目錄

任務題解流程

任務題解流程基本如下:

1. 任選擇任務
2. 新增程式檔案及編輯程式檔案
3. 新增輸入檔案將輸入資料及設為預設輸入: 這步驟可省略, 但好處是測試程式時可以不需要重複輸入。測試數據會自動在預設的檔案輸入到運行中的程式。
4. 本地測試程式運行
5. 提交程式
6. 留意提交狀態的更新
7. 選擇另一任務並重複以上步驟 ...

▲返回求助目錄▲

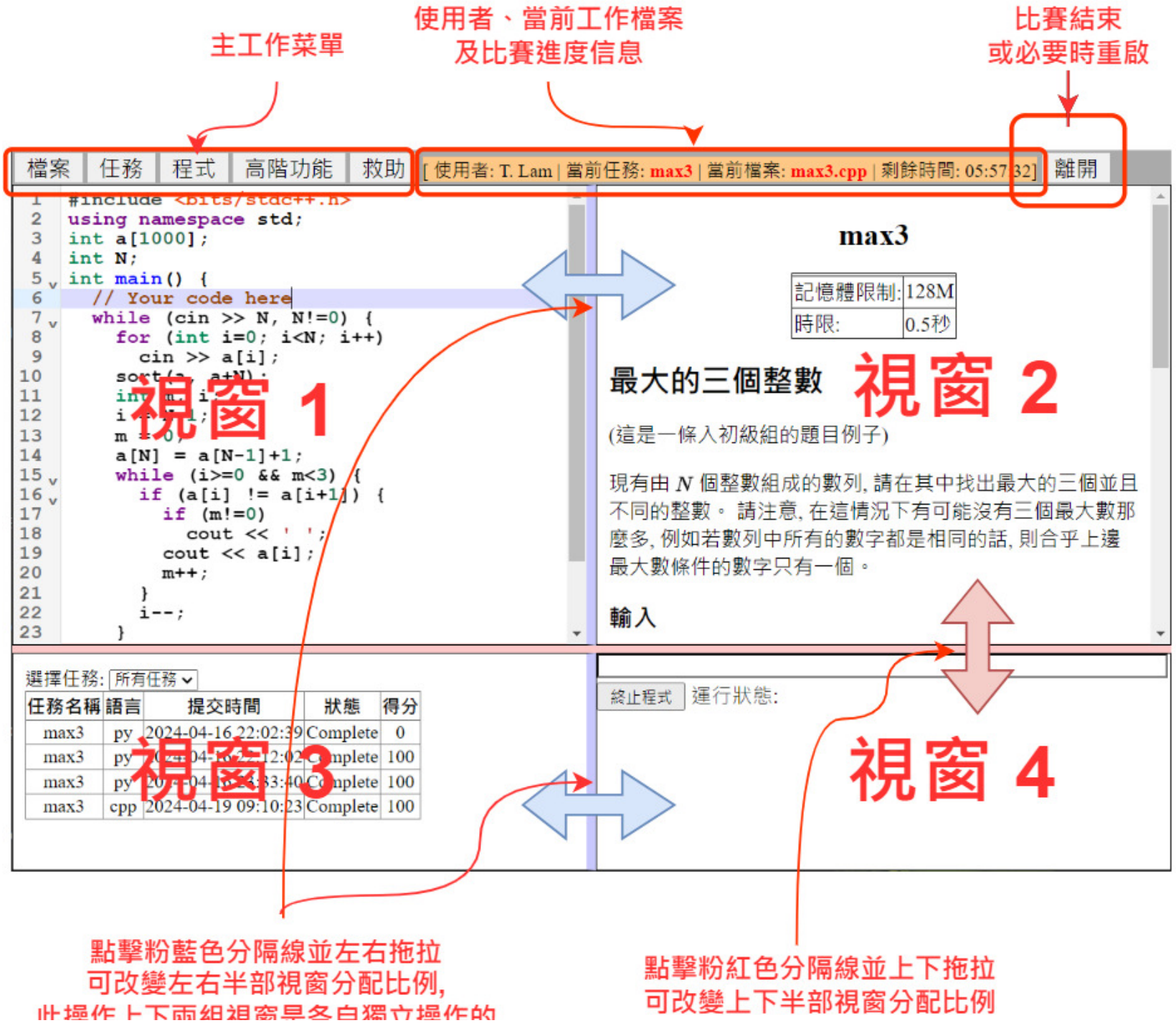
登入介紹

本系統是在 MOI 及 NOIP 系列比賽的編程系統。登入時須跟據你所參加的比賽而選擇不同的登入方式:

