

# 大學程式能力檢定(CPE)

## 考生使用手冊

2023 年 11 月 28 日

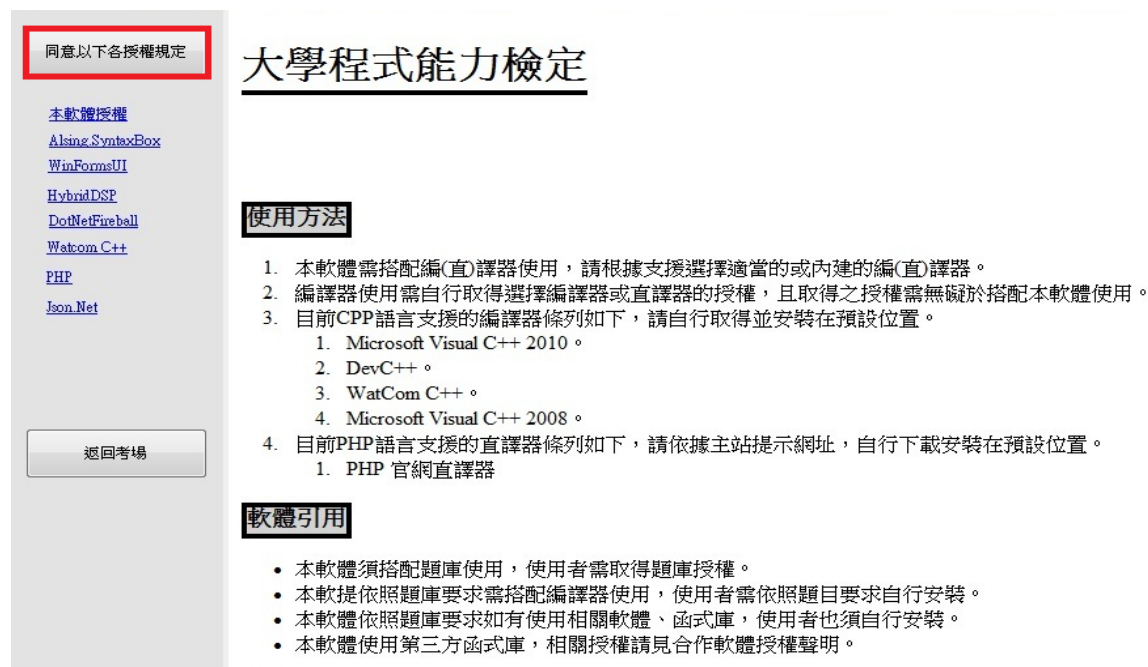
這份手冊提供給參加 CPE 檢定考試的考生。內容包含考試環境的使用，以及解題時所使用 I/O 的基本知識。

1. 如欲報名參加 CPE 考試，請先於 CPE 網站完成帳號註冊，然後再報名該次考試。若已有帳號，可免註冊，但仍需報名，才能參加考試。報名時，請慎選考場，並於指定時間前往報到、考試。
2. 自 2013/10/1 起的 CPE 考試，採用新的評判系統「瘋狂程設」：  
<http://coding-frenzy.arping.me/>  
考生可至「瘋狂程設」另外註冊後，自行上網練習。(注意：CPE 帳號與瘋狂程設帳號不同，參加 CPE 考試須使用 CPE 帳號，在瘋狂程設練習須使用瘋狂程設帳號)
3. CPE 網站 <http://cpe.cse.nsysu.edu.tw/> 內，「環境與教材」有更詳細資料。

# 1. CPE 考試環境

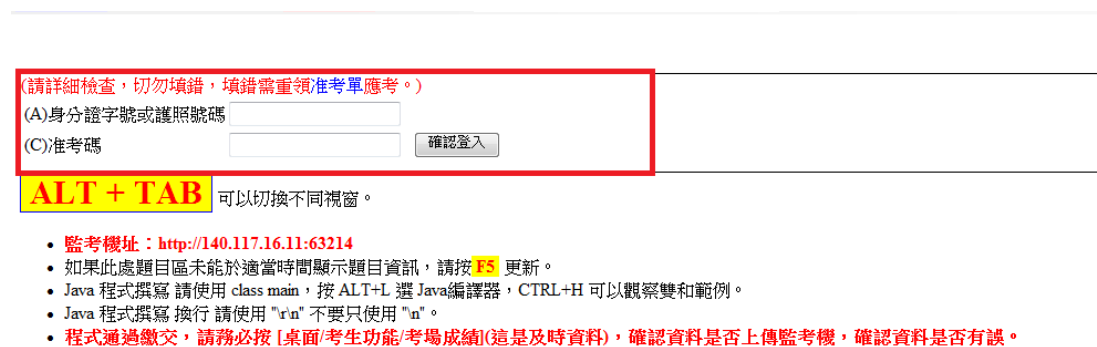
## 1.1 瘋狂程設：系統註冊與登入

請點選「同意以下各授權規定」進入瘋狂程設系統的登入畫面，如圖 1.1.1。



(圖 1.1.1)

進入後，請填寫「身分證」與「准考碼」(准考碼於考前由監考人員發放)完成登入動作，如圖 1.1.2。



## 題目

請先登入系統

## 大學程式能力檢定(CPE)考生注意事項

1. 考試時程：
  1. 考生報到時間：17:30~17:40。
  2. 考生進行考試環境測試：17:40~18:30。
  3. 休息與準備：18:30~18:40。
  4. 正式考試：18:40~21:40。
  5. 考生於 18:00 後，不准入場。
  6. 考生於 19:40 後，始得離場不再考試。

(圖 1.1.2)

完成登入後的畫面如圖 1.1.3，考生可看到考生功能，現在時間，以及考試題目列表。



(圖 1.1.3)

## 1.2 考生功能

包括問題提問，觀看考官問題回應、成績欄、英漢字典和一些額外的工具，如圖 1.2.1。



(圖 1.2.1)

