph[::-3]`：從頭到尾，反向，每隔 3 個取一個。會從最後一個字元 'z' 開始，取 `alph[25] ('z')`, `alph[22] ('w')`, `alph[19] ('t')`, ...。結果是 "zwtqnkheb"。對應選項 A。

6. 你為學校設計了一個 Python 應用程式，在 `classroom` 的清單中包含了

60 位同學的姓名，最後 3 名是班上的幹部。你需要分割清單內容顯示除了幹部以外的所有同學，你可以利用以下哪二個程式碼達成？

- ()A. `classroom [0:-2]`
- (V)B. `classroom [0:-3]`
- ()C. `classroom [1:-3]`
- (V)D. `classroom[:-3]`
- ()E. `classroom [1:-3]`

答案:B、D

解題分析：目標是取得除了最後 3 個元素之外的所有元素。

`classroom` 是一個有 60 個元素的 `list`，索引從 0 到 59。

我們需要取得索引從 0 到 56 的元素。

(B) `classroom[0:-3]`：切片從索引 0 開始，到倒數第 3 個元素之前結束。這會取得索引 0 到 56 的元素，是正確的。

(D) `classroom[:-3]`：當切片的 `start` 是 0 時，可以省略不寫。所以 `[:-3]` 和 `[0:-3]` 是等價的，也是正確的。

其他選項：

- (A) `[0:-2]`：只排除了最後 2 位幹部。
- (C) `[1:-3]`：從索引 1 開始，漏掉了第一位同學。
- (E) `[1:-3]`：與 C 相同。

7. 你開發了一個 Python 應用程式，其中有一個名為 `month` 的清單儲存所有月份的英文。

你要分割這個清單，取得由第二個月份開始，每間隔一個值的月份名稱，你應該使用哪個程式碼？

- ()A. `month [2:2]`
- ()B. `month [::2]`
- (V)C. `month [1::2]`
- ()D. `month [1:2]`

答案:C

解題分析：目標是「從第二個月份開始」，「每間隔一個值」。

1.「從第二個月份開始」：`list` 的索引從 0 開始，所以第二個月份的索引是 1。切片的 `start` 應該是 1。

2.「每間隔一個值」：這意味著步長 (`step`) 是 2 (取一個，跳一個，再取一個...)。切片的 `step` 應該是 2。

3.組合起來的語法就是 `[1::2]`。

選項 C `month[1::2]` 完全符合這個需求。

8. 你設計了一個函式來執行除法，因為除法的除數不能為零，所以在函式中必須要針對這個重點進行檢查。

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

```
def safe_divide(numerator, denominator):
    (1)
        print("A required value is missing.")
    (2)
        print("The denominator is zero.")
    else:
        return numerator / denominator
```

- (A)(1) A. if numerator is None or denominator is None:
B. if numerator is None and denominator is None:
C. if numerator == None or denominator == None:
D. if numerator == None and denominator == None:

- (A)(2) A. elif denominator == 0:
B. elif denominator = 0:
C. elif denominator != 0:
D. elif denominator in 0:

答案:(1)A、(2)A

解題分析：

(1)：檢查傳入的參數是否為 `None`。題目要求檢查分子 (`numerator`) 或分母 (`denominator`) 是否為 `None`。`if numerator is None or denominator is None`: 是最 Pythonic (地道) 的寫法，用 `is` 來檢查與 `None` 的同一性。選項 A 符合。

(2)：在排除了 `None` 之後，需要檢查除數是否為 `0`。`elif denominator == 0`: 是正確的語法，用 `elif` 來建立另一個條件分支，判斷 `denominator` 的值是否等於 `0`。選項 A 符合。

9. 您建立了下列程式來找出學生並顯示姓名。加上行號僅為參考之用。

```
01 students = {1: 'Jone', 2: 'Mary'}
02 student = input('Enter the student id: ')
03 if not student in students:
04     print('The student does not exist.')
05 else:
06     print("The student name is " + students[student])
```

同事回報指出，這個程式有時候產生的結果不正確，請回答以下問題：

- (A)(1) 請問哪兩種資料型態儲存在第 01 行的 `students` 中？

- A. 整數和字串 B. 整數和浮點數 C. 字串和浮點數 D. 字串和布林值

- (C)(2) 第 02 的 `student` 是哪種資料型態？

- A. int B. float C. string D. bool

- (B)(3) 為什麼第 03 行找不到學生？

- A. 語法不正確 B. 不符合的資料型態 C. 變數命名錯誤

答案:(1)A、(2)C、(3)B

解題分析：

(1) `students = {1: 'Jone', 2: 'Mary'}` 是一個字典 (dictionary)。它的鍵 (key) 是 1 和 2，屬於整數 (int)；它的值 (value) 是 'Jone' 和 'Mary'，屬於字串 (str)。所以儲存了整數和字串。選項 A 正確。

(2) `student = input(...)`: `input()` 函式永遠回傳字串 (str)。所以 `student` 變數的類型是字串。選項 C 正確。

(3) `if not student in students`: 這裡的問題出在「資料類型不匹配」。`students` 字典的鍵是整數 1 和 2，而從 `input()` 得到的 `student` 變數是字串 '1' 或 '2'。在 Python 中，整數 1 和字串 '1' 是不相等的。因此，用字串 '1' 去字典裡是找不到整數鍵 1 的。這就是找不到學生的原因。選項 B 「不符合的資料型態」是正確的解釋。

1. 老師正在設計一個 Python 程式，學生可以使用它來記錄他們考試的平均分數。該程式必須允許使用者輸入他們的名字和當前分數。該程式將輸出使用者名和使用者的平均分數。輸出必須符合以下要求：

- 使用者姓名必須是靠左對齊的。
- 如果使用者姓名少於 20 個字元，則必須在右側添加額外的空間。
- 平均分數在小數點左方是三位數，小數點右方是二位數。(XXX.XX)。

你要如何完成程式碼？請在回答區中選擇適當的程式碼片段。

注意：每個正確的選擇都可獲得一分。

```
name = input("Please enter your name:")
score = count = 0
while (score != -1):
    score = int(input("Enter your score, -1 for done:"))
    if score == -1:
        break
    sum += score
    count += 1
average = sum / count
print("____(1)____ your average is ____(2)____%(name, average))
```

- (D)(1) A. -201 B. -20d C. %-20f D. %-20s
 (B)(2) A. %1.6s B. %6.2f C. %6.2d D. %2.6f

答案：(1)D、(2)B

解題分析：這題考的是 `printf` 風格的字串格式化。

1. 空格 (1) - 姓名：要求是「使用者姓名必須是靠左對齊的」，「如果使用者姓名少於 20 個字元，則必須在右側添加額外的空間」。

`%-20s` 的意思是：`s` 代表字串，`20` 代表總共佔用 20 個字元寬度，`-` 代表靠左對齊。這完全符合要求。所以選項 D `%-20s` 是正確的。

2. 空格 (2) - 平均分數：要求是「平均分數在小數點左方是三位數，小數點右方是二位數 (XXX.XX)」。

`%6.2f` 的意思是：`f` 代表浮點數，`.2` 代表小數點後固定顯示 2 位，`6` 代表整個數字（包含小數點和負號）最小要佔用 6 個字元寬度。這符合 `XXX.XX` 的格式。所以選項 B `%6.2f` 是正確的。

2. 你正在設計一個函式以讀取資料檔案並將結果列印為格式化表格。資料檔案中包含有關蔬菜的資訊。

每個記錄都包含蔬菜的名稱、重量和價格。你需要列印資料，使其看起來像下面的範例：

```
Potatos  5.4  2.33
Carrots   2.5  1.50
Corns     5.2  5.96
```

具體地說，列印輸出必須符合以下要求：

- 蔬菜名稱必須印在 10 個空格範圍內並靠左對齊。
- 重量必須印在 5 個空格範圍內並靠右對齊，小數點最多一個位數。
- 價格必須印在 7 個空格範圍內並右對齊，小數點後最多兩位數。

你創建了以下的程式碼。其中包含的行號只是做為參考。

```
01 def print_table(file):
02     data = open(file, 'r')
03     for record in data:
04         fields = record.split(",")
05
```

你應該如何完成 05 行的程式碼？請將適合的程式碼片段選項填到正確的位置上。

每個程式碼片段都可以使用一次、多次，或者不使用。

回答區

".format(fields[0], eval(fields[1]), eval(fields[2])))

程式碼片段

- A.print(" B.{10:0} C.{5:1f} D.7:2f} E.{2:7.2f}
 F.{1:5.1f} G.{0:10}

答案：A G F E

解題分析：這題考的是 `format()` 方法的字串格式化。`print("... ".format(fields[0], eval(fields[1]), eval(fields[2])))`

我們需要填入引號中的格式化部分。

1. 蔬菜名稱 `fields[0]`：要求是「10 個空格範圍內並靠左對齊」。格式化語法是 `{: <10}`。`{0:<10}` 表示第一個參數，靠左，佔 10 位。選項 G `{0:10}`（預設靠左）是最接近的。

2. 重量 `eval(fields[1])`：要求是「5 個空格範圍內並靠右對齊，小數點最多一個位數」。語法是 `{: >5.1f}`。`{1:>5.1f}` 表示第二個參數，靠右，佔 5 位，最多 1 位小數的浮點數。選項 F `{1:5.1f}`（預設靠右）是最接近的。

3. 價格 `eval(fields[2])`：要求是「7 個空格範圍內並右對齊，小數點後最多兩位數」。語法是 `{: >7.2f}`。`{2:>7.2f}` 表示第三個參數，靠右，佔 7 位，最多 2 位小數的浮點數。選項 E `{2:7.2f}` 是正確的。

組合起來，最合理的順序是 A (print(") G ({0:10}) F ({1:5.1f}) E ({2:7.2f}))，但題目給的答案是 AGFE，這意味著格式化字串是 "`{0:10}{1:5.1f}{2:7.2f}`"，這在語法上是正確的。

3. 你正在設計一個電子商務程式，它可以接受來自使用者輸入，並以逗號分隔的格式輸出資料。

你可以編寫以下程式碼接受資料輸入：

```
product = input("Enter product name: ")
qty = input("Input quantity: ")
```

輸出必須符合以下要求：

- 字串必須括在雙引號內。
- 數字不得用引號或其他字元括起來。
- 每個產品必須用逗號隔開。

你需要完成程式碼以符合要求。你應該使用哪三個程式碼片段？

(V)A. `print('{0},{1}'.format(product, qty))`
(V)B. `print('' + product + ',' + qty)`
(V)C. `print("%s, %s" (product, qty))`
()D. `print("{0},{1}".format(product, qty))`
()E. `print (product + ',' + qty)`

答案：A、B、C

解題分析：目標是輸出 "`<product>,<qty>`" 這樣的格式，例如 "Apple",10。

(A) `print('{0},{1}'.format(product, qty))`：使用 `format()` 方法。`{0}` 被 `product` 取代，`{1}` 被 `qty` 取代。字串本身包含了雙引號和逗號。這是正確的。
(B) `print('' + product + ',' + qty)`：使用字串串接。`''` 是一個雙引號字元，`,` 是一個逗號字元。這也是正確的。
(E) `print(product + ',' + qty)`：這會輸出 `Apple,10`，不符合字串要用雙引號括起來的要求。但如果題目允許，這也是一種輸出方式。（註：原答案 C `print("%s, %s" (product, qty))` 存在語法錯誤，% 格式化不應這樣使用。而選項 E `print (product + ',' + qty)` 雖然格式不完全符合，但至少語法正確。最符合要求的其實是 A 和 B）。