- MOI: 輸入使用者編號 (或電郵) 及密碼，再按下 "以 MOI 模式登入"
- NOIP: 先輸入使用者編號 (或電郵) 及密碼，再輸入比賽 server 的 URL (這網址會在比賽開始前提供), 再按下 "以 NOIP 模式登入"

▲返回求助目錄▲

主畫面


主工作菜單
使用者、當前工作檔案
及比賽進度信息
比賽結束
或必要時重啟
檔案 任務 程式 高階功能 救助
[使用者: T. Lam | 當前任務: max3 | 當前檔案: max3.cpp | 剩餘時間: 05:57:32]
離開
#include <bits/stdc++.h>
using namespace std;
int a[1000];
int N;
int main() {
// Your code here
while (cin >> N, N!=0) {
for (int i=0; i<N; i++)
cin >> a[i];
sort(a, a+N);
int m, i;
i = N-1;
m = 0;
a[N] = a[N-1]+1;
while (i>=0 && m<3) {
if (a[i] != a[i+1]) {
if (m!=0)
cout << ' ';
cout << a[i];
m++;
}
i--;
}
視窗 1
max3
記憶體限制: 128M
時限: 0.5秒
最大的三個整數
視窗 2
(這是一條入初級組的題目例子)
現有由 N 個整數組成的數列, 請在其中找出最大的三個並且不同的整數。 請注意, 在這情況下有可能沒有三個最大數那麼多, 例如若數列中所有的數字都是相同的話, 則合乎上邊最大數條件的數字只有一個。
輸入
選擇任務: 所有任務
任務名稱 語言 提交時間 狀態 得分
max3 py 2024-04-16 22:02:39 Complete 0
max3 py Complete 100
max3 py Complete 100
max3 cpp 2024-04-19 09:10:23 Complete 100
視窗 3
終止程式
運行狀態:
視窗 4
點擊粉藍色分隔線並左右拖拉
可改變左右半部視窗分配比例,
此操作上下兩組視窗是各自獨立操作的
點擊粉紅色分隔線並上下拖拉
可改變上下半部視窗分配比例

這是比賽系統的主畫面。主畫面的頂部是系統功能清單及當前任務資料。其下則是比賽系統的主介面。

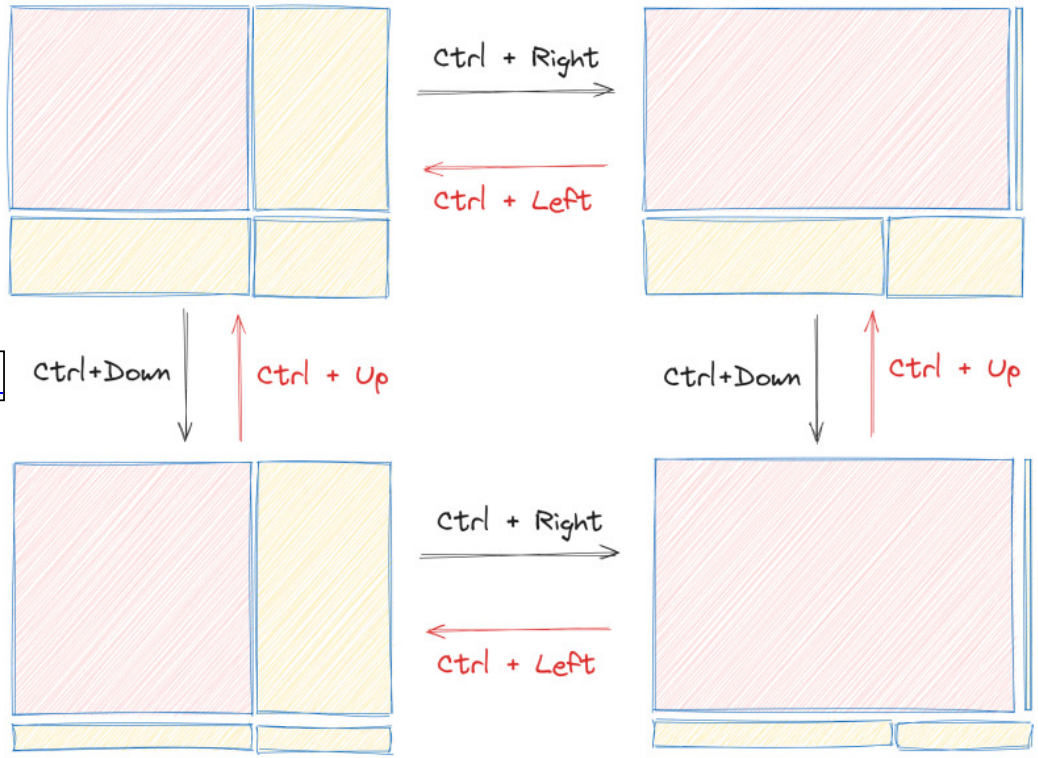
主介面與畫面中分了4 個視窗, 視窗 1 至 4 分別為: 程式編輯視窗, 題目顯示視窗, 程式遞交狀態視窗及程式執行視窗。 在以下的介紹中, 會為大家介紹每一個視窗的功能。