### 1.2.1 電子發問

點擊後出現一張發問單，請填寫欲詢問的問題，與相關資訊(題號、問題描述)，如圖 1.2.2。問題送出後，只有被監考官回答的問題才會顯示出來，如圖 1.2.3。

- **中文輸入法:**在考場中，應該已經設定快捷鍵 **左 ALT+SHIFT+1**、**左 ALT+SHIFT+2**、**左 ALT+SHIFT+3** 來切換輸入法。
- **中文輸入法:**如果未設定，則需透過額外的視窗輸入，按住「CTRL」鍵與滑鼠左擊欲輸入的欄位，在產生的視窗中利用「新注音」輸入。  
考場上

(圖 1.2.2)

時間戳記	發問相關之題號	發問之內容	官方回應
------	---------	-------	------

(圖 1.2.3)

## 1.2.2 練習成績

展示本次 CPE 練習的考生即時成績(解題數、所花時間)，範圍包含全部考場的考生。此成績每五分鐘更新一次。

### 1.2.3 聯賽成績

展示本次 CPE 考試的考生即時成績(解題數、所花時間)，如圖 1.2.4，範圍包括全部考場的考生。此成績每五分鐘更新一次。

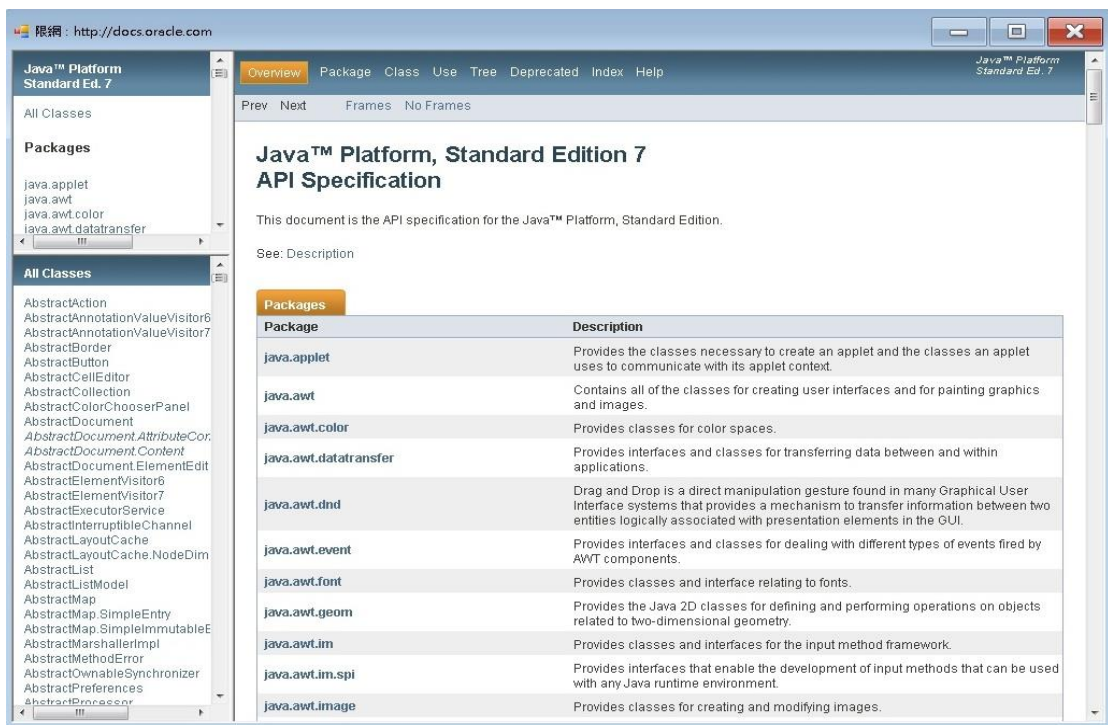
序	學校	場別	題數	解時	0081.UVA10019	0082.UVA12503	0083.UVA11639	0084.UVA11475	0085.UVA10055	0086.UVA10954	0087.UVA10090	身分證	核准	准考證、註冊碼
1	國立雲林科技大學	場C九	7題	7分	1 (0=)	1 (0=)	1 (0=)	1 (1=)	1 (1=)	1 (0=)	1 (1=)	146		YUNTECH01119-132472
2	國立雲林科技大學	場C七	7題	10分	1 (0=)	1 (0=)	1 (1=)	1 (1=)	1 (1=)	1 (1=)	1 (2=)	184		YUNTECH005188-896169
3	國立臺北大學	場C九	7題	15分	1 (2=)	1 (1=)	1 (1=)	1 (2=)	1 (2=)	1 (2=)	1 (3=)	380		NTPU005080-956328
4	長庚大學	場C四	7題	16分	1 (1=)	1 (1=)	1 (1=)	1 (2=)	1 (2=)	1 (2=)	1 (3=)	366		CGU109438-089653
5	國立臺灣大學	場C七	7題	46分	1 (3=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	360		NTTU002274-546402
6	長庚大學	場C四	7題	49分	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	166		CJCU001172-246972
7	國立臺中教育大學	場C八	7題	50分	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	182		NTCU017081-374512
8	國立中興大學	場C四	7題	52分	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	168		NCHU092240-752988
9	國立臺灣大學	場C四	7題	54分	1 (1=)	2 (1=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	966		NTTU007176-589840
10	國立中興大學	場C三	7題	70分	3 (3=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (5=)	706		NCHU091350-310261
11	大同大學	場C零	8題	14分	1 (2=)	1 (2=)	1 (2=)	1 (2=)	0 (0=)	1 (1=)	1 (2=)	726		TJU002138-993174
12	國立臺北大學	場C七	8題	22分	1 (2=)	1 (4=)	1 (2=)	1 (2=)	1 (1=)	1 (4=)	0 (0=)	722		NTPU003380-384079
13	國立臺北大學	場C二	8題	13分	1 (0=)	0 (0=)	1 (2=)	0 (0=)	1 (2=)	1 (2=)	1 (2=)	520		NTPU002083-124284 NTPU018033-010759
14	國立金門大學	場C二	8題	17分	1 (1=)	2 (0=)	1 (1=)	1 (3=)	1 (4=)	1 (4=)	1 (0=)	364		NQU001310-214058
15	國立金門大學	場C零	8題	22分	0 (0=)	0 (0=)	1 (4=)	1 (4=)	1 (4=)	1 (4=)	1 (0=)	540		NQU016431-790915
16	國立金門大學	場C一	8題	28分	1 (2=)	0 (0=)	1 (4=)	1 (5=)	1 (4=)	1 (4=)	0 (0=)	162		NQU007281-994375
17	國立金門大學	場C一	8題	32分	1 (3=)	0 (0=)	1 (4=)	1 (6=)	1 (4=)	1 (10=)	0 (0=)	560		NQU004293-906823
18	國立金門大學	場C二	8題	35分	1 (3=)	0 (0=)	1 (5=)	1 (7=)	1 (4=)	1 (10=)	0 (0=)	948		NQU013275-101949
19	國立金門大學	場C六	8題	36分	1 (3=)	0 (0=)	1 (6=)	1 (7=)	1 (4=)	1 (10=)	0 (0=)	546		NQU010991-503905
20	國立金門大學	場C五	8題	41分	1 (4=)	0 (0=)	1 (6=)	1 (9=)	1 (10=)	1 (10=)	0 (0=)	188		NQU002039-489559 NQU0081300-180450