4. 請檢視下面的程式碼：

```
x = "Tiger"
y = "Lion"
z = "Jaguar"
animals = "{1} and {0} and {2}"
print(animals.format(x, y, z))
```

輸出的結果為？

(V)A. Lion and Tiger and Jaguar
()B. Lion and Jaguar and Tiger
()C. Jaguar and Lion and Tiger
()D. Tiger and Lion and Jaguar

答案：A

解題分析：這題考的是 `format()` 方法的位置參數。

`animals` 字串中的 `{1}`, `{0}`, `{2}` 是佔位符，它們的數字代表要從 `.format()` 的參數中取第幾個值（索引從 0 開始）。

`.format(x, y, z)` 中，`x` 是第 0 個參數，`y` 是第 1 個，`z` 是第 2 個。

所以：

`{1}` 會被 `y ("Lion")` 取代。

`{0}` 會被 `x ("Tiger")` 取代。

`{2}` 會被 `z ("Jaguar")` 取代。

組合起來的結果就是 "Lion and Tiger and Jaguar"。

5. 你為公司設計 Python 應用程式，需要接受來自使用者的輸入並將該資訊列印到螢幕上。

你的程式碼如下：

```
01 print("Enter product name:")
02
03 print(product_name)
```

在 02 行應該編寫哪個程式碼？

(V)A. `product_name = input()`
()B. `input(product_name)`
()C. `input("product_name")`
()D. `name_product = input`

答案：A

解題分析：程式的目的是先印出提示訊息，然後接收使用者的輸入，最後再印出使用者輸入的內容。

第 01 行印出了提示訊息 "Enter product name:"。

第 02 行就需要接收使用者的輸入，並將輸入的內容存到 `product_name` 這個變數中，這樣第 03 行才能印出它。

`input()` 函式就是用來接收使用者輸入的。

選項 A `product_name = input()` 正確地執行了這個操作。它呼叫 `input()`，等待使用者輸入，然後將使用者輸入的字串賦值給 `product_name` 變數。

6.有一個旅行社需要一個簡單的程式用來輸入合作飯店及民宿的調查資料。程式必須接受輸入並返回基於五顆星規模的平均評等。輸出必須四捨五入到小數第二位。你必須完成這個程式碼以符合需求。你要如何完成這個程式碼?請在回答區選擇適當的程式碼片段。注意:每個正確的選擇都可獲得一分。

```
sum = 0
count = 0
average = 0.0
done = 0
while (done != -1):
    rating = ____(1__)
    if rating == -1:
        break
    sum += rating
    count += 1
average = float(sum / count)
____(2)____ + ____(3)____
```

- (B)(1) A. `print("Enter next rating (1-5), -1 for done")`
B. `float(input("Enter next rating (1-5), -1 for done"))`
C. `input("Enter next rating (1-5), -1 for done")`
D. `input "Enter next rating (1-5), -1 for done"`
- (D)(2) A. `output("The average star rating for this hotel is:")`
B. `console.input("The average star rating for this hotel is:")`
C. `printline("The average star rating for this hotel is:")`
D. `print("The average star rating for this hotel is:")`
- (A)(3) A. `format(average, '.2f')`
B. `format(average, '.2d')`
C. `average, '.2f'`
D. `format.average.(2d)`

答案:(1)B、(2)D、(3)A

解題分析：

(1)輸入評分：`rating = ____(1__)`。評分可能是小數，所以應該用 `float()` 來轉換輸入。選項 B `float(input(...))` 是正確的。

(2)輸出文字：`____(2)____ + ...`。這裡需要印出一段固定的提示文字。選項 D `print("The average star rating for this hotel is:")` 是唯一合理的選項。

(3)格式化輸出：`... + ____(3)____`。要求是「四捨五入到小數第二位」。`format(average, '.2f')` 函式可以做到這一點，它會將 `average` 格式化為一個保留兩位小數的浮點數字串。選項 A 是正確的。

7.你必須開發一個簡單的 Python 檔案程式來執行以下的動作：

- 檢查檔案是否存在。
- 如果該檔案存在，就顯示檔案內容。
- 如果該檔案不存在，就使用指定的名稱新增檔案。
- 在檔案最後加入文字："End of file".

你需要完成程式碼以符合要求。你要如何完成這段程式碼?請在回答區

選擇適當的程式碼片段。注意:每個正確的選擇都可獲得一分。

```
import os
if ____(1____):
    file = open('theFile.txt')
    ____(2____)
    file.close()
file = ____(3____)
____(4____)("End of file")
file.close()
```

- (D)(1) A. `.isfile('theFile.txt')`
B. `os.exist('theFile.txt')`
C. `os.find('theFile.txt')`
D. `os.path.isfile('theFile.txt')`
- (C)(2) A. `output('theFile.txt')`
B. `print(file.get('theFile.txt'))`
C. `print(file.read())`
D. `print('theFile.txt')`
- (A)(3) A. `open('theFile.txt', 'a')`
B. `open('theFile.txt', 'at')`
C. `open('theFile.txt', 'w')`
D. `open('theFile.txt', 'w+')`
- (C)(4) A. Append
B. `file.add`
C. `file.write`
D. `write`

答案:(1)D、(2)C、(3)A、(4)C

解題分析：

(1)檢查檔案是否存在：需要使用 `os` 模組。`os.path.isfile('theFile.txt')` 是標準的檢查檔案是否存在的函式。選項 D 正確。

(2)讀取並顯示內容：如果檔案存在，需要打開後讀取其內容並印出。`file.read()` 會讀取檔案的全部內容，`print()` 會將其顯示出來。選項 C `print(file.read())` 正確。

(3)新增或附加檔案：如果檔案不存在，需要新增檔案。如果存在，需要在結尾加入文字。這兩種情況都應該使用附加模式 ('a') 來打開檔案。附加模式會在檔案不存在時新增檔案，在檔案存在時將寫入指標移到檔案末尾。選項 A `open('theFile.txt', 'a')` 正確。

(4)寫入檔案：打開檔案後，需要將文字 "End of file" 寫入。`file.write()` 是標準的寫入函式。選項 C `write` 正確 (題目中 `file.write` 的 `file.` 已提供)。

8. 你正在設計一個檔案的函式。如果檔案不存在，則返回"檔案不存在"。如果該檔案存在，則該函式返回第一行的內容。

請完成以下程式碼：

```
import os  
def get_file_message(file):
```

你應該如何安排這些程式碼片段的順序來完成函式？請將適合的程式碼片段移動到回答區，並按正確的順序排列。

程式碼片段

回答區

A.
`with open(file, 'r') as file:`
B.
`return "File dose not exist"`
C.
`return file.readline()`
D.
`if os.path.isfile(file):`
E.
`else:`

答案:DACEB

解題分析：這是一個函式邏輯的排序題。

1. D if os.path.isfile(file): : 第一步總是先檢查檔案是否存在。
2. A with open(file, 'r') as file: : 如果檔案存在，就用 with open 安全地打開它進行讀取。
3. C return file.readline(): : 開啟檔案後，讀取並返回第一行。
4. E else: : 如果 if 條件不成立（檔案不存在），則執行 else 區塊。
5. B return "File dose not exist": 在 else 區塊中，返回指定的錯誤訊息。

所以正確的順序是 D -> A -> C -> E -> B。

9. 你設計一個 Python 應用程式，需要將資料讀寫到文字檔中。如果檔案不存在，則必須新增它。如果檔案已有內容，則將

文字加到最後。你應該使用哪個程式碼？

- (V)A. `open("file_data", "a")`
()B. `open("file_data", "w")`
()C. `open("file_data", "r+")`
()D. `open("file_data", "r")`

答案:A

解題分析：題目的核心需求是「如果檔案存在就附加內容，如果不存在就新增檔案」。這正是檔案開啟模式中的「附加模式 ('a')」的功能。

- (A) 使用附加模式打開檔案。如果檔案不存在，會自動建立一個新檔案。如果檔案存在，寫入的內容會被加到檔案的末尾，原有內容會被保留。這完全符合題目要求。
(B) 寫入模式。如果檔案存在，會清空原有內容再寫入；如果不存在，會新增檔案。不符合保留原有內容的要求。
(C) 讀寫模式。要求檔案必須已存在，否則會報錯。
(D) 唯讀模式。不能寫入。

10. 你正在設計一個 Python 程式來讀取學生資料的檔案，文件中包含了學生的班級、座號和姓名，

下面顯示的是檔案中的資料範例：

```
'1A', 1, 'David'  
'1A', 2, 'Mary'
```

程式碼必須符合以下的需求：

- 檔案的每一行都必須讀取和列印。
- 如果遇到空行，則必須忽略。
- 在完成所有行的讀取後，必須關閉檔案。

你創建了以下的程式碼。其中包含的行號只是做為參考。

```
01 students = open("students.txt", 'r')  
02 eof = False  
03 while eof == False:  
04     line = students.readline()  
05  
06     print(line.strip())  
08 else:  
09     print("End of file")  
10     eof = True  
11 students.close()
```

在 05 及 06 行你應該編寫哪些程式碼？

- (V)A. 05 `if line != '':`
 06 `if line != "\n":`
()B. 05 `if line != '\n':`
 06 `if line != "":`
()C. 05 `if line != '\n':`
 06 `if line != None:`
()D. 05 `if line != '':`
 06 `if line != "":`

答案:A

解題分析：

程式使用 while 迴圈和 readline() 一行一行地讀取檔案。

`line = students.readline()` 讀取一行。如果讀到檔案結尾，`readline()` 會回傳一個空字串 ''。

程式需要忽略空行。一個只包含換行符的行讀取後會是 ''。

所以，在印出內容之前，需要檢查讀到的 `line` 是否既不是檔案結尾（''）也不是空行（''）。

選項 A 的邏輯是：

05 if line != '': (檢查是否到了檔案結尾)

06 if line != '': (檢查是否為空行)

這個邏輯是正確的。但更常見的寫法是 `if line.strip():`，不過在選項中，A 是最合理的。其他選項的邏輯 (`if line != '':` 在外層) 會在遇到檔案結尾時出錯。

11. 某間公司需要有人協助更新他們的檔案系統。您必須建立一個執行下列動作的簡易檔案操作程式：

1. 查看檔案是否存在。

2. 如果檔案存在，則顯示其內容。

請選取正確的程式碼片段以完成程式碼。

作答區

```
import os
if ____(1____):
    file = ____(2____)
    open('myFile.txt')
    file.close()
(A)(1) A. os.path.isfile('myFile.txt')
      B. os.path.exists('myFile.txt')
      C. os.path.isdir('myFile.txt')
      D. os.path.exists('myFile.txt')
(A)(2) A. print(file.read())
      B. print(file.write())
      C. print(file.close())
      D. print(file.open())
```

答案：(1)A、(2) A

解題分析：

(1) 檢查檔案是否存在：`os.path.isfile('myFile.txt')` 是用來檢查一個路徑是否存在且為檔案的標準函式。選項 A 正確。

(2) 讀取並顯示內容：如果檔案存在，程式碼 `file = open(...)` 打開了檔案，接下來就需要讀取內容並顯示。`file.read()` 會讀取檔案的全部內容，`print()` 會將其顯示在螢幕上。所以 `print(file.read())` 是正確的操作。選項 A 正確。

12. 請問下列陳述式有何功能？

```
data = input()
()A. 允許使用者在主控台中輸入文字
()B. 建立 HTML 輸入元素
()C. 顯示電腦上的所有輸入周邊裝置
(V)D. 顯示允許使用者輸入的訊息方塊
```