快速編程視窗大小切換

在編程過程中，有時需要把左上方的編程視窗放到最大，以方便進行編程工作。 這時可以使用以下快捷鍵來快速切換編程視窗的大小。

- Ctrl+RightArrow: 將編程視窗的闊度擴至最大。
- Ctrl+LeftArrow: 將編程視窗的闊度變回原本的大小。
- Ctrl+UpArrow: 將編程視窗的高度變回原本的大小。
- Ctrl+DownArrow: 將編程視窗的高度擴至最大。

▲返回求助目錄▲



系統功能菜單

- 檔案
  - **新增**: 對當前的任務新增檔案, 檔案的種類只可能是以下三種: .cpp, .py 及 .txt。 前兩種是程式檔, 最後一種是作為儲存輸入資料的檔案。所有檔案名稱會默認為當前任務的名稱, 但參賽者可以自由更換為其他名稱。名稱的選擇不會影響任務的提交。 檔案的儲存區域, 是以任務來劃分, 所以在解決其中一個任務時, 所有新增的檔案均屬於該任務的且會被存放在相對的區域。

    - **存檔目前檔案**: 把目前編輯中的檔案存入系統中。 系統有部份功能會自動將編輯中的檔案存檔, 例如進行編譯程式時, 系統會先行把編輯中的程式存檔再進行編譯。
    - **重載當前檔案**: 這個功能可以讓使用者放棄目前編輯中的改動, 而重新載入未編輯前的版本。 但要注意的是，系統只會保持一個最近存檔的版本, 所以本功能的用途實際上很有限。 另外一個恢復以前版本的方法, 是在視窗 3 中下載之前已提交的版本, 重新再進行編輯。
    - ***重載目前目錄***: 所有就當前任務而增加的檔案, 其名稱都會出現在這裏。你可以點選其中一個, 以切換到該檔案進行編輯。
- 任務
    - ***任務清單***: 這裡會別出該次比賽中所有的任務, 你可以點選其中一個任務名稱以更換當前工作的任務。
    - **提交**: 提交當前顯示中的任務及檔案, 所提交的檔案為視窗 1 正在顯示的那個檔案, 且該檔案必定需要是一個程式檔案 (.cpp 或 .py)。 MOI-C 組在比賽過程式亦可以編寫及運行 Python 程式, 但只能提交 C++ 檔案。
- 程式
    - **編譯**: 編譯目前在編輯視窗中的程式檔案。即是說, 要進行編譯前要先將程式檔案在編輯視窗中開啟著。若是 Python 程式的說, 則不需要編譯。
    - **選擇預設輸入**: 在目前處理中的任務檔案目錄內, 選擇一個檔案作為程式運行時的預設自動輸入。
    - **執行(互動式)**: 進行互動式運行, 即在運行中所有的輸入是由使用者直接通過鍵盤輸入。
    - **執行(檔案輸入式)**: 程式運行中所需要的輸入資料, 會直接由預設檔案中讀取, 並不需要使用者的介入。
    - **除錯**: 啟動除錯視窗, 詳見下面描述。
    - **自動樣例測**: 給使用者可以對每個輸入輸出的樣例自動進行測試並比對你的輸出與樣例輸出的差異。
- 救助