(圖 1.2.4)

### 1.2.4 Java 參考資料

可查詢 Java 語法、資料型態、類別成員函式等，如圖 1.2.5。資料來源：

<http://docs.oracle.com/javase/7/docs/api/>

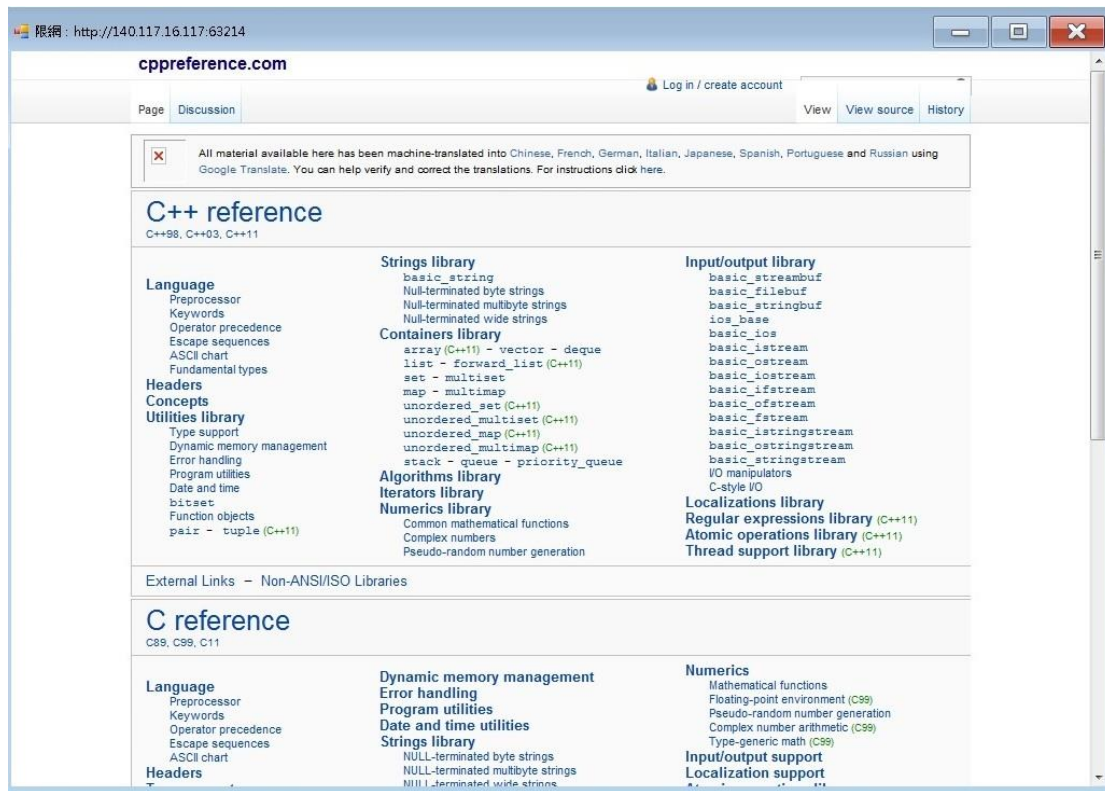


(圖 1.2.5)

## 1.2.5 C++參考資料

可查詢 C++ 語法、資料型態、類別成員函式等，如圖 1.2.6。資料來源：

<http://en.cppreference.com/w/>



(圖 1.2.6)

## 1.2.6 英漢字典

可以查詢英文單字的意義，但是題目中的英文單字無法直接複製貼上。

## 1.2.7 剩餘時間

考試的剩餘時間會顯示在視窗的右上角，如圖 1.2.7。



(圖 1.2.7)