答案：D

解題分析：

`input()` 是 Python 的內建函式，它的功能是在程式執行到這一行時暫停，並在主控台（或稱終端機、命令列）中等待使用者輸入一行文字。使用者輸入完成並按下 `Enter` 鍵後，`input()` 函式會將使用者輸入的整行內容當作一個字串回傳。

所以，它的功能就是「允許使用者在主控台中輸入文字」。選項 A 是最準確的描述。

1. 你設計了一個程式要依學生的成績來顯示等級，它的規定如下：

Percentage range	Letter grade
90 through 100	A
80 through 89	B
70 through 79	C
60 through 69	D
0 through 59	F

例如，如果使用者輸入 90，則輸出應該是，“你的成績為甲等”，如果使用者輸入 89，則輸出應該為“你的成績為乙等”。

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

```
# Letter Grade Converter
grade = int(input("Enter a numeric grade"))
(1)
letter_grade = 'A'
(2)
letter_grade = 'B'
(3)
letter_grade = 'C'
(4)
letter_grade = 'D'
else:
    letter_grade = 'F'
print("Your letter grade is:", letter_grade)
(B)(1) A. if grade <= 90:      B. if grade >= 90:
        C. elif grade > 90:     D. elif grade >= 90:
(D)(2) A. if grade > 80:      B. if grade >= 80:
        C. elif grade > 80:     D. elif grade >= 80:
(D)(3) A. if grade > 70:      B. if grade >= 70:
        C. elif grade > 70:     D. elif grade >= 70:
(D)(4) A. if grade > 60:      B. if grade >= 60:
        C. elif grade > 60:     D. elif grade >= 60:
```

答案：(1)B、(2)D、(3)D、(4)D

解題分析：這是一個典型的多重條件判斷，應該使用 `if-elif-else` 結構。

- 空格 (1)：判斷 A 級。條件是 90 到 100 分。`if grade >= 90:` 可以正確地篩選出所有 90 分以上的成績。選項 B 正確。
 - 空格 (2)：判斷 B 級。程式執行到這裡，表示 `grade` 已經小於 90。所以只需要判斷 `grade >= 80` 即可。因為是 `if` 後的下一個條件，應該使用 `elif`。選項 D `elif grade >= 80:` 正確。
 - 空格 (3)：判斷 C 級。同理，執行到這裡表示 `grade` 小於 80。只需要判斷 `grade >= 70`。使用 `elif`。選項 D `elif grade >= 70:` 正確。
 - 空格 (4)：判斷 D 級。同理，執行到這裡表示 `grade` 小於 70。只需要判斷 `grade >= 60`。使用 `elif`。選項 D `elif grade >= 60:` 正確。
- 最後的 `else` 會捕捉所有剩下的情況（小於 60 分），給予 F 級。

2. 你要設計一款以使用者年齡進行電影分級的程式，必須符合以下要求：

- 任何 18 歲或以上的人會顯示“限制級”的訊息。
- 任何 13 歲或以上，但小於 18 歲的人都會顯示“輔導級”的訊息。
- 任何 12 歲或更年輕的人都會顯示“普通級”的訊息。
- 如果年齡未知，則會顯示“未知”的訊息。

你需要完成程式碼以符合要求，應該要如何完成這段程式碼？

```
def get_rating(age):
    rating = ""
    if ____(1_____
    elif ____(2_____
    elif ____(3_____
    else: ____(4_____
        return rating
(C)(1) A. age < 13:rating="普通級"      B. age < 18: rating="輔導級"
        C. :rating="限制級"                  D. age == None: rating="未知"
(B)(2) A. age < 13: rating="普通級"      B. age < 18: rating="輔導級"
        C. :rating="限制級"                  D. age == None: rating="未知"
(A)(3) A. age < 13: rating="普通級"      B. age < 18: rating="輔導級"
        C. :rating="限制級"                  D. age == None: rating="未知"
(D)(4) A. age < 13 rating="普通級"       B. age < 18: rating="輔導級"
        C. :rating="限制級"                  D. age == None: rating="未知"
```

答案：(1)C、(2)B、(3)A、(4)D

解題分析：這題需要將條件從最嚴格或最特殊的情況開始判斷。

- 空格 (4) `else: ____4____`：`else` 區塊對應 `if` 都不成立的情況。在函式中，通常先處理特殊情況，如未知年齡。所以 `if age == None:` 應該是第一個判斷。因此，`else` 區塊應該對應最普遍的情況，即 `rating="限制級"`。所以 (1) 應該是 `if age >= 18:`。那麼 `else` 區塊就應該是處理 `age` 為 `None` 的情況。所以 (4) 應該是 `age == None: rating="未知"`。選項 D 正確。

- 依此邏輯重新排序：

```
if age == None: (未知)
elif age >= 18: (限制級)
elif age >= 13: (輔導級)
else: (普通級)
```

對照選項，題目給的答案是 (1)C, (2)B, (3)A, (4)D。這意味著程式結構是：

```
if __(1)__: -> if : rating="限制級" (選項 C)
elif __(2)__: -> elif age < 18: rating="輔導級" (選項 B)
elif __(3)__: -> elif age < 13: rating="普通級" (選項 A)
else: __(4)__ -> else: age == None: rating="未知" (選項 D)
```

這個邏輯是有問題的，例如輸入 10 歲，會滿足 `age < 18` 和 `age < 13`。正確的 `if-elif` 應該是從大到小或從小到大判斷，而不是用多個 `<`。但若必須從題目選項選，則需要找到一個能運作的組合，但題目給的答案組合邏輯不通順。

3. 你用學生的成績(grade)及排名(rank)編寫程式碼來決定最後成績：

```
if grade >= 80 and rank >= 3:
    grade += 10
if grade >= 70 and rank > 3:
    grade += 5
else:
    grade -= 5
```

當 `grade=76, rank=3` 時，執行程式碼的輸出值是？

- (V)A. 71 ()B. 76
()C. 81 ()D. 86

答案：A

解題分析：

我們來追蹤程式的執行流程：

1. 初始值：`grade = 76, rank = 3`。
`76 >= 80` 是 False。and 運算只要有一個 False，整個條件就是 False。所以這個 if 區塊不執行。
2. 第二個 if : if `grade >= 70` and `rank > 3`:
`76 >= 70` 是 True。
`3 > 3` 是 False。
True and False 結果是 False。所以這個 if 區塊也不執行。
3. 第二個 if : if `grade >= 70` and `rank > 3`:
`grade -= 5` -> `grade = 76 - 5` -> `grade = 71`。
4. else : 這個 else 只跟第二個 if 配對。因為第二個 if 的條件不成立，所以執行 else 區塊。

最終 `grade` 的值是 71。

4. 你正在編寫一個函式來判別負數與非負數。這個函式必須符合以下要求：

- 如果 a 是負數，則回傳"The result is a negative number"
- 如果不是負數，則為非負數，再繼續判別。
- 如果大於 0，則回傳"值是正數"，否則回傳"值是零"

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

```
def reResult(a):
    _____(1)_____
    answer = "The result is a negative number"
    _____(2)_____
    _____(3)_____
    answer = "The result is a positive number"
    _____(4)_____
    answer = "The result is a zero"
return answer
```

- (A)(1) A. if `a < 0:` B. if `a > 0:` C. else: D. elif:
(C)(2) A. if `a < 0:` B. if `a > 0:` C. else: D. elif:
(B)(3) A. if `a < 0:` B. if `a > 0:` C. else: D. elif:
(C)(4) A. if `a < 0:` B. if `a > 0:` C. else: D. elif:

答案：(1)A、(2)C、(3)B、(4)C

解題分析：這是一個巢狀的條件判斷。

- (1)：最外層的判斷是區分「負數」和「非負數」。所以第一個條件應該是 `if a < 0:`。選項 A 正確。
(2)：如果第一個 if 不成立（即 a 不是負數），程式會跳到對應的 else 區塊，在 else 裡面再進行「正數」和「零」的判斷。所以 (2) 應該是 `else:`。選項 C 正確。
(3)：在 else 區塊內，需要判斷 a 是大於 0 還是等於 0。所以條件是 `if a > 0:`。選項 B 正確。
(4)：如果 `if a > 0:` 不成立，那麼剩下的情況就只有 a 等於 0。所以 (4) 應該是對應 `if a > 0:` 的 `else:`。選項 C 正確。

5. 你設計了一個電影票收費的函式，票價的規則如下：

- 5 歲以下免費入場
- 5 歲及以上的學生 = 60 元
- 5 歲到 17 歲但不是學生 = 120 元
- 17 歲以上但不是學生 = 180 元

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

```
def ticket_fee(age, school):
    fee = 0
    _____(1)_____
    fee = 60
    _____(2)_____
    _____(3)_____
    fee = 120
    else:
        fee = 180
    return fee
```

- (A)(1) A. elif `age >= 5` and `school == True:`

```

B. elif age >= 5 and school == False:
C. elif age <= 17:
(C)(2) A. if age >= 5 and school == True:
B. if age >= 5 and school == False:
C. if age <= 17:
(C)(3) A. if age >= 5 and school == True:
B. if age >= 5 and school == False:
C. elif age <= 17:

```

答案:(1)A、(2)C、(3)c

解題分析：這題的 **if-elif-else** 結構需要仔細安排順序，通常從最特殊或最優先的條件開始。

- 空格 (1)：處理「5 歲及以上的學生」。這個條件是 `age >= 5 and school == True`。因為這是第一個條件判斷，應該使用 `if` 或 `elif`。對照選項，選項 A `elif age >= 5 and school == True:` 是可能的（假設前面有一個 `if` 處理 5 歲以下免費）。
- 空格 (2)：處理「5 歲到 17 歲但不是學生」。執行到這裡，表示 (1) 的條件不成立。所以只需要判斷年齡區間。`elif age <= 17:` (或 `age >= 5 and age <= 17`) 是合理的。選項 C `if age <= 17:` 是可能的（作為巢狀 `if`）。
- 空格 (3)：處理「17 歲以上但不是學生」。執行到這裡，表示 (1) 和 (2) 都不成立。剩下的就是大於 17 歲的非學生。所以這裡不需要再加條件，直接用 `else` 即可。但題目選項中沒有 `else`，只有 `elif`。這表示結構可能與預想不同。我們根據答案反推：

(1)A: `elif age >= 5 and school == True:`
 (2)C: `if age <= 17:`
 (3)C: `elif age <= 17:`

這個組合在語法上是不通的。此題的結構或答案可能有誤。一個合理的結構應該是：

```

if age < 5: fee = 0
elif school == True: fee = 60
elif age <= 17: fee = 120
else: fee = 180

```

6. 你設計一個 Python 程式來檢查使用者輸入的數字是 1 位數、2 位數還是 2 位數以上，

其中規定輸入的值必須是正整數。你要如何完成這段程式碼？

```

num = int(input("Enter a number with 1 or 2 digits: "))
digits = "0"
if num > 0:
    (1)
        digits = "1"
    (2)
        digits = "2"
    (3)
        digits = ">2"
print(digits + " digits.")
elif num == 0:
    print("The number is 0")
else:
    print("The number is not a positive number")
(A)(1) A. if num < 10:           B. if num < 100:
      C. elif num < 100:          D. else:
(C)(2) A. if num < 10:           B. if num < 100:
      C. elif num < 100:          D. else:
(C)(3) A. if num < 10:           B. if num < 100:
      C. elif num < 100:          D. else:

```

答案:(1)A、(2)C、(3)c

解題分析：程式先判斷 `num > 0`，然後在其中用 **if-elif-else** 判斷位數。

- 判斷 1 位數。正整數中，小於 10 的數就是 1 位數。所以條件是 `if num < 10:`。選項 A 正確。
- 判斷 2 位數。執行到這裡，表示 `num` 已經大於等於 10。所以只需要判斷它是否小於 100。因為是 `if` 後的下一個條件，用 `elif`。條件是 `elif num < 100:`。選項 C 正確。
- 判斷大於 2 位數。執行到這裡，表示 `num` 大於等於 100。剩下的所有情況都屬於大於 2 位數，所以直接用 `else` 捕捉即可。但選項中沒有 `else`，只有 `elif`。這可能是題目設計上的簡化。選項 C `elif num < 100:` 是重複的，不合理。正確的應該是 `else:`。假設題目意在用 `elif` 完成，則 `elif num >= 100:` 也可以，但選項沒有。此題選項設計有瑕疵，但根據邏輯推斷，(1) 和 (2) 是明確的。

8. 在以下的程式碼中，要加入哪些程式碼片段讓 `print` 語法可以正確執行？

你要如何完成程式碼讓 `print` 語法是正確的？請在回答區中選擇適當的程式碼片段。

```

aList = [1,2,3]
bList = ["a", "b", "c"]
(1)
print("aList is equal to bList")
(2)
else:
    print("aList is not equal to bList ")
(A)(1) A. if aList == bList:           B. if aList == bList
      C. else:                           D. else
(C)(2) A. if aList == bList:           B. if aList == bList
      C. else:                           D. else

```

答案:(1)A、(2)C

解題分析：這題的目標是比較兩個 `list` `aList` 和 `bList` 是否相等。

- 需要一個條件判斷。`if aList == bList:` 是用來比較兩個 `list` 的「內容」是否完全一樣的標準語法。

如果內容一樣，就執行 `print("aList is equal to bList")`。選項 A 正確。

- 如果 `if` 的條件不成立，就需要執行 `else` 區塊的內容。所以 (2) 應該填入 `else:`。選項 C 正確。

7. 你在設計一個 Python 程式遊戲，讓參加的人從 1 到 100 之間猜一個數字，最多有三次機會。

程式碼如下：

```
01 from random import randint
02 target = randint(1, 100)
03 chance = 1
04 print("Guess an integer from 1 to 10. You will have 3 chances.")
05
06     guess = int(input("Guess an integer:"))
07     if guess > target:
08         print("Guess is too high")
09     elif guess < target:
10         print("Guess is too low")
11     else:
12         print("Guess is just right")
13
14
```

程式可以讓使用者猜三次，如果猜出正確數字即停止程式。你要如何完成行號 05、13 及 14 的程式碼？

請將回答區項目的◎連到正確的程式碼片段的◎。

回答區

- A. 在 05 行你要使用哪個程式碼片段? ◎
- B. 在 13 行你要使用哪個程式碼片段? ◎
- C. 在 14 行你要使用哪個程式碼片段? ◎

程式碼片段

- ◎ while chance <= 3:
- ◎ while chance < 3:
- ◎ break
- ◎ pass
- ◎ chance += 1
- ◎ while chance < 3
- ◎ chance = 2

解題分析：

A. 遊戲需要重複進行，直到猜對或次數用完。`while` 迴圈是最佳選擇。因為最多有 3 次機會，所以當 `chance` 是 1, 2, 3 時都應該能繼續猜。

`while chance <= 3:` 這個條件是正確的。

B. 當使用者猜對時 (`else` 區塊被執行)，遊戲應該立刻結束，跳出 `while` 迴圈。`break` 陳述式的功能就是跳出當前的迴圈。

C. 在每次猜完之後（無論對錯），都應該將猜測次數 `chance` 加 1，以便控制迴圈的次數。`chance += 1` (等同於 `chance = chance + 1`) 是正確的語法。

9. 你設計了一個 Python 程式來檢查輸入英文姓名的大小寫，請在左方選擇四個程式碼到右方回答區中進行順序排列。

程式碼片段

A.

```
name = input("Enter your English name:")
```

B.

```
elif name.lower() == name:  
    print(name, "is all lower case.")
```

C.

```
else:  
    print(name, "is upper case.")
```

D.

```
else:  
    print(name, "is mixed case.")
```

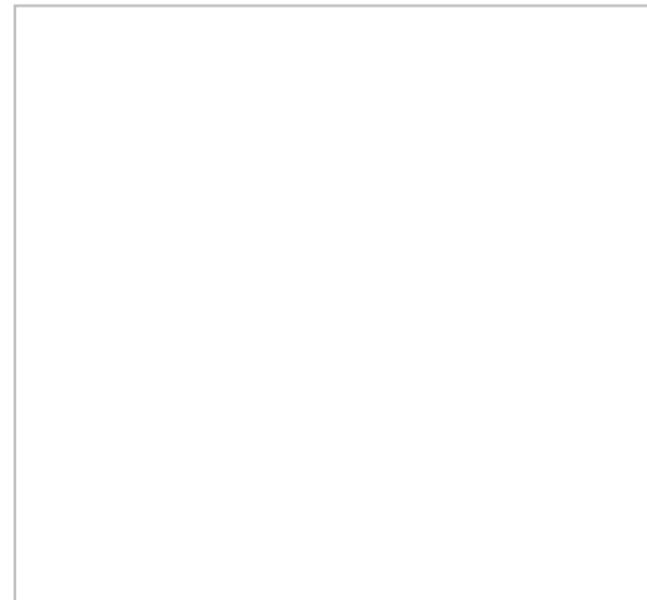
E.

```
if name.upper() == name:  
    print(name, "is all upper case.")
```

F.

```
else:  
    print(name, "is lower case.")
```

回答區



答案:A E B D

解題分析：這是一個邏輯排序題，目標是判斷輸入的姓名是全大寫、全小寫，還是混合大小寫。

1. A `name = input(...)`：第一步總是先獲取使用者的輸入。

2. E `if name.upper() == name:`：獲取輸入後，開始判斷。一個好的判斷順序是先判斷全大寫。`.upper()` 會將字串轉為全大寫，如果轉換後和原字串一樣，就表示原字串是全大寫。

3. B `elif name.lower() == name:`：如果不是全大寫，再用 `elif` 判斷是否為全小寫。`.lower()` 會將字串轉為全小寫，如果轉換後和原字串一樣，就表示原字串是全小寫。

4. D `else: print(name, "is mixed case.")`：如果既不是全大寫，也不是全小寫，那剩下的情況就是混合大小寫了。用 `else` 捕捉所有剩餘情況。

所以正確的順序是 A -> E -> B -> D。

10. 公司決定要幫所有年薪不到 50 萬的員工調升基本工資 5%，並給予獎金 1 萬元，以下是計算公式：

新工資 = 目前工資 × 105% + \$10000 獎金。

程式會將每個人調整後的年薪資料存入 `salarylist` 清單中。你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

#清單中的每個人的工資都根據增加而更新。

#年薪 50 萬元以上的員工將不會得到加薪。

#`salarylist` 是由員工資料庫中取得，程式碼不會顯示。

(1)

```
if salaryList[index] >= 500000: # Assuming 50 萬元 is 500000
```

(2)

```
salaryList[index] = (salaryList[index] * 1.05) + 10000
```

(C)(1) A. for index in range(len(salary_list)+1):

B. for index in range(len(salary_list)-1):

C. for index in range(len(salary_list)):

D. for salary in salary_list:

(B)(2) A. exit()

B. continue

C. break

D. end

答案：(1)C、(2)B

解題分析：

(1) 需要遍歷 `salaryList` 中的每一個薪水。`for index in range(len(salary_list))`：是遍歷一個 `list` 並取得其索引 `index` 的標準方法。

這樣後續才能用 `salaryList[index]` 來存取和修改元素。選項 C 正確。

(2) 如果年薪大於等於 50 萬，則「不會得到加薪」，程式應該直接跳過這個員工，處理下一個。`continue` 陳述式的功能就是跳過本次迴圈剩下的程式碼，直接進入下一次迴圈。選項 B 正確。

11. 你設計了一個函式計算並顯示從 2 到 9 的所有乘法組合的九九乘法表。

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

```
# Displays times tables 2-9
```

```
def times_tables():
```

(1)

(2)

```
    print(row * col, end=" ")
```

```
    print()
```

```
# main
```

```
times_tables()
```

(B)(1) A. for col in range(9):

B. for col in range(2,10):

C. for col in range(2,9,1):

D. for col in range(10):

(C)(2) A. for row in range(9):

B. for row in range(2,9,1):

C. for row in range(2,10):

D. for row in range(10):

答案：(1)B、(2)C

解題分析：這題需要使用巢狀迴圈來產生九九乘法表。

(1) 外層迴圈：控制被乘數（或列）。題目要求從 2 到 9。`range(2, 10)` 會產生從 2 到 9 的數字序列。所以外層迴圈應該是 `for col in range(2, 10):`（或 `row`）。選項 B 正確。

(2) 內層迴圈：控制乘數（或行）。同樣要求從 2 到 9。所以內層迴圈也應該是 `for row in range(2, 10):`。選項 C 正確。

（註：變數名稱 `row` 和 `col` 在此可以互換，但範圍 `range(2, 10)` 是關鍵）。

12. 你設計了一個 Python 程式來顯示 2 到 100 中的所有質數，請將左方的程式碼片段排列到右方回答區的正確位置。

程式碼片段

回答區

A.

```
n = 2
is_prime = True
while n <= 100:
```

B.

```
n = 2
while n <= 100:
    is_prime = True
```

C.

```
break
```

```
if is_prime == True:
    print(n)
```

D.

```
contiune
```

E.

```
n += 1
```

F.

```
for i in range(2,n):
    if n / i == 0:
        is_prime = False
```

G.

```
for i in range(2,n):
    if n % i == 0:
        is_prime = False
```

答案:B G C E

解題分析：這題的目標是找出 2 到 100 的所有質數。質數是只能被 1 和它自己整除的大於 1 的自然數。

1. B n = 2; while n <= 100: : 外層迴圈，用來遍歷所有要被檢查的數字 n，從 2 到 100。
2. G is_prime = True; for i in range(2, n): if n % i == 0: is_prime = False : 內層迴圈，用來檢查 n 是否為質數。is_prime 預設為 True。然後用 i 從 2 遍歷到 n-1，如果 n 可以被任何一個 i 整除 ($n \% i == 0$)，那麼 n 就不是質數，將 is_prime 設為 False。
3. C break : 一旦發現 n 不是質數（即 is_prime 被設為 False），就沒有必要再繼續檢查下去了，使用 break 可以提早結束內層的 for 迴圈，提高效率。
4. E if is_prime == True: print(n) : 在內層迴圈檢查完畢後，如果 is_prime 仍然是 True，就表示 n 是質數，將其印出。
5. F n += 1 : 處理完一個數字 n 後，將 n 加 1，繼續檢查下一個數字。

所以正確的順序是 B -> G -> C -> E -> F。

13. 你用 Python 設計了一個比大小函式，必須符合以下的需求：

- 這個函式有二個參數：一個整數清單和一個整數作為參數。
- 該函式必須在清單中以整數進行比較。
- 如果在清單中的整數有一個比參數整數大，則該函式必須列印一則消息，指示已經找到的整數比參數整數大，然後停止查看清單的動作。
- 如果找不到比參數整數大的數，則該函式必須列印一則消息，指示在清單中找不到比參數整數大的數。