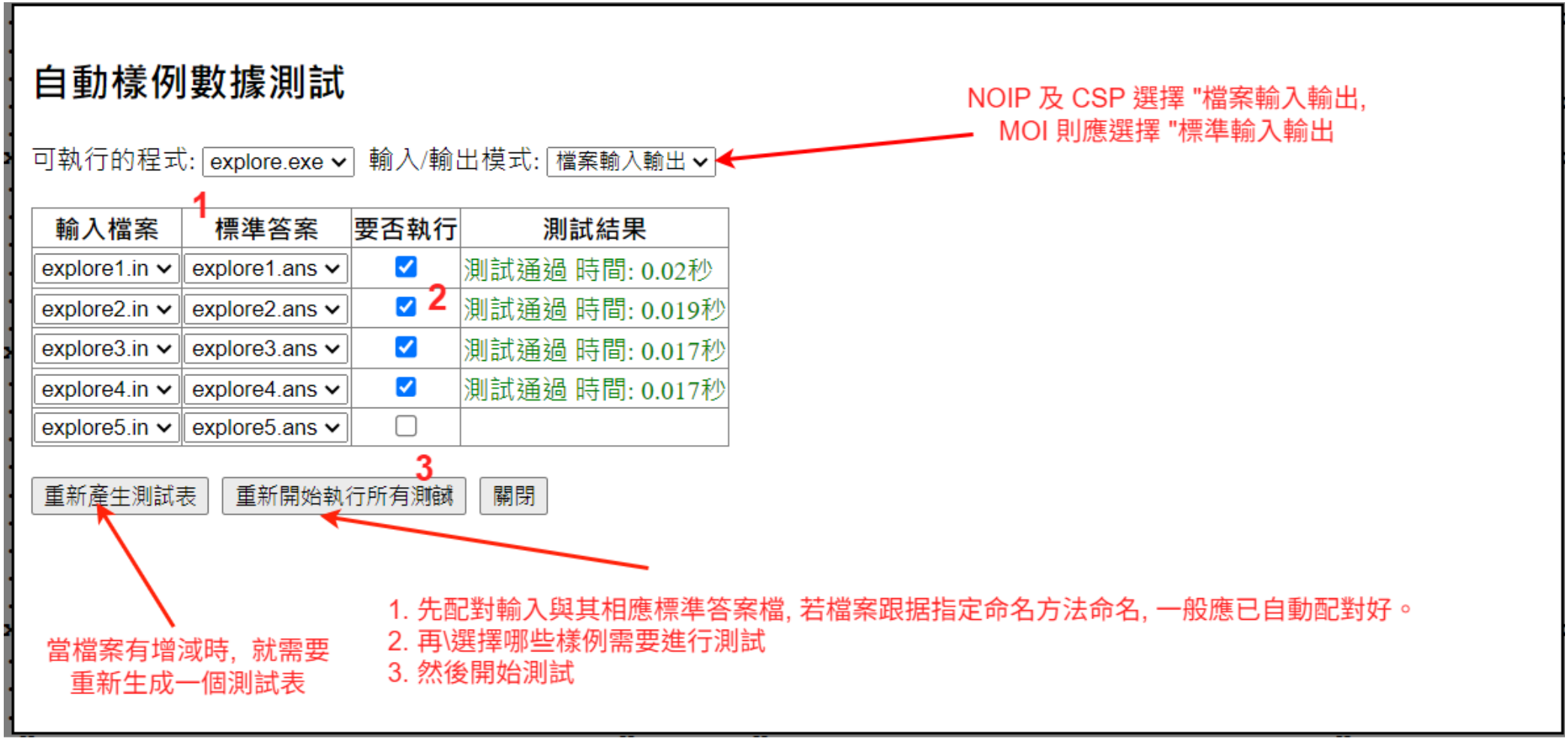
▲返回求助目錄▲

離開功能

比賽是設計在一個封閉的系統內進行, 所以非必要時不能離開本系統。 離開功能的設置是以防系統出現未能預見的問題時, 可以重新啟動系統。 所以離開或者重新啟動系統需要按照一個特定的程序進行。 這個程序基本上是需要有監考人員在場監督, 防止參賽者在重啟過程中進入其他系統。