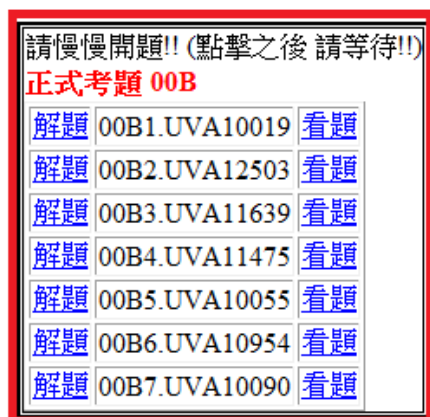
## 1.2.8 結束評量

結束本次考試。如果不小心按到，請跟監考官反映。

## 1.3 考試題目

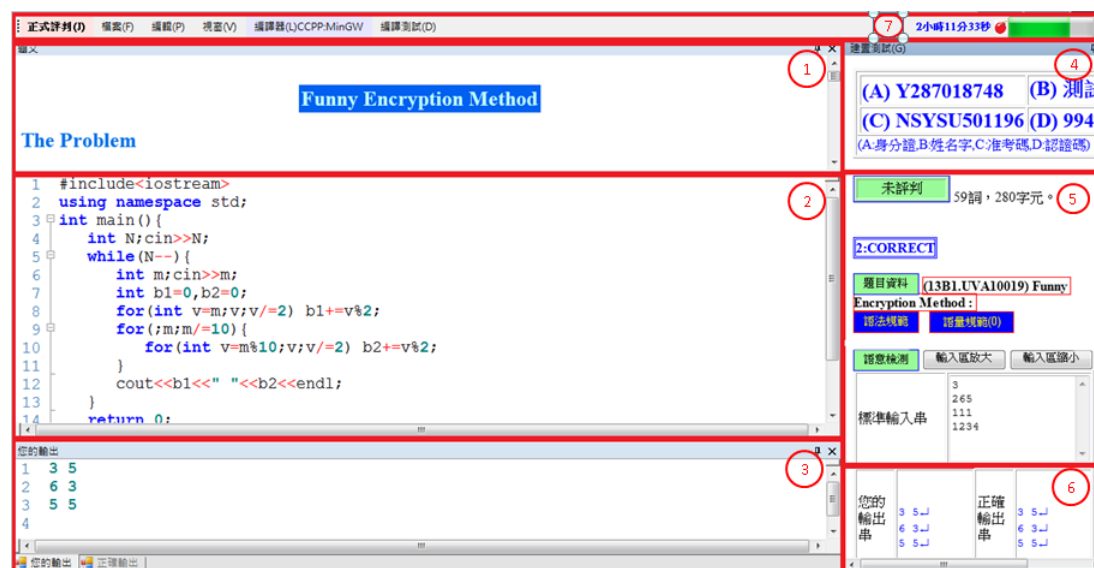
如圖 1.3.1，點選正式考題下方的「解題」，即可進入解題視窗。若下方沒有顯示考題，請重新刷新頁面。

### 題目



(圖 1.3.1)

## 1.4 考題介面



(圖 1.4.1)

進入題目後，視窗如圖 1.4.1，依照視窗各方框的編號，各部分功能如下：

- (1) 題目的介紹與描述。
- (2) 撰寫程式的視窗(請在此撰寫程式)。
- (3) 程式輸出與使用測資的正確輸出。
- (4) 顯示考生的身分證、姓名、准考碼。



- (5) 題目資料、語法規範、語量規範、語意檢測。語法規範會顯示編譯錯誤的資訊，語量規範是顯示程式碼中出現不允許的語法，語意檢測則需要輸入測試資料。
- (6) 根據語意檢測的輸入資料以及考生的程式碼所產生的輸出資料，左半邊為考生程式碼的輸出，右半邊為題目正確的輸出。
- (7) 功能選項，與剩餘時間。

以下介紹功能選項的各項功能。

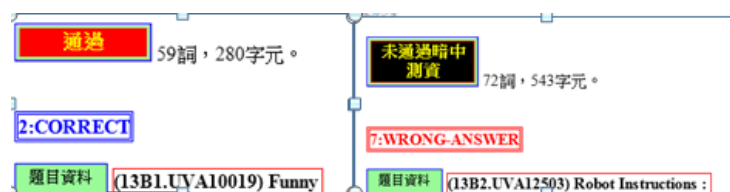
### 1.4.1 正式評判

這是考試的「正式評判」，評判的結果將列入考試的正式紀錄，也會顯示於所有考生的成績表上。如果評判的結果是不通過，將會依照計分規則，進行罰扣分。考生應於進行各項測試後，再執行「正式評判」。點選「正式評判」後，會出現確認視窗。在確認視窗中點選是(Y)，則可以「正式評判」考生的程式，如圖 1.4.2。



(圖 1.4.2)

考生進行評判後，如果考生通過此題的「暗中測試資料」(不公開)，則系統會顯示通過，反之顯示不通過，如圖 1.4.3。

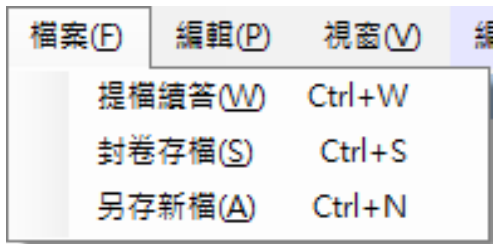


(圖 1.4.3)