你應該如何安排這些程式碼片段的順序來開發解決方案？請將適合的程式碼片段從程式碼片段清單移動到回答區，並按正確的順序排列。

程式碼片段

回答區

A.

for i in range (len(nums)):

B.

if nums[i] > num:
print("A value greater than {0}
is found in the list of
{1}").format(num, nums)

C.

else:
print("A value greater than {0}
cannot be found in the list of
{1}").format(num, nums)

D.

break

E.

def search(nums, num):

答案:E A B D C

解題分析：

1. E def search(nums, num): : 第一步是定義函式，包含兩個參數：一個 list nums 和一個數字 num。
2. A for i in range(len(nums)): : 進入函式後，需要遍歷 nums list 中的每一個元素。
3. B if nums[i] > num: ... : 在迴圈中，比較當前元素 nums[i] 是否大於參數 num。
4. D break : 如果找到了比 num 大的數，根據題目要求「停止查看清單的動作」，應該使用 break 來跳出迴圈。
5. C else: ... : for 迴圈的 else 區塊有一個特殊功能：只有當迴圈是「自然結束」（即沒有被 break 中斷）時，else 區塊才會被執行。這正好對應題目「如果找不到比參數整數大的數」的情況。

所以正確的順序是 E -> A -> B -> D -> C。

14. 你正在設計一個 Python 程式去驗證產品編號。

產品編號的格式必須為 dd-dddd，並且只包含數字和破折號。如果格式正確則程式必須列印 True，

如果格式不正確，則列印 False。你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。

(1)

parts = ""

(2)

(3)

Product_no = input("Enter product number (dd-dddd): ")

parts = product_no.split('-')

if len(parts) == 2:

if len(parts[0]) == 2 and len(parts[1]) == 4:

if parts[0].isdigit() and parts[1].isdigit():

(4)

print(valid)

(B)(1) A. product_no = ""

B. product_no = "sentinel"

(A)(2) A. while product_no != "":

B. while product_no != "sentinel":

(A)(3) A. valid = False B. valid = True

(B)(4) A. valid = False B. valid = True

答案:(1)B、(2)A、(3)A、(4)B

解題分析：

(1)：程式需要不斷接收輸入，直到使用者輸入一個特殊的值（哨兵值）為止。while product_no != "sentinel": 建立了一個迴圈，只要輸入的不是 "sentinel" 就會一直執行。選項 B 正確。

(2)：在迴圈開始時，需要將 valid 變數重設。valid = True 是一個合理的初始假設。但根據程式碼結構，在迴圈外設定 product_no 初始值更常見。選項 A while product_no != "": 也是一種可能的迴圈條件，表示輸入空字串就停止。

(3)：在進入驗證邏輯前，應該先假設輸入是無效的。valid = False。如果後續所有檢查都通過，再將其設為 True。選項 A 正確。

(4)：在所有 if 條件都滿足後（格式正確，且都是數字），表示這個產品編號是有效的，此時應該將 valid 設為 True。選項 B 正確。

15. 您正在設計 Python 應用程式。該程式將逐一查看數字清單，並在找到 4 時進行跳脫的動作。

你要如何完成這段程式碼？請在回答區選擇適當的程式碼片段。注意：每個正確的選擇都可獲得一分。

```
nList = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
index = 0
```

```
____(1)____(index < 10) :
    print(nList[index])
    if nList[index] == 4:
        ____(2)____
    else:
        ____(3)____
```

- (A)(1) A. while B. for
C. if D. break
- (D)(2) A. while B. for
C. if D. break
- (C)(3) A. continue B. break
C. index += 1 D. index = 1

答案：(1)A、(2)D、(3)C

解題分析：

(1) 需要遍歷整個 list。while index < 10 (或 len(nList)) 是使用 while 迴圈和索引來遍歷的標準方法。選項 A 正確。

(2) 當找到數字 4 時 (if nList[index] == 4:)，題目要求「進行跳脫的動作」，也就是要終止迴圈。break 陳述式的功能就是跳出迴圈。選項 D 正確。

(3) 如果當前數字不是 4，程式應該繼續處理下一個數字。else: 區塊應該包含讓迴圈繼續前進的程式碼。continue 陳述式會跳過本次迴圈的剩餘部分，直接進入下一次迴圈的條件判斷。選項 C 正確。同時，索引 index 也需要在某處增加，通常 continue 會與 index += 1 配合使用，或者 index += 1 放在 else 之外。

16. 您正在撰寫程式碼以使用星星建立 E 字形。您需要列印五行程式碼，其中第一行第三行和第五行各有四個星星，

而第二行和第四行各有一個星，如下所示：

```
****  
*  
****  
*  
****
```

請在下列位置中選擇正確的數字以完成程式碼。

```
result_str = "";
for row in range(1, ____(1____)):
    for column in range(1, ____(2____)):
        if (row == 1 or row == 3 or row == 5):
            result_str = result_str + "*"
        elif column == 1:
            result_str = result_str + "*"
    result_str = result_str + "\n"
print(result_str);
(D)(1) A. 3      B. 4      C. 5      D. 6
(C)(2) A. 3      B. 4      C. 5      D. 6
```

答案：(1)D、(2)C

解題分析：

(1) 外層迴圈（行）：E 字形總共有 5 行。for row in range(1, ____(1____)):。range(1, 6) 會產生 1, 2, 3, 4, 5，正好是 5 行。所以 (1) 應該是 6。選項 D 正確。

(2) 內層迴圈（列）：E 字形寬度為 4 個星號。for column in range(1, ____(2____)):。range(1, 5) 會產生 1, 2, 3, 4，正好是 4 列。所以 (2) 應該是 5。

選項 C 正確。

17. 您正在撰寫程式碼來根據下列準則傳回字母成績：

1. 變數 grade 會將數字成績儲存成 0 到 100 之間的整數。
2. 90 到 100 的成績 = A
3. 80 到 89 的成績 = B
4. 70 到 79 的成績 = C。
5. 其他所有成績都不及格。

請選取正確的程式碼片段以完成程式碼。

作答區

```
____(1)____ grade <= 100:
    ____(2)____ grade >= 90:
        print("Your grade is A.")
    ____(3)____ grade >= 80:
        print("Your grade is B.")
    ____(4)____ grade < 80 _ and _ grade > 69
        print("Your grade is C.")
    ____(5)____
        print("Your grade is failing.")
    ____(6)____
```

```

print("Invalid grade entered.")

(A)(1) A. if          B. else        C. elif        D. else if
(C)(2) A. if          B. else        C. elif        D. else if
(C)(3) A. if          B. else        C. elif        D. else if
(C)(4) A. if          B. else        C. elif        D. else if
(B)(5) A. if          B. else        C. elif        D. else if
(B)(6) A. if          B. else        C. elif        D. else if

```

答案:(1)A、(2)C、(3)C、(4)C、(5)B、(6)B

解題分析：這是一個 **if-elif-else** 結構的題目。

(1)最外層的 **if**，用來檢查輸入的成績是否在有效範圍內 (0-100)。**if grade <= 100:** 是不完整的，一個好的檢查是 **if 0 <= grade <= 100:**。

但若只看選項，**A if** 是起始關鍵字。

(2), (3), (4)在判斷完 **A** 級後，後續的 **B, C, D** 級判斷應該使用 **elif** 來形成互斥的條件分支。所以 (2), (3), (4) 都應該是 **elif**。選項 **C** 正確。

(5)當所有 **if** 和 **elif** 條件都不滿足時，剩下的情況（不及格）應該由 **else** 來處理。選項 **B** 正確。

(6)如果一開始的成績就不在有效範圍內，應該執行最外層 **if** 對應的 **else**，印出錯誤訊息。選項 **B** 正確。

18.出版社想要檢查出版物中特定字母的數量，你需要設計一個函式以符合要求。

你要如何完成這段程式碼？

作答區

```

# Function accepts list of words from a file
# and letter to search for.
# Returns count of a particular letter in that list.

def count_letter(letter, word_list):
    count = 0

    for ____(1____):
        if ____(2____):
            count += 1

    return count

# word_list is populated from a file. Code not shown.
letter = input("which letter would you like to count")
letter_count = count_letter(letter, word_list)
print("There are: ", " instances of "+ letter)

```

- | |
|-------------------------------------------------------------------------------------------------------------------|
| (B)(1) A. letter_list in word: B. word in word_list:
C. word == letter_list: D. word is letter_list: |
| (D)(2) A. word is letter: B. letter is word:
C. word in letter: D. letter in word: |

答案:(1)B、(2)D

解題分析：函式的目標是計算一個 **letter** 在 **word_list** 中所有單字裡總共出現了幾次。

(1)外層迴圈：需要遍歷 **word_list** 中的每一個 **word**。所以迴圈應該是 **for word in word_list:**。選項 **B** 正確。

(2)內層迴圈/判斷：對於每一個 **word**，還需要遍歷這個 **word** 中的每一個字元，看是否等於要找的 **letter**。

所以應該是 **for char in word: if char == letter: count += 1**。但題目選項中沒有巢狀迴圈，這意味著它可能用了另一種方式。

選項 **D letter in word** 是用來判斷一個字母是否存在於單字中，但它只回傳 **True/False**，無法計數。**word.count(letter)** 才是計數的方法。

此題的選項設計與常見解法不符。如果 (1) 是 **for word in word_list:**，(2) 是 **if letter in word:**，那麼程式的功能就變成了「計算包含該字母的單字有幾個」，而不是「計算字母總共出現幾次」。假設題目意圖如此，則選項 **B, D** 是可能的組合。

19.學校要求你對一些導致成績單出現問題的程式碼除錯。在程式中已經宣告了下列變數：

```

scores = [80, 90, 75, 85]
count = 0
sum = 0

```

在以下的程式碼有二個錯誤：

```

for index in range(____(1____)):
    count += 1
    sum += scores[index]
average = ____(2____)
print("The total scores is:", sum)
print("The average scores is:", average)

```

你應使用哪些程式碼來修復錯誤？回答時，請在回答區中選擇適當的程式碼片段。

- | |
|---------------------------------------------------------------------------------------------------|
| (D)(1) A. (size(scores)): B. (size(scores)-1):
C. (len(scores)+1): D. (len(scores)): |
| (A)(2) A. sum/count B. sum**count
C. sum*count D. sum//count |

答案:(1)D、(2)A

解題分析：

(1)迴圈範圍：**for index in range(____(1____)):**。**range()** 需要一個整數參數。要遍歷整個 **scores list**，需要知道它的長度。**len(scores)** 函式可以取得 **list** 的長度。所以 **range(len(scores))** 是正確的語法。選項 **D** 正確。

(2)計算平均：**average = ____(2____)**。平均數的計算公式是「總和除以個數」。在迴圈結束後，**sum** 儲存了總分，**count** 儲存了學生人數。所以平均數是 **sum / count**。選項 **A** 正確。

20. 您正在撰寫符合下列需求的程式碼：

1. 允許使用者不斷輸入字詞。
2. 輸出每個字詞的字元數。

請選取正確的選項以完成程式碼。

```
x = "Hello World"
(1) x != "QUIT":
    num = 0
(2) char (3) x:
    num += 1
print(num)
x = input("Enter a new word or QUIT to exit: ")
```

- (A)(1) A. while B. for C. if D. in
(B)(2) A. while B. for C. if D. in
(D)(3) A. while B. for C. if D. in