因此, 若要重啟系統, 務必要確保有監考人員在場監督及記錄整個過程, 才可進行, 否則可能因觸犯比賽規則而被取消比賽資格及成績。

自動樣例測試功能



系統設有一個自動樣例測試功能，可以用以輸入樣例中的數據，並於運行後把程式的輸出和標準樣例答案作比較。 若要使用這個功能，就先要把輸入檔案及標準答案準備好。其中:

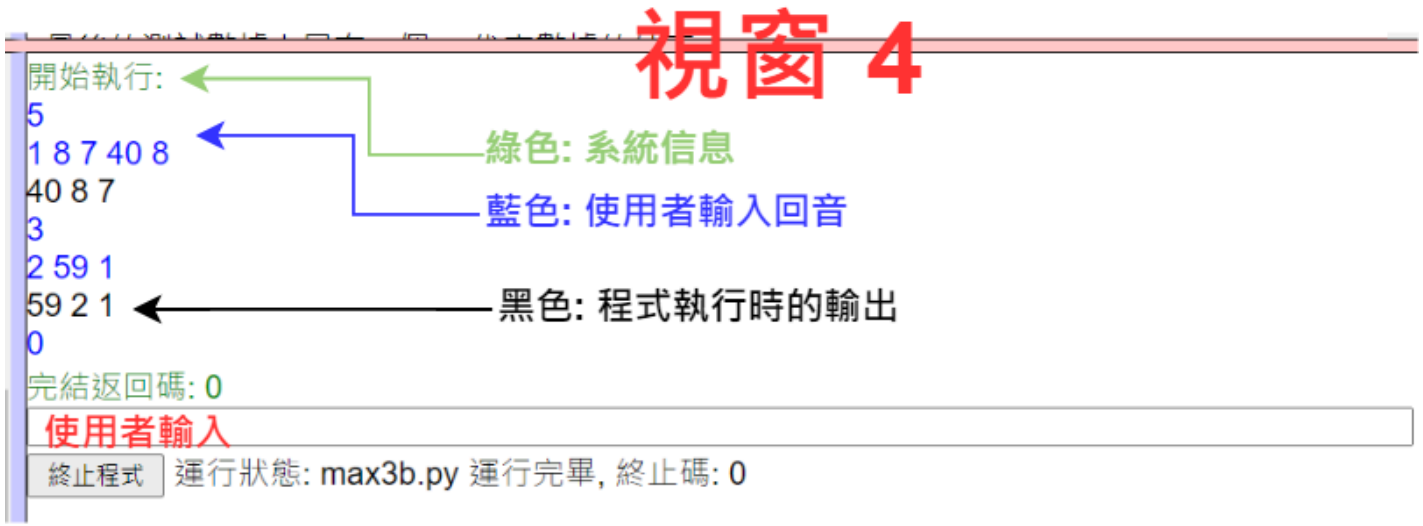
- 輸入檔案的名稱必為 taskname1.in, taskname2.in, taskname3.in, ...
- 相應的標準答案檔案的名稱必為 taskname1.ans, taskname2.ans, taskname3.ans, ...