### 1.4.2 檔案

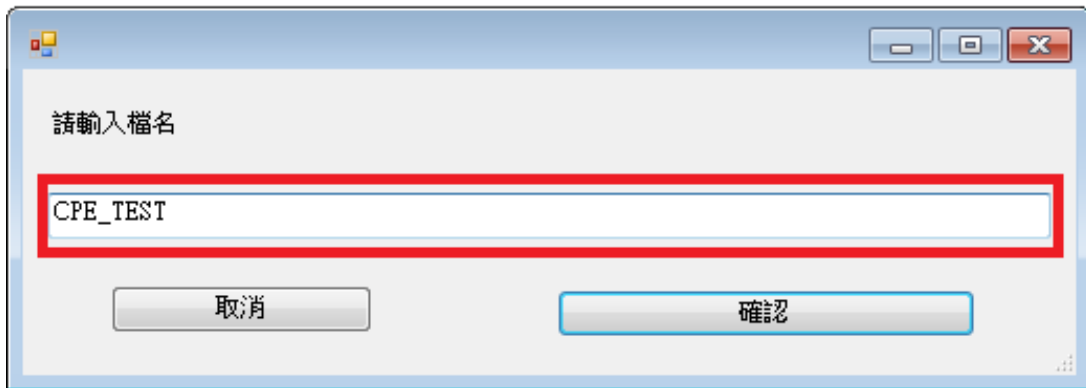
如圖 1.4.4，這選項的功能有提檔續答、封卷存檔、另存新檔。





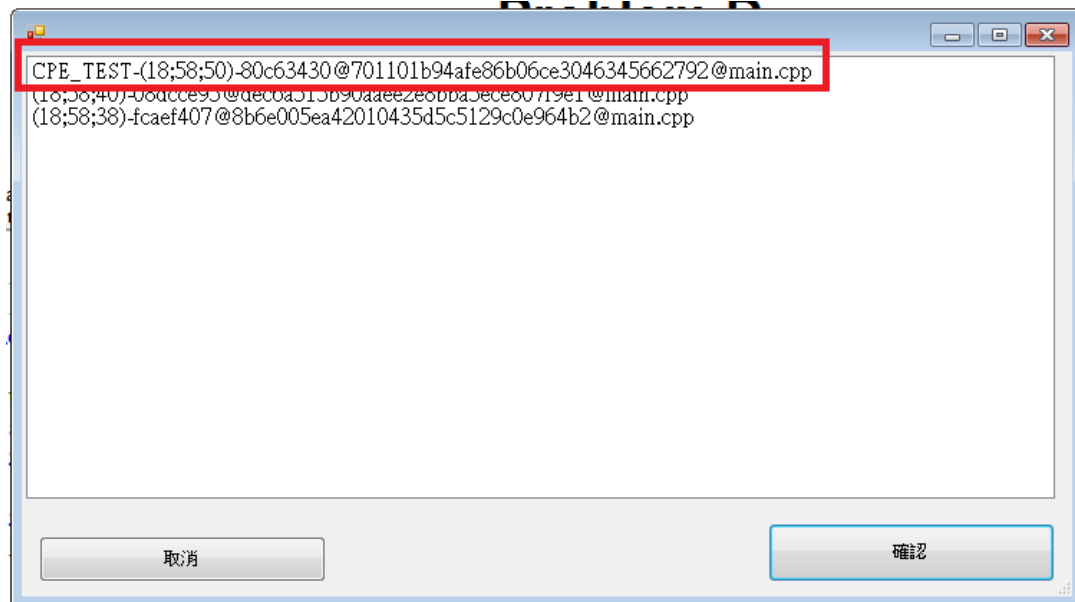
(圖 1.4.4)

**另存新檔:**可以將目前程式碼依照考生選擇的檔名存檔，如圖 1.4.5。



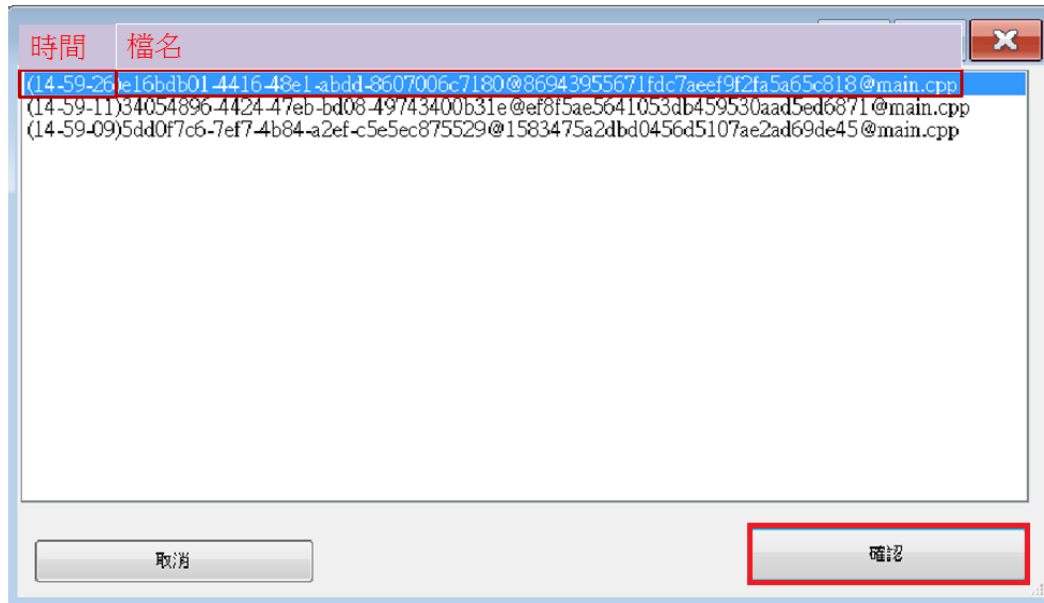
(圖 1.4.5)

**提檔續答:** 將已經儲存的檔案(例如，CPE\_TEST)開啟後繼續編寫，如圖 1.4.6



(圖 1.4.6)

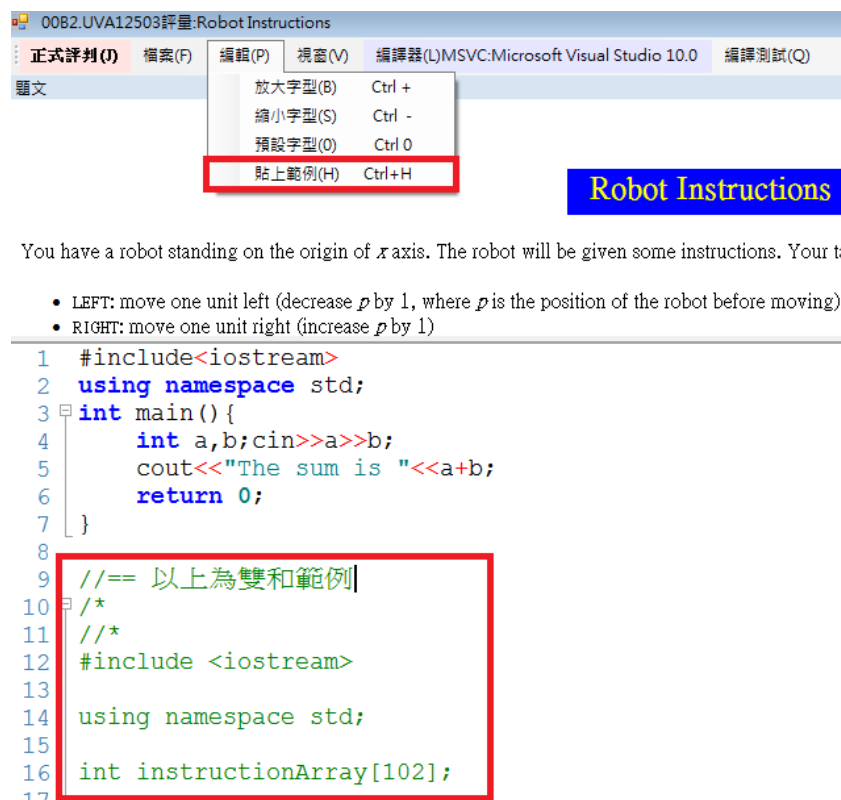
**封卷存檔:**將目前的程式碼儲存，存檔方式依照時間先後儲存，檔名則由系統隨機產生。之後，亦可以在提檔續答中選擇所儲存的檔案，如圖 1.4.7。