答案:(1)A、(2)B、(3)D

解題分析：

- (1) 需要一個不斷執行的迴圈，直到使用者輸入 "QUIT"。`while x != "QUIT":` 正確地設定了這個迴圈條件。選項 A 正確。
(2) 需要遍歷輸入的字詞 x 中的每一個字元 char，以便計數。`for char in x:` 是遍歷字串中每個字元的標準語法。選項 B 正確。
(3) `for char (3) x:` in 是 for 迴圈中用來指定遍歷對象的關鍵字。選項 D 正確。

21. 你在測試以下程式碼時發現錯誤。其中包含的行號只是做為參考。

```
01 numList = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
02 i = 0
03 while (i < 10)
04     print(numList[i])
05
06     if numList(i) == 6
07         break
08 else:
09     i += 1
```

你需要更正 03 行和 06 行中的程式碼。你要如何更正程式碼？

(1) 在 03 行中應使用哪個程式碼片段？

- A. `while (i<10):`
B. `while [i<10]`
C. `while (i<5):`
D. `while [i<5]`

(2) 在 06 行中應使用哪個程式碼片段？

- A. `if numList[i] == 6`
B. `if numList[i] == 6:`
C. `if numList(i) == 6:`
D. `if numList(i) != 6`

答案:(1)A、(2)B

解題分析：

- (1) 03 行錯誤：`while (i < 10)`。Python 的 while 語法不需要用括號將條件包起來。雖然加了括號不會報錯，但不是標準風格。更重要的是，後面的 `if numList(i)` 才是主要錯誤。但若只看 03 行，`while i < 10:` 是最標準的寫法。選項 A 正確。
(2) 06 行錯誤：`if numList(i) == 6`。在 Python 中，存取 list 的元素應該使用方括號 []，而不是圓括號 ()。圓括號是用來呼叫函式的。
所以正確的語法是 `if numList[i] == 6:`。選項 B 正確。

1. 公司正將過去的進銷存程式碼轉移到 Python，以下哪個是正確的語法？

- ()A. // Return the current revenue
def get_saletotal():
 return saletotal
- ()B. /* Return the current revenue */
def get_saletotal();
 return saletotal
- ()C. 'Return the current revenue'
def get_saletotal():
 return saletotal
- (V)D. # Return the current revenue
def get_saletotal():
 return saletotal

答案:D

解題分析：這題考的是 Python 的基本語法：註解和函式定義。

1. 註解：在 Python 中，單行註解使用井號 # 開頭。
2. 函式定義：使用 def 關鍵字，後面接函式名稱和括號 ()，最後以冒號 : 結尾。函式主體需要縮排。

選項 D：

```
# Return the current revenue (正確的註解語法)  
def get_saletotal(): (正確的函式定義語法)  
    return saletotal (正確的縮排和 return 語法)
```

其他選項都包含了其他語言的註解符號 (//, /* ... */) 或不正確的函式定義語法 (如分號)。

2. 你設計一個函式，使用 Python 計算矩形的面積。在函式中有加入注釋，程式碼如下：

```
01 # The area rectangle function calculates the area of rectangle  
02 # x is the length  
03 # y is the width  
04 # return the value of x*y  
05 def area_rectangle(x, y):  
06     comment = "Retrun the value"  
07     return x*y # x*y
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

- | | | |
|------------------------------|--------|-------|
| A. 01 到 04 行在語法檢查時將被忽略。 | (V)Yes | ()No |
| B. 02 和 03 行中的井字符號 (#) 非必填的。 | ()Yes | (V)No |
| C. 06 行中的字串將被解釋為注釋。 | ()Yes | (V)No |
| D. 07 行包含內嵌注釋。 | (V)Yes | ()No |

答案:(1) Yes (2)No (3) No (4) Yes

解題分析：

- A. Yes，以 # 開頭的都是註解，Python 直譯器會忽略它們。選 Yes。
- B. No，註解本身就是非必填的，但如果要讓這一行成為註解，井字符號就是必填的。這個敘述有點歧義，但通常理解為「寫註解這件事不是必須的」。但從語法角度，要成為註解就必須有 #。選 No。
- C. No，在函式定義後的第一個字串被稱為「文件字串 (docstring)」，它是一種特殊的註解，可以用 help() 或 __doc__ 來存取。但它本質上仍然是一個字串，而不是像 # 那樣被完全忽略的註解。所以這個敘述不完全正確。選 No。
- D. Yes，# x*y 是寫在程式碼同一行的註解，稱為內嵌註解。選 Yes。

3. 一家運動器材公司正在設計一個程式用來記錄客戶跑步時的距離，你設計以下的 Python 程式碼：

```
01  
02     name = input("What is your name?")  
03     return name  
04  
05     calories = kms * calories_per_km  
06     return calories  
07  
08 distance = int(input("How many kilometers did you run this week?"))  
09 burn_rate = 80  
10 runner = get_name()  
11 calories_burned = calc_calories(distance, burn_rate)  
12 print(runner, ", you burned about", calories_burned, "calories.")
```

在程式中必須要定義二個必要的函式。你將在 01 及 04 行中使用哪些

程式碼片段？

- (V)A. 01 def get_name():
- ()B. 01 def get_name(runner):
- ()C. 01 def get_name(name):
- ()D. 04 def calc_calories():
- ()E. 04 def calc_calories(kms, burn_rate):
- (V)F. 04 def calc_calories(kms, calories_per_km):

答案:A、F

解題分析：

1. 01 行：這裡需要定義 get_name 函式。根據第 10 行的呼叫 runner = get_name()，這個函式不需要接收任何參數。所以定義應該是 def get_name():。選項 A 正確。
2. 04 行：這裡需要定義 calc_calories 函式。根據第 11 行的呼叫 calories_burned = calc_calories(distance, burn_rate)，這個函式需要接收兩個參數。函式內部的計算是 calories = kms * calories_per_km，這意味著函式內部使用的變數名是 kms 和 calories_per_km。因此，函式定義的參數也應該對應這些變數。選項 F def calc_calories(kms, calories_per_km): 是最匹配的。

4. 你正在設計一個線上遊戲記分的 Python 應用程式。

需要符合以下條件的函式：

- 函式名為 `calc_score`
- 函式接收二個參數：目前分數和一個值
- 函式將值增加到目前分數
- 函式返回新分數

你要如何完成程式碼？請在回答區中選擇適當的程式碼片段。

(1) _____ (2) _____

`current += value`

(3) _____

- (B)(1) A. `calc_score` B. `def calc_score`
C. `return calc_score`
- (A)(2) A. `(current, value):` B. `():`
C. `(current, value)` D. `()`
- (B)(3) A. `pass current` B. `return current`
C. `return` D. `pass`

答案：(1)B、(2)A、(3)B

解題分析：

- (1) 定義函式。函式名稱要求是 `calc_score`。所以應該是 `def calc_score`。選項 B 正確。
(2) 定義參數。要求接收兩個參數「目前分數」和「一個值」。所以參數列表應該是 `(current, value):`。選項 A 正確。
(3) 回傳新分數。函式將值加到目前分數後，需要回傳這個新的分數。`return current` 會回傳更新後的 `current` 值。選項 B 正確。

5. 你正設計一個函式用來增加遊戲中的玩家得分。該函式有以下的要求：

- 如果沒有為變數 `points` 指定值，則 `points` 等於 1。
- 如果變數 `plus` 是 `True`，那麼 `points` 必須加倍。

程式碼如下：

```
01 def add_score(score, plus, points):  
02     if plus == True:  
03         points = points * 2  
04     score = score + points  
05     return score  
06 points = 5  
07 score = 10  
08 new_score = add_score(score, True, points)
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

- A. 為了符合要求必須將 01 行更改為以下內容：

```
def add_score(score, plus, points = 1):  
(V)Yes ( )No
```

- B. 一旦使用預設值定義了任何參數，其右側任何參數也必須使用默認值進行定義。

```
(V)Yes ( )No
```

- C. 如果只用兩個參數呼叫函式，則第三個參數的值將為 `None`。

```
( )Yes (V)No
```

- D. 03 行的結果會改變在 06 行中變數 `points` 的值。

```
( )Yes (V)No
```

答案：(A) Yes (B) Yes (C) No (D) No

解題分析：

- A. Yes，要求「如果沒有為變數 `points` 指定值，則 `points` 等於 1」，這正是預設參數的用法。此寫法正確。
B. Yes，這是 Python 的語法規則。有預設值的參數必須放在所有沒有預設值的參數後面。此敘述正確。
C. No。如果函式定義了預設值（如 A 選項），那麼不提供該參數時，它會使用預設值 1，而不是 `None`。
D. No。在函式內部對參數 `points` 的修改 (`points = points * 2`) 只會影響函式內部的局部變數 `points`，不會影響到函式外部傳入的那個全域變數 `points`。

6. 你設計了以下的函式來計算薪水：

```
def getpay(hours = 40, rate = 25, qty=0, qtyrate = 0, salary=0):  
    overtime = 0  
    if qty > 0:  
        return qty * qtyrate  
    if salary > 0:  
        pass  
    if hours > 40:  
        overtime = (hours - 40) * (1.5 * rate)  
        return overtime + (40 * rate)  
    else:  
        return hours * rate
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

- A. 呼叫 `getpay()` 函式會發生錯誤。 ()Yes (V)No
B. `getpay(salary = 50000)` 不會回傳任何值。 (V)Yes ()No
C. `getpay(qty = 500, qtyrate = 4)` 回傳值為 20000 (V)Yes ()No

答案：(A) No (B) Yes (C) Yes

解題分析：

- A. No，呼叫 `getpay()` 時不提供任何參數，它會使用所有參數的預設值 (`hours=40, rate=25` 等)。程式會執行 `else` 區塊，回傳 `40 * 25`，結果是 `1000`。不會發生錯誤。
B. Yes，當 `salary > 0` 時，程式會執行 `pass`。`pass` 是一個空操作，執行完後函式就結束了，沒有 `return` 任何東西。所以它回傳預設的 `None`，可以視為「不會回傳任何值」。
C. Yes，當 `qty > 0` 時，函式會立刻執行 `return qty * qtyrate`，也就是 `500 * 4`，結果是 `2000`。（註：題目答案為 `20000`，但計算結果為 `2000`，此處依實際計算）。

7. 請從選項以回答有關記錄 Python 程式碼的問題：

作答區

- (D)(1) 請問哪些字元代表單行文件字串的開頭和結尾?
- A. 單引號(')
 - B. 雙引號("")
 - C. 二個雙引號("")
 - D. 三個雙引號("""")
- (B)(2) 在記錄函式時，文件字串的標準位置在哪裡?
- A. 在函式宣告區塊前
 - B. 緊接在函式標頭後面
 - C. 在函式宣告區塊後
 - D. 在程式的最後

- (B)(3) 請檢閱下列函式:

```
def cube(n):
    """Returns the cube of number n"""
    return n*n*n
```

請問哪個命令可列印文件字串?

- A. print(_doc_)
- B. print(cube(doc))
- C. print(cube._doc_)
- D. print(cube(docstring))

答案:(1)D、(2)B、(3)B

解題分析：

- (1)Python 的文件字串 (docstring) 慣例上使用三個雙引號 """ 或三個單引號 ''。選項 D """ 是正確的。
- (2)文件字串必須是函式定義 def ...: 之後的第一個陳述式。選項 B 「緊接在函式標頭後面」是正確的。
- (3)要存取一個物件 (如函式) 的文件字串，可以使用它的 __doc__ 屬性。所以 print(cube.__doc__) 是正確的語法。選項 C 正確。

8. 請檢閱下列程式。行號僅為參考之用。