這裡要注意的是:

- noip 類比賽這些檔案系統內自動已生成。moi 類比賽則需要選手自行輸入相關數據

- 系統在測試程式時，會自動把相應的輸入檔案抄到 taskname.in 檔案中。因此，對於 noip 類比賽，學生不必改變程式中開始輸入及輸出的代碼部份。
- 由於輸出會輸出到 taskname.out 檔案內，所以若果答案有錯時，可以打開這個檔案看看內容。打開檔案後，記得要重載檔案內容一次。
- 程式的輸出及標準答案的比對是使用簡單的逐行內容比對，因此要求每行輸出中數據之間的空格和標準答案必須一至。

程式運行介紹



程式的執行, 無論是通過檔案輸入，或是通過互動輸入, 其運行時的輸出及輸入, 均是在視窗三顯示。 在視窗 3 的下邊有一個輸入方框, 是程式運行用作輸入所需輸入數據的地方。 當然, 如果你是用"檔案輸入模式"執行的話, 則所有輸入數據均會自動由預設檔案中讀取, 不必人手輸入。

視窗中有一個 "終止程式" 的按鈕, 若在程式運行的途中希望重新執行, 步驟上最好先行 "終止程式", 再按 F10 鍵重新執行該程式。

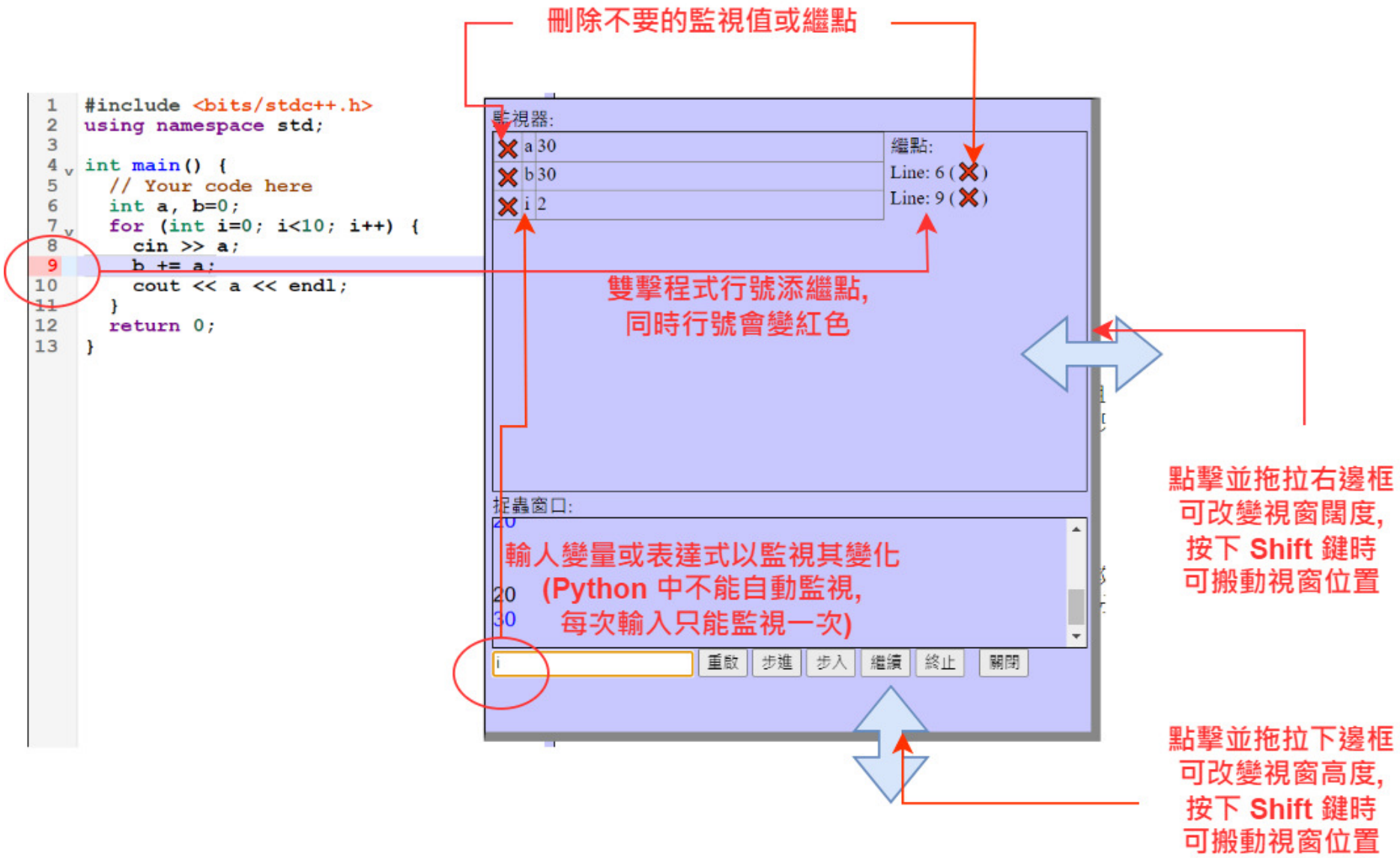
C++ 程式編譯時, 其編譯結果也是在這個視窗內顯示出來:

```
開始執行:
完結返回碼: 0
```