(圖 1.4.7)

### 1.4.3 編輯

此功能可以將程式碼字形放大、縮小，也可以在貼上範例程式碼供考生參考。貼上範例會以註解的形式產生在編輯視窗的最下面，如圖 1.4.8。



(圖 1.4.8)

### 1.4.4 視窗

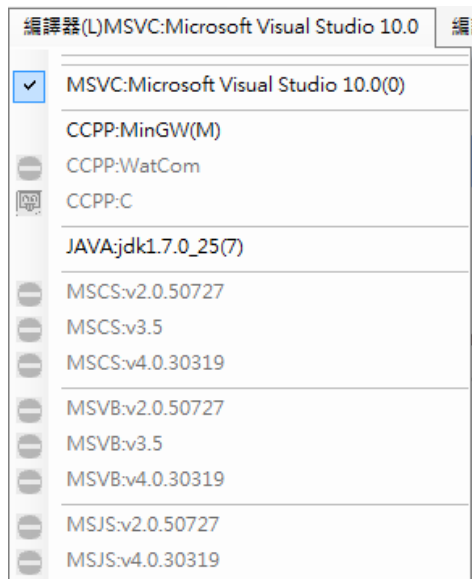
考生可以依照喜好將目前不需要資訊視窗關閉，如題目、建置測試、正確輸出、你的輸出這四種視窗。如要恢復相關視窗，再按一下即可恢復視窗，如圖 1.4.9。



(圖 1.4.9)

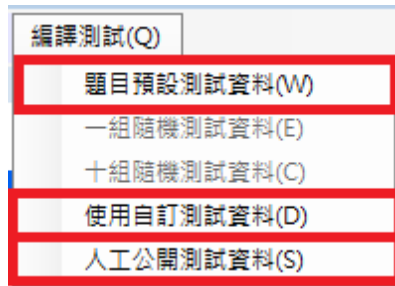
### 1.4.5 編譯器

考生在此選項可以選擇自己熟悉的語言編譯器(compiler)來編譯程式碼，但前提必須是題目所允許使用的語言，如圖 1.4.10。



(圖 1.4.10)

## 1.4.6 編譯測試



(圖 1.4.11)

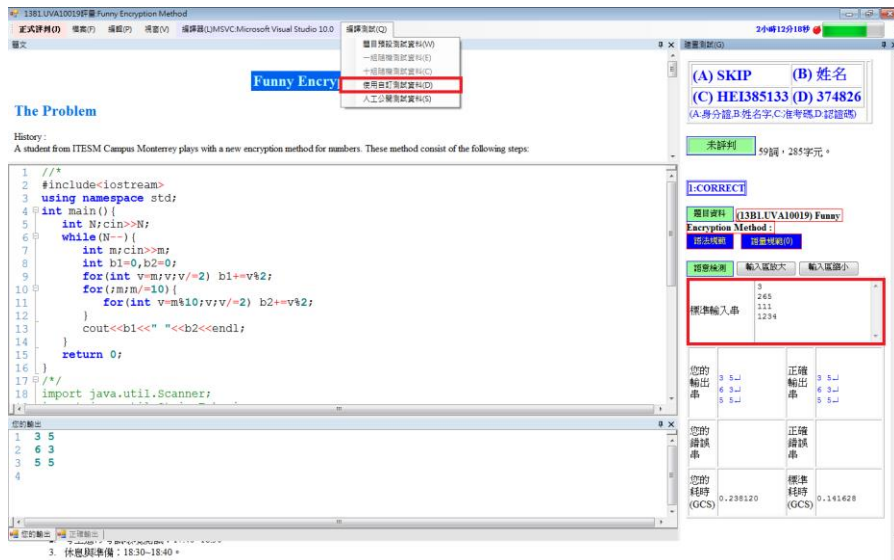
編譯測試為檢驗考生程式碼之各功能，如圖 1.4.11 所示，功能介紹如下：

### (1) 題目預設測試資料：

考生可使用此功能來執行並驗證程式碼，測試資料是由系統產生(難度較低)。此項測試不列入考試成績紀錄。

### (2) 使用自訂測試資料：

考生可使用此功能來執行並驗證程式碼，測試資料是由考生自行輸入，如圖 1.4.12。此項測試不列入考試成績紀錄。



(圖 1.4.12)