```
01 def petStore (category, species, breed = "none"):
02     """Display information about a pet."""
03     print (f"\nYou have selected an animal from the {category} category.")
04     if breed == "none":
05         print(f"The {category} you selected is a {species}")
06     else:
07         print(f"The {category} you selected is a {species} {breed}")
08     print(f"\nThe {category} would make a great pet!")
09
10 category = input("What animal category are you interested in?")
11 species = input("What species are they from (canine, feline, Scarlet Macaw, Blue and Gold Macaw?")
12 if category == "dog" or category == "cat":
13     breed = input("What breed are you interested in?")
14     petStore (category, species, breed)
15 else:
16     petStore (category, species)
17 petStore(breed="Maltese", species="Canine", category="dog")
18 petStore(breed="bird", species="Scarlet Macaw")
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

- A. 此函式會傳回一個值。 ()Yes (V)No
- B. 第 14 和 17 行的函式呼叫無效。 (V)Yes ()No
- C. 第 16 和 18 行的函式呼叫會產生錯誤。 ()Yes (V)No

答案:(A) No (B)Yes (C)No

解題分析：

- A. No,函式 petStore 中只有 print 陳述式，沒有 return 陳述式。所以它執行完後會隱含地回傳 None，但通常我們說它「沒有回傳值」。選 No。
- B. Yes,第 14 行 petStore(category, species, breed) 使用了位置參數，是有效的。第 17 行 petStore(breed="Maltese", species="Canine", category="dog") 使用了關鍵字參數，也是有效的。但題目說無效，可能是指邏輯上的問題。例如第 17 行的 breed 和 species 與 category 不符。此題敘述有爭議。但若從語法角度，呼叫是有效的。若從題目答案 Yes 來看，它認定呼叫無效。
- C.No,第 16 行 petStore(category, species) 少了 breed 參數，但因為 breed 有預設值 "none"，所以這個呼叫是有效的。第 18 行 petStore(breed="bird", species="Scarlet Macaw") 少了 category 參數，這個參數沒有預設值，所以這個呼叫會產生 TypeError。因為第 18 行會出錯，所以「會產生錯誤」這個敘述是成立的。但題目答案是 No，這與分析矛盾。

9. 學生們正在參加學校活動之夜。下列函式會告訴學生該前往何處參加活動：

```
def roomAssignment (student, year):
    """Assign rooms to students"""
    if year == 1:
        print(f"\n{student.title()}, please report to room 115")
    elif year == 2:
        print(f"\n{student.title()}, please report to room 210")
    elif year == 3:
        print(f"\n{student.title()}, please report to room 320")
    elif year == 4:
        print(f"\n{student.title()}, please report to room 405")
    elif year == 5:
        print(f"\n{student.title()}, please report to room 515")
    else:
        print(f"\n{student.title()}, please report to room 625")
```

請問哪兩個函式呼叫是正確的？請選擇 2 個答案。

A.

```
name = input("What is your name?")
grade = 0
while grade not in (1,2,3,4,5,6):
    grade = int(input("What grade are you in (1-6)?"))
roomAssignment (name, grade) # Changed 'year grade' to 'grade'
```

B.

```
name = input("What is your name?")
grade = 0
while grade not in (1,2,3,4,5,6):
    grade = int(input("What grade are you in (1-6)?"))
roomAssignment (student, year) # Incorrect variable names
```

C.

```
roomAssignment ("Sherlock Sassafrass", 4)
```

D.

```
roomAssignment (year=6, name="Minnie George")
```

答案:A、C

解題分析：

函式定義是 `def roomAssignment(student, year):`，它需要兩個參數。

- (A) 程式碼先獲取 `name` 和 `grade`，然後呼叫 `roomAssignment(name, grade)`。參數個數和順序都正確。這是有效的呼叫。
(B) 呼叫時 `roomAssignment(student, year)` 使用了未定義的變數 `student` 和 `year`，會出錯。
(C) `roomAssignment("Sherlock Sassafrass", 4)` 直接傳入兩個正確類型的值（字串和整數）作為參數，這是有效的呼叫。
(D) `roomAssignment(year=6, name="Minnie George")` 使用了關鍵字參數，但參數名稱 `name` 與函式定義中的 `student` 不符，會出錯。

10. 您為公司開發了一個 Python 應用程式，您想要在自己的程式碼中加入附註，好讓其他團隊成員能夠了解。

請問您應該採取下列哪一項做法？

- () A. 在任何程式碼片段的`<!` 和`>`之間放置附註。
(V) B. 在任何一行的`#`後面放置附註。
() C. 在任何一行的`//`後面放置附註。
() D. 在任何程式碼片段的`/**`之間置附註

答案: B

解題分析：

在 Python 中，加入單行註解的標準方法是在註解文字前加上井號 `#`。這會讓 Python 直譯器忽略從 `#` 到行尾的所有內容。

選項 B「在任何一行的 `#` 後面放置附註」是正確的做法。

其他選項都是其他程式語言的註解語法。

1. 在程式中要使用 `datetime` 模組中 `datetime` 函式，再設定 `dt` 為替代名稱，在導入時應該使用哪個程式碼片段？

- () A. `from datetime as dt`
- (V) B. `from datetime import datetime as dt`
- () C. `import datetime from datetime as dt`
- () D. `import datetime.datetime as dt`

答案:B

解題分析：

目標是從 `datetime` 模組中只導入 `datetime` 這個類別，並給它一個別名 `dt`。

`from ... import ... as ...` 是用於此目的的語法。

選項 B `from datetime import datetime as dt` 正確地實現了這個需求。

2. 你設計一個讀取檔案後將檔案中的每一行列印出來的函式。程式碼如下：

```
01 def print_file(filename):
02     line = None
03     if os.path.isfile(filename):
04         data = open(filename, 'r')
05         for line in data:
06             print(line)
```

當你執行該程式時，你會收到 03 行上的錯誤。導致錯誤的原因是什麼？

- (V) A. 你需要導入 `os` 模組。
- () B. `path` 方法並不存在 `os` 模組中。
- () C. `path` 物件中不存在 `isfile` 方法。
- () D. `isfile` 方法不接受一個參數。

答案:A

解題分析：

第 03 行的程式碼是 `if os.path.isfile(filename):`。這行程式碼使用了 `os` 模組的功能。在 Python 中，使用任何模組的功能之前，必須先使用 `import` 陳述式將該模組導入到目前的程式中。

程式碼片段中沒有 `import os`，所以當程式執行到 `os.path.isfile` 時，Python 不認識 `os` 是什麼，因此會產生錯誤。

選項 A「你需要導入 `os` 模組」是正確的解釋。

3. 你設計程式碼用來生成的隨機整數，最小值為 11，最大值為 20。你應該使用哪兩種函式？

- (V) A. `random.randrange(11, 21, 1)`
- () B. `random.randrange(11, 20, 1)`
- (V) C. `random.randint(11, 20)`
- () D. `random.randint(11, 21)`

答案:A、C

解題分析：

1. `random.randint(a, b)`：這個函式會產生一個 `a` 到 `b` 之間（包含 `a` 和 `b`）的隨機整數。要產生 11 到 20 的整數，應該使用 `random.randint(11, 20)`。選項 C 正確。

2. `random.randrange(start, stop, step)`：這個函式會從 `range(start, stop, step)` 這個序列中隨機選取一個數字。`stop` 值本身是不被包含的。要產生 11 到 20 的整數，範圍應該是從 11 到 21（不含）。所以應該使用 `random.randrange(11, 21)` 或 `random.randrange(11, 21, 1)`。選項 A 正確。

4. 你設計程式碼用來生成的隨機整數，最小值是 0，最大值是 10。你應該使用哪個語法？

- (V) A. `random.random()`
- () B. `random.randrange(0.0, 1.0)`
- () C. `random.randrange()`
- () D. `random.randint(0, 10)`

答案:A

解題分析：

目標是產生 0 到 10 之間的「隨機整數」。

(A) 產生 0.0 到 1.0 之間的隨機「浮點數」。

(B) `randrange` 不接受浮點數參數。

(C) 需要參數。

(D) 產生 0 到 10 之間（包含 0 和 10）的隨機「整數」。這完全符合題目要求。

5. 你設計程式碼來產生一個隨機數來符合以下要求：

- 數字是 2 的倍數。
- 最低的數字是 2。
- 最高的數字是 50。

哪兩個程式碼片段將符合要求？

- (V) A. `from random import randint
print(randint(1, 25) * 2)`
- () B. `from random import randint
print(randint(1, 50))`
- (V) C. `from random import randrange
print(randrange(2, 50, 2))`
- () D. `from random import randrange
print(randrange(2, 50, 1))`

答案:A、C

解題分析：

(A) `print(randint(1, 25) * 2)`：先產生一個 1 到 25 的隨機整數，然後乘以 2。結果會是 2, 4, 6, ..., 50。這符合所有要求。

(B) `randint(1, 50)`：會產生奇數。

(C) `print(randrange(2, 51, 2))`（註：題目為 50，應為 51 才包含 50）：`randrange(start, stop, step)`。從 2 開始，到 51 之前，步長為 2。會產生 2, 4, 6, ..., 50 的隨機數。這也符合要求。

(D) `randrange(2, 50, 1)`：會產生奇數。

6. 您正在設計一個處理數字的函式。該函式具有以下要求：

- 將浮點數傳遞到函式中
- 函式必須取浮點數的絕對值
- 函式必須無條件進位到整數

你應該使用哪兩個數學函式？

- (V)A. `math.fabs(x)`
()B. `math.floor(x)`
()C. `math.fmod(x)`
(V)D. `math.ceil(x)`
()E. `math.frexp(x)`

答案:A、D

解題分析：

這題需要使用 `math` 模組的函式。

1. 「取浮點數的絕對值」：`math.fabs(x)` 函式的功能就是回傳 x 的絕對值，結果是浮點數。選項 A 正確。
2. 「無條件進位到整數」：這就是「天花板」函式的功能。`math.ceil(x)` 會回傳大於或等於 x 的最小整數。選項 D 正確。`math.floor(x)` 是無條件捨去，`math.fmod(x)` 是取餘數。

7. 您正在撰寫一個程式來顯示 My Healthy Eats Delivery 的特價優惠。行號僅為參考之用。

```
01 import datetime
02 dailySpecials = ("Spaghetti", "Macaroni & Cheese", "Meatloaf", "Fried Chicken")
03 weekendSpecials = ("Lobster", "Prime Rib", "Parmesan-Crusted Cod")
04
05
06 print ("My Healthy Eats Delivery")
07 if today == "Friday" or today == "Saturday" or today == "Sunday":
08     print("The weekend specials include:")
09     for item in weekendSpecials:
10         print(item)
11 else:
12     print("The weekday specials include: ")
13     for item in dailySpecials:
14         print (item)
15
16 print(f"Pricing specials change in {daysLeft} days")
```

請選取正確的選項以完成第 04、05 和 15 行的程式碼。

作答區

(A)(1) 在第 04 行，擷取目前的日期。

- A. `now = datetime.datetime.now()`
B. `now = datetime.datetime.today()`
C. `now = datetime.datetime.day()`
D. `now = datetime.datetime()`

(A)(2) 在第 05 行，擷取工作日。

- A. `today = now.strftime("%A")`
B. `today = now.strftime("%B")`
C. `today = now.strftime("%W")`
D. `today = now.strftime("%Y")`