若編譯程式的返回碼為 0, 則代表編譯成功。否則視窗內應有其他一些信息顯示出來。

▲返回求助目錄▲

除錯操作介紹



除錯器在 C++ 及 Python 程式中略有不同的操作方式。除錯 C++ 是通過 GDB 進行, 而除錯 Python 則是通過 Pdb 進行。

除上圖顯示的操作外, 以下是其下端按鈕所代表的操作:

- **重啟開始**: 重新由頭開始程式運行。(在 Python 中，基本上不需要重啟, 程式運行到最後時，她會自動重新再開始。)
- **下一步**: 執行下一行指令, 這裡函數呼叫會當為單一行指令對待。在執行完下一步之後系統會重新等待使用者的指令
- **繼續運行**: 繼續執行以下指令直至遇到下一個斷點為止或程式結束為止。
- **終止**: 終止目前除錯中的程式, 當除錯程式在運行時, 你是不可重新編譯該程式的, 因為程式的可執行檔是被鎖起了, 所以這時編譯會出現錯誤。但若你先終止了程式運行, 則可進行編譯。
- **離開**: 關閉除錯視窗

針對 C++ 的除錯功能

除錯視窗中有一個用以輸入 "變量或表達式" 的輸入方塊, 你可在其內輸入變量的名稱或一些表達式, 除錯系統將於每一次暫停時將你輸入的變量或表達式的值顯示出來。 若你只想顯示這些內容一次, 你可在輸入的表達式的前面加一個等號 "=", 這樣你所輸入的變量或表達式的值就只會顯示一次。

除錯系統是透過 GDB 進行, 所以表達式可以運用一些 GDB 特定方式輸入。例如, 針對矩陣及向量而言, 以下是一些特別表達式的使用及其意思。

數据類型	表達式	意思
`int A[200];`	`A[0]@10`	除列 A 中, 由索引 0 開始顯示 10 個元素的內容
`vector<int> V;`	`*(&V[0])@10`	在向量 V 中, 由索引 0 開始, 顯示 10 個元素的內容

另外還有一種特別的除錯方法, 就是在程式中設定一個特別的函數, 用以顯示某些數據結構的內容。 但這個函數從來不會直接被程式仲任何一部分呼叫。 它的作用是在除錯器中有必要時透過表達式去呼叫它。 這樣做法的好處是你提該程式時, 不必修改程式的任何一個部份, 且同時在除錯的時又可隨心所欲地使用這些函數來監視一些複雜的數据結構內容。

▲返回求助目錄▲

C++ 注意事項

輸出字元

在本系統內運行的程式, 每次運行的總輸出不能超過 1M 字元。這是為了保護本系統的正常運行而設置的。 在正常的情況下, 題目所要求的輸出的字元數是不會太多的。 這限制主要受影響的會是那些通過正常輸出除錯的程式。因此, 若要輸出資料進行除錯的話, 請盡量減少不必要的輸出。

沒有使用編譯器優化參數

今年我們會試行編譯器沒優化, 即是在程式編譯時不會加入 -O2 這個參數, 亦不容許在程式內使用任何 #pragma 的指令。

在這個情形下, 你在編程過程中任何的優化, 很有可能會改進程式的執行速度。

-D_MOI 編譯參數

在本系統中使用了 -D_MOI 這個編譯參數，你可以用它來識別程式是否在本地機器運行因而運行不同部份(如用以除錯的部份)的程式。以下是一個可能的例子， 它可以用來取代系統中的自動檔案輸入的功能。特點是在除錯時也可以用到。

```
int main() {
  #ifdef _MOI
    freopen("input.txt", "r", stdin);
  #endif
  ...
```