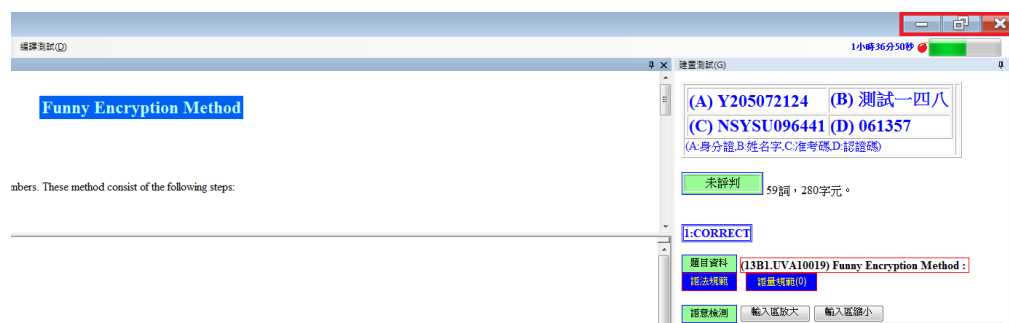
### (3) 人工公開測試資料：

考生可使用此功能來執行並驗證程式碼，測試資料是由選命題老師根據題意所擬定的測試資料，其困難度與系統內的「暗中測試資料」大約相同。此項測試不列入考試成績紀錄。建議考生在進行「正式評判」之前，先通過此「人工公開測試資料」之測試。

## 1.4.7 換題續答

考生若要回到最初視窗選擇題目，可以將所有題目視窗縮小或關閉即可回到

選擇題目視窗，如圖 1.4.13。



(圖 1.4.13)

## 1.5 測試後的訊息

考生進行測試或評判，系統將顯示相關訊息，如下表：

COMPILER-ERROR	程式碼未通過編譯。(點入連結可以查閱編譯器所產生的錯誤訊息。)
CORRECT	程式已經正確，並通過測試。
NO-OUTPUT	程式沒有輸出任何資料。
PENDING	送出的程式碼仍在處理中。
PRESENTATION-ERROR	輸出的結果正確，但格式錯誤，例如未依規定空格或換行(多空格或少空格，多換行或少換行)。
RUN-ERROR	無法順利將程式執行完畢，亦即程式執行過程發生錯誤，例如記憶體存取錯誤。
TIMELIMIT	程式執行所花費的時間超過題目限制。程式可能落入無窮迴圈，或是必須改進解題方法。
WRONG-ANSWER	輸出的結果錯誤。(若輸出的格式產生過大錯誤，也可能造成此結果。)

## 2. C/C++ 基本輸入與輸出

CPE 考試的程式設計，所有輸入與輸出均採取「標準輸入」(stdin)與標準輸出」(stdout)，不可使用檔案讀寫。撰寫程式時，於 C 語言，可使用如 scanf 與 printf 函式；於 C++，可使用如 cin 與 cout 物件。輸入與輸出資料全為純文字資料，必須完全依照题目的輸入與輸出格式撰寫程式。而且题目的「隱藏測試資料」之輸入資料，保證依照题目所描述的格式，因此考生不必檢查輸入資料的正確性。

如果考生不知如何讀取資料，並進一步嘗試撰寫题目所要求的功能，將是很可惜的。這裡將介紹一些常見的測試資料型式，以及讀取的方式，並附上實際题目的範例，希望能夠帶領你快速進入解題的狀況。每小節後面還有相關的練習題，嘗試作些練習，對 CPE 考試成績一定有所助益。

### 2.1 C 語言的 scanf 與 printf

談到輸入與輸出，對 C 語言來說便是 scanf 與 printf 兩個函式。其中，scanf 的書寫格式如下：

```
int scanf(const char *format, ...);
```

函式第一個參數是個字串，裡面你敘述要如何讀取接下來的輸入。對於你所打算讀取的資料格式與型態，依序填入對應的符號，如以下的表格：

字元	char*	%c	字串	char*	%s
		整數	正整數	八進位	十六進位
(unsigned) char*		%hhd	%hhu	%hho	%hhx
(unsigned) short*		%hd	%hu	%ho	%hx
(unsigned) int*		%d	%u	%o	%x
(unsigned) long*		%ld	%lu	%lo	%lx
(unsigned) long long*		%lld	%llu	%llo	%llx
浮點數	float*	%f	浮點數	double*	%lf

接下來的參數便依序填入存放對應資料變數的位址，即變數名稱前面加上&符號。特別注意，字串變數名稱本身已代表是位址，所以唯有字串不必再加上&符號。

例如，下列的程式：

```
char ch, str[64];
```

```
int num;
float value;

scanf("%c%s%d%f",&ch,str,&num,&value);
```

除了 %c 以外，各種輸入方式皆會忽略前方多餘的空白字元，即 space、tab 與 enter。因此輸入列中，只要資料的順序相同，其間如何空隔、或有多少空隔，其實都是無所謂的。像是下面這些情況，都能以 scanf("%d%f%f", ...) 的方式正確讀取，因此對我們來說都是一樣的。

3 ↓ 1.10 ↓ 2.20 ↓ 3.30	3.1.10.2.20.3.30	3 ↓ 1.10.2.20.3.30	3..1.10 ↓ ↓ .2.20...3.30
---------------------------------	------------------	-----------------------	--------------------------------

## 2.2 C++ 語言的 cin 與 cout

對於 C++ 而言，輸入與輸出就是 cin 與 cout。cin 會根據你給它的變數型態，讀取相應的資料，使用的運算元是指向變數的 >>。相對來說，簡單且直覺。

## 2.3 Python 語言的 input 與 print

對於 Python 而言，輸入與輸出就是 input 與 print。input 會根據你給它的變數型態，讀取相應的資料，使用的運算元是()。

## 2.4 讀入 n 筆資料

測試資料最常見的就是這種型式了。處理方式就是讀進這個 n，然後跑 n 次的迴圈。

C language

```
int main() {
    int n;
    scanf("%d",&n);
    while (n-->0) {
        /* 讀取每筆資料 */
    }
    return 0;
}
```



```
}  
}
```

C++

```
int main() {  
    int n;  
    cin>>n;  
    while (n-->0) {  
        // 讀取每筆資料  
    }  
    return 0;  
}
```

Python

```
n = int(input())  
for i in range(n):  
    // 讀取每筆資料
```

範例：

UVA10406: [Vito's family](#)