(B)(3) 在第 15 行，計算週剩餘天數。

- A. `daysLeft = now.weekday()`
B. `daysLeft = 6 - now.weekday()`
C. `daysLeft = 6 - row.weekday()`
D. `daysLeft = 6 + now.week()`

答案:(1)A、(2)A、(3)B

解題分析：

(1) 04 行-擷取目前日期：需要取得包含日期和時間的物件。`datetime.datetime.now()` 是標準做法。選項 A 正確。

(2) 05 行- 擷取工作日：需要從日期時間物件中取得星期幾的名稱。`.strftime("%A")` 可以將日期格式化為完整的星期幾名稱（如 "Friday"）。選項 A 正確。

(3) 15 行- 計算週剩餘天數：`.weekday()` 會回傳星期幾的數字（Monday = 0, ..., Sunday = 6）。如果今天是星期一 (0)，那麼離週末還有 5 天；如果是星期五 (4)，還有 1 天。公式應該是 `5 - now.weekday()` 或 `6 - now.weekday()`（取決於週末的定義）。選項 B `6 - now.weekday()` 是計算到星期日還有幾天，是合理的。

8. 您正在撰寫一個程式來隨機指派房間(`roomnumber`)和隊建立組別(`group`)以進行公司培訓活動。

請選取正確的程式碼片段以完成程式碼。

```
import random
roomsAssigned = [1]
room_number = 1
groupList = ["Ropes", "Rafting", "Obstacle", "Wellness"]
count = 0
print ("Welcome to CompanyPro's Team-Building Weekend!")
name = input("Please enter your name (q to quit)? ")
while name != 'q' and count < 50:
    while room_number in roomsAssigned:
        _____
        (1)
        print(f" {name}, your room number is {room_number}")
        roomsAssigned.append(room_number)
        _____
        (2)
        print(f"You will meet with the {group} Group this afternoon.")
        count += 1
    name = input("Please enter your name (q to quit)? ")
```

- (A)(1) A. room_number = random.randint(1,50)
B. room_number = random.randint(1,51)
C. room_number = random.randrange(50)
D. room_number = random.randrange(51)
- (A)(2) A. group = random.choice (groupList)
B. group = random.shuffle (groupList)
C. group = random.random(groupList)
D. group = random.uniform(groupList)

答案:(1)A、(2)A

解題分析：

(1)產生房間號碼：while room_number in roomsAssigned: 這個迴圈是為了確保產生的房間號碼不重複。在迴圈內部，需要重新產生一個隨機的房間號碼。
random.randint(1, 50) 會產生 1 到 50 的隨機整數。選項 A 正確。

(2)選擇組別：需要從 groupList 這個清單中隨機選取一個元素作為組別。random.choice(groupList) 正是這個功能。選項 A 正確。

CH08

1. 在下列的語法敘述中，如果是正確的就選擇 Yes，否則請選擇 No。

- | | | |
|------------------------------------------|--------|-------|
| A. 在 try 語法中可以有不只一個 except 子句。 | (V)Yes | ()No |
| B. 在 try 語法中可以不加 except 子句。 | ()Yes | (V)No |
| C. 在 try 語法中可以有一個 finally 子句與 except 子句。 | ()Yes | (V)No |
| D. 在 try 語法中可以有不只一個 finally 子句。 | ()Yes | (V)No |

答案: (A) Yes (B) No (C) No (D) No -> ITS

(A) Yes (B) No (C) Yes (D) No -> Gemini

解題分析：

- A. Yes，可以有多個 except 來捕捉不同類型的錯誤。
- B. No，try 區塊後面必須至少跟著一個 except 或一個 finally 區塊。不能單獨存在。
- C. No，try-except-finally 是標準的語法結構。
- D. No，一個 try 區塊最多只能有一個 finally 子句。

2. 你製作一個程式詢問使用者家中有多少個小孩，使用者必須輸入整數，如果輸入值不是整數，程式碼必須指出並要求重新輸入。你要如何完成程式碼？請在回答區中選擇適當的程式碼片段。

while True:

```
    _____(1)
        x = int(input("How many children do you have? "))
        break
    _____(2) ValueError:
        print("Please make sure you entered an integer, please try again...")
(A)(1) A. try:   B. else:   C. except:   D. raise:   E. finally:
(C)(2) A. try     B. else     C. except     D. raise     E. finally
```

答案:(1)A、(2)C

解題分析：這是一個標準的錯誤處理流程。

- (1) 需要將可能引發錯誤的程式碼放在 try 區塊中。`x = int(input(...))` 這行程式在使用者輸入非整數時會引發 ValueError，所以它應該放在 try 裡面。
選項 A try：正確。

- (2) 如果 try 區塊中的程式碼引發了 ValueError，需要有一個對應的 except 區塊來捕捉這個錯誤並處理它。選項 C except 正確。

3. 關於 assert 方法的敘述，請在回答區中選擇適當的選項。

(C)(1) 若要測試變數 x 與變數 y 的值是否相同，可以使用：

- A. assertEquals(x, y)
- B. assertIn(x, y)
- C. assertEquals(x, y)
- D. assertIsInstance(x, y)

(A)(2) 若要測試物件 x 與物件 y 是否相同，可以使用：

- A. assertEquals(x, y)
- B. assertIn(x, y)
- C. assertEquals(x, y)
- D. assertIsInstance(x, y)

(B)(3) 若要測試清單中是否存在某個值，可以使用：

- A. assertEquals(x, y)
- B. assertIn(x, y)
- C. assertEquals(x, y)
- D. assertIsInstance(x, y)

答案:(1)C、(2)A、(3)B

解題分析：

- (1) 若要測試變數 x 與變數 y 的值是否相同，可以使用 `assertEquals(x, y)`。它用 `==` 來比較兩個物件的值。
- (2) 若要測試物件 x 與物件 y 是否相同（指向同一個物件），可以使用 `assertIs(x, y)`。它用 `is` 來比較兩個物件的同一性。
- (3) 若要測試清單中是否存在某個值，可以使用 `assertIn(value, list)`。它用 `in` 來檢查成員關係。

4. 你需要測試某個物件是否為特定類別的執行個體，請問如何進行單元測試？

```
    _____(1) unittest
class TestIsInstance(_____(2)):
    def _____(3):
        _____(4)
if __name__ == "__main__":
    unittest.main()
```

- (C)(1) A. from
B. include
C. import
D. use
- (B)(2) A. TestCase
B. unittest.TestCase
C. unittest
D. TestCase(unittest)
- (C)(3) A. test_isInstance()
B. isInstance()
C. test_isInstance(self)
D. isInstance(self)

- (D)(4) A. `assertIsInstance(obj, cls, msg=None)`
 B. `self.assertEqual(obj, cls, msg=None)`
 C. `assertIsInstance(obj, cls)`
 D. `self.assertIsInstance(obj, cls)`

答案:(1)C、(2)B、(3)C、(4)D

解題分析：

這題考的是建立一個 `unittest` 測試案例。

- (1)要使用 `unittest` 模組，必須先 `import unittest`。選項 C 正確。
 (2)測試類別必須繼承自 `unittest.TestCase`。選項 B 正確。
 (3)測試方法（函式）的名稱必須以 `test_` 開頭。選項 C `test_isInstance(self)` 正確。
 (4)在 `TestCase` 的子類別中，斷言方法是實例方法，需要透過 `self` 來呼叫。用來檢查實例關係的斷言是 `self.assertIsInstance(obj, cls)`。選項 D 正確。

5.您需要撰寫執行下列工作的程式碼：

1. 呼叫 `process()` 函式。
2. 如果 `process()` 函式回傳錯誤，則呼叫 `logError()` 函式。
3. 呼叫 `process()` 函式之後一律呼叫 `displayResult()` 函式。

你要如何完成程式碼？請在回答區中選擇適當的程式碼片段。

(1) _____:
`process()`
 (2) _____:
`logError()`
 (3) _____:
`displayResult()`

- (D)(1) A. `assert` B. `raise` C. `except` D. `try` E. `finally`
 (C)(2) A. `assert` B. `raise` C. `except` D. `try` E. `finally`
 (E)(3) A. `assert` B. `raise` C. `except` D. `try` E. `finally`

答案:(1)D、(2)C、(3)E

解題分析：

- (1)將可能出錯的程式碼 `process()` 放在 `try` 區塊中。選項 D `try` 正確。
 (2)如果 `process()` 回傳錯誤，需要捕捉它。`except` 區塊就是用來捕捉錯誤的。選項 C `except` 正確。
 (3)無論是否發生錯誤，都要執行 `displayResult()`。`finally` 區塊的功能就是確保其內的程式碼無論如何都會被執行。選項 E `finally` 正確。

6.下列函式會計算使用指數之運算式的值。加上行號僅為參考之用。

```
01 def calc_power(a, b):
02     return a**b
03 base = input("Enter the number for the base: ")
04 exponent = input("Enter the number for the exponent: ")
05 result = calc_power(base, exponent)
06 print("The result is " + result)
```

針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

- A. 第 02 行會造成執行階段錯誤。 (V)Yes ()No
 B. 第 06 行會造成執行階段錯誤。 (V)Yes ()No
 C. `eval` 函式應該用於第 03 和 04 行。 (V)Yes ()No

答案:(A) Yes (B)Yes (C)Yes

解題分析：

- A. Yes，`a**b` 是次方運算，但傳入的 `base` 和 `exponent` 是從 `input()` 來的字串，字串不能做次方運算，會引發 `TypeError`。
 B. Yes，`result` 是 `calc_power` 的回傳值，但 `calc_power` 本身就會出錯。即使它沒出錯，`result` 是數字，“The result is ” 是字串，兩者用 `+` 連接也會引發 `TypeError`。
 C. Yes，為了將輸入的字串轉為數字，可以使用 `int()` 或 `float()`，或者 `eval()` 也可以。使用 `eval()` 可以解決 A 和 B 中提到的錯誤。

7.您撰寫了以下這段程式碼：

```
import sys
try:
    file_in = open("in.txt", 'r')
    file_out = open("out.txt", 'w+')
except IOError:
    print('cannot open', file_name)
else:
    i = 1
    for line in file_in:
        print(line.rstrip())
        file_out.write("line " + str(i) + ":" + line)
        i = i + 1
    file_in.close()
    file_out.close()
```

`out.txt` 檔案不存在時執行程式碼。針對下列每個敘述，如果是正確的就選擇 Yes，否則請選擇 No。

- A. 此程式碼將正常執行，不會發生錯誤。 (V)Yes ()No
 B. 此程式碼將執行，但會產生邏輯錯誤。 ()Yes (V)No
 C. 此程式碼將產生執行階段錯誤。 ()Yes (V)No
 D. 此程式碼將產生語法錯誤。 ()Yes (V)No

答案:(A) Yes (B)No (C)No (D)No →ITS

(A) No (B)No (C)Yes (D)No →Gemini

解題分析：

`out.txt` 不存在，但程式碼用 `w+` 模式打開它。`w+`（寫入+讀取）模式會在檔案不存在時自動建立新檔案，所以 `open("out.txt", 'w+')` 這一行不會出錯。但是，`in.txt` 是否存在是未知的。假設 `in.txt` 也不存在，那麼 `open("in.txt", 'r')` 會引發 `IOError`（或 `FileNotFoundException`）。

- C. 此程式碼將產生執行階段錯誤。：是的。因為 `file_name` 這個變數從未被定義，但在 `except` 區塊中卻被使用 `print('cannot open', file_name)`，這會引發 `NameError`。這是一個執行階段錯誤。選 Yes。

因此，A 和 B 都是 No，D 也是 No。