這個用法的好處是程式可以直接提交而不用在提交前刪去除錯用的部份指令。

輸出指令除錯

已知 GDB 在除錯時, 若嘗試 "步進" 或 "步入" 輸出指令 cout << 時很可能出現問題, 所以盡量不要進入這些指令。 簡單的做法可以在 cout << 指令後設置繼點, 這就可以跳過輸出指令而停在其後方。

若你在除錯時遇到這個問題, 你可以把除錯視窗關閉後重新進入除錯系統。

▲返回求助目錄▲

NOIP 類競賽模式

NOIP (及類似項目，如 CSP-SJ 第二輪等) 有特定的源代碼收集及檔案輸入輸出的要求。 本系統內亦相應有特定的勁能是對應這類比賽而設的。主要功能包括以下:

- 登入時參加者可以指定檔案提供及收集的伺服器的 URL。這是對應在 NOIP 所用的伺服器一般設在內聯網上的不同電腦上
- NOIP 輸入輸出到目前為止還是使要通過檔案進行，本版本提供檔案自動對應測試及自動改名功能。這使參賽者在不用更改程式內的輸入輸出檔案的名稱下對各組題目提供的測試數據進行測試及比對結果
- NOIP 在比賽過程中，參加者所供的源代碼不會即場進行評分測。

▲返回求助目錄▲

快捷鍵

本系統的大部分功能, 都可以直接通過快捷鍵來執行。快捷鍵可使工作更加有效率。以下是系統提供的快捷鍵的清單。 本系統的大部分功能都可以直接通過快捷鍵調用快捷鍵可使工作更加由效率以下是系統用到的快捷鍵的清單:

檔案	Ctrl+N	新增
	Ctrl+S	存檔目前檔案
	Ctrl+L	重載當前檔案
	Alt+1 ... Alt+0 (或 Ctrl+1 ... Ctrl+0)	調用檔案 1 至 10
任務	Alt+F1 ... Alt+Fn (或 Ctrl+F1 ... Ctrl+Fn)	開啟任務 1 至 n
程式	F9	編輯
	F10	執行(互動式)
	Shift+F10 (或 Ctrl+F10)	執行(檔案輸入式)
	F12	進入除錯模式
	Shift+F12	進入自樣例測試模式
除錯功能	F10	下一步

▲返回求助目錄▲

系統重載注意事項

因比賽電腦出現問而需要重載系統時, 請注意以下事項:

- 重載或重啟過程必須要在有監考人員的指導下進行
- 重載或重啟後, 若之前以編輯的檔案未有出現在工作區內, 請點選"檔案"->"重載目前目錄"

▲返回求助目錄▲

版本升級記錄

2.250301

- 曾強防止切換其他程式功能
- 改進部份使用者介面

2.241111

- 新增了 NOIP 類賽事支援
- 新增自動樣例評測功能
- 改良部份除錯模式的功能及顯示

1.xxxx

- 新增預設程式設定, 若當前在視窗開啟著的不是一個程式的話, 編譯或執行時就會嘗試使預設的程式作為目標
- 增加了 `-D_MOI` 編譯參數。這個參數可以讓你在編譯時識別是否在本地電腦運行因而加入一些除錯的指令

1.240426

- 部份快捷鍵增加 Ctrl 的選項, 盡可能統一所有快捷鍵
- 確最後10分鐘可以多次提交任務

0.240422

- 全善求助檔案內容
- 解決 Python 無法進行自動輸入輸出運行的問題
- 增加編輯檔案切換時的顯示動畫
- 增加已提交任務的狀態改變提示動畫

0.240419b

- 完善更新版本時的顯示功能
- 增加 "進階功能" 菜單及其內的兩項功能: "自動輸入輸出運行" 及 "檔案比對"
- 新設每個任務最多可以打開 10 個檔案的限制, 並增設快捷鍵用以直接更換目前檔案。

▲返回求助目錄▲