黑道老大 Vito 搬來街上住，希望與每位親戚家的距離總和要最短。

C Language

```
#include <stdio.h>  
#define MAX_R 100  
  
int num[MAX_R];  
  
int main() {  
    int n,r,s,i;  
    int sum;  
  
    scanf("%d",&n);
```

```
while (n--) {
    scanf("%d",&r);
    for (i=0;i<r;i++) {
        scanf("%d",&s);
        num[i]=s;
    }

    /* 剩下的留給你完成，提示：中位數。 */

    printf("%d\n",sum);
}

return 0;
}
```

## C++

```
#include <iostream>
#include <vector>
using namespace std;

vector<int> num;

int main() {
    int n,r,s;

    cin>>n;
    while (n-->0) {
        cin>>r;
        num.clear();
        for (int i=0;i<r;i++) {
            cin>>s;
            num.push_back(s);
        }

        // 剩下的留給你完成，提示：中位數。

        cout<<endl;
    }

    return 0;
}
```

## Python

```
n = int(input())
for i in range(n):
    arr = list(map(int, input().split()))
    r = arr[0]
    s = arr[1:]

    /* 剩下的留給你完成，提示：中位數。 */
```

其實，不同題目，可能有很多層的「讀入 n 筆資料」。

練習：

UVA10401: [Fibonaccimal Base](#)

UVA10403: [Funny Encryption Method](#)

UVA10408: [What is the Probability?](#)

## 2.5 讀至檔案結束

這種測試資料不會告訴你有多少筆資料，你必須一直處理到沒有資料為止。

### C language

檢查 `scanf` 函式的回傳值可以知道資料是否結束。平常 `scanf` 會回傳它成功讀進的元素個數；當讀到檔案結束時，`scanf` 則回傳 EOF。把它整合在 `while` 的判斷條件裡就成如下的程式：

```
int main() {
    int x;
    while (scanf("%d",&x)!=EOF) {
        /* 處理目前這筆資料 */
    }
    return 0;
}
```

### C++

若把 `cin` 放進 `while` 的判斷條件裡，`cin` 會自動轉型成 `void*`。而當檔案結束時，其值會變成 `NULL`，也就是 0，代表 `false`。所以我們寫成如下的程式：

```
int main() {
    int x;
    while (cin>>x) {
        // 處理目前這筆資料
    }
    return 0;
}
```

### Python

```
while True:
    x = input()
    if not x:
        break
    // 處理目前這筆資料
```

範例：

UVA10407: [Hashmat the brave warrior](#)

算出敵方軍隊與我方人數的差距。

C Language

```
#include <stdio.h>

int main() {
    long long a,b; /* 可能等於 2^32，所以 unsigned int 還不夠。 */
    while (scanf("%lld%lld",&a,&b)!=EOF) {
        /*試著完成它吧！ */
    }
    return 0;
}
```

請注意，scanf 對於 long long 要使用 %lld。

C++

```
#include <iostream>
using namespace std;

int main() {
    long long a,b; // 可能等於 2^32，所以 unsigned int 還不夠。
    while (cin>>a>>b) {
        //試著完成它吧！
    }
    return 0;
}
```

Python

```
while True:
    try:
        a, b = map(int, input().split())
```

```
        /*試著完成它吧！ */  
  
    except EOFError:  
  
        break
```

練習：

10400: [The 3n + 1 problem](#)

10405: [Jolly Jumpers](#) (混合兩種方式)

10411: [Back to High School Physics](#)

## 2.6 讀至 0 結束

這種類型也是時常出現的型式，只要簡單地加個判斷跳出迴圈即可。

C language

```
int main() {  
    int n;  
    while (scanf("%d",&n)!=EOF) {  
        if (n==0) break;  
  
        /* ... */  
    }  
    return 0;  
}
```

C++

```
int main() {  
    int n;  
    while (cin>>n) {  
        if (n==0) break;  
  
        // ...  
    }  
    return 0;  
}
```



## Python

```
while True:
    try:
        n = list(map(int, input().split()))
    except EOFError:
        break
    if n[0] == 0:
        break
    // ...
```

範例：

UVA10404: [Primary Arithmetic](#)

計算進行加法時進位的次數。

## C Language

```
#include <stdio.h>

int main() {
    int a,b;
    while (scanf("%d%d",&a,&b)!=EOF) {
        if (a==0&&b==0) break;

        /* 留給你完成 */
    }
    return 0;
}
```

## C++

```
#include <iostream>
using namespace std;

int main() {
    int a,b;
```

```

while (cin>>a>>b) {
    if (a==0&&b==0) break;

    /* 留給你完成 */
}
return 0;
}

```

### Python

```

while True:

    a, b = map(int , input().split())

    if(a == 0 and b == 0):

        break

    /* 留給你完成 */

```

練習：

10416: [Last Digit](#)

10418: [Minesweeper](#)

## 2.7 一次讀取一列資料直到檔案結束

一列的資料中可能包含空白，無法當一般字串處理，因此需要一次將一整列資料進行讀取。可以使用 C 語言的 `fgets`，或 C++ 語言的 `getline`。

### C language

```

#include <stdio.h>
int main() {
    char s[100];
    while ( fgets(s,100,stdin) != NULL ) { // 100 為一列的 size
        /* ... */
    }
    return 0;
}

```

C++

```
#include <cstring>
int main() {
    string s;
    while ( getline(cin,s) ) {
        /* ... */
    }
    return 0;
}
```

Python

```
while True:
    try:
        n,m=map(int,input().split())
        /* ... */
    except:
        break
```

範例：

UVA272: [TEX Quotes](#)

將配對的雙引號”和”替換成‘與’。

C language

```
int main() {
    char s[100005];
    while ( fgets(s,100005,stdin) != NULL ) {
        /* 留給你完成 */
    }
    return 0;
}
```

C++

```
int main() {  
    string s;  
    while ( getline(cin,s) ) {  
        /* 留給你完成 */  
    }  
    return 0;  
}
```

Python

```
while True:  
    try :  
        n = input()  
        /* 留給你完成 */  
    except EOFError:  
        break
```