程式judge系統

1. 研究動機:

愛沙尼亞，全球第一個國家將程式設計融入國小一年級課程，在 2012 年 9 月推 動虎躍計畫，於次年將計畫升格為義務教育；2014 年 9 月，英國將程式設計納入中小 學課程。科技快速發展，從網頁、大數據到物聯網，不少國家針對程式設計課程的進 度急起直追，國外許多教師利用 Scratch 軟體進行課程，台灣也逐漸帶入積木程式等 相關軟體使學生與世界接軌。

比起以往的考試及評量，程式是一門綜合性的教育，不單再使用記憶性學法，更 多的是自主學習，面對各個程式專案，設計者面臨的是如何解決問題及邏輯思考，學 習的過程中不但增加個人價值，也可以增加產業導向。根據 104 求職網資訊科技統計， 企業方面程式人力需求方面供不應求，從無形的電子通訊到有形的電子設備，資訊與 我們形影不離，若國家資訊技術發展更加完備，不論是報稅或支付等，對政府及中小 型企業也會帶來帶來高度的競爭力，因此以往個課程對程式解題的要求已經無法因應社會發展，所以我們決定開發一款新的程式解題系統來訓練學生。

1. 研究目的：

在未來軟體研究人員不應該只會寫死板的程式依題目的敘述給出答案，而應該往多樣性發展，甚至能幫別的工程師debug找出別人程式裡面的bug，而要訓練這種能力一般的題型已經無法達到因此我們使用以下題型來完成目的

(1) normal，傳統的解題類型

(2) hack，給使用者一個錯誤的程式碼請使用者找出輸入甚麼測資會使他輸出錯誤訊息

(3) debug，給使用者一個錯誤的程式碼請他修正，並且不能更改到整體的架構

(4) special，給使用者一些無特定答案的題目或是跟使用者進行互動

並且在配合嚴格比對(程式輸出必須一模一樣)和非嚴格比對(程式輸出可以有多餘的空白換行或是tab)來達到檢測程式正確性。

1. 研究及器材：

一:硬體

一台cjscope gt550的筆電與其他周邊硬體

二:軟體

Codeblocks ,visture studio,gcc,g++,bash,vim

1. 研究過程與方法：

以基本程式解題系統為架構，開發出可以進行編譯與執行功能的系統，並且在以std標準資料庫內建的getline做出嚴格比對，與輸入string進行非嚴格比對來完成normal模式的運行。

Hack模式則是將正確答案與題目程式碼做成雙重normal模式執行，並且在判斷使用者輸入的側資範圍後進行執行程式並且比對雙方的輸出。

Debug則是以normal為繼承做檢測，但在確認使用者無更改程式基本架構(編輯數量少於一定數字)時需要使用LCS演算法找出最長遞增子序列來完成。

Special模式則是需要出題者寫出各自的special judge而本系統將使用者的輸出輸出到special judge並由回傳值判斷結果

伍、研究結果:

經過一些學生測試後我們發現大部分寫過程式的人幾乎都只有寫過normal模式的題目，而debug和hack類型的磨練也只能從在normal模式下wrong answer後進行debug時訓練，因此在上面兩種題目大部分卡關的人較多，而special因為題型特殊的關係會比較類似開方式解答的normal因此答對人數會比normal還多。

而在測試完後大部分的學生都同意這類型的題目確實比一般的模式好玩，可以促進學習程式的動力。而在我們的judge下也可以變成一種互動的模式，由A方負責寫題目然後B方進行答題在由C方進行hack並且在丟回B方進行debug最後再由D方寫一個special judge判斷是否能完成A方出的題目要求，如此一來既可脫離大家對寫程式是個人活動的認知達成交流的目的。

1. 討論:

目前這種多功能的judge並不是由我們首創，但我們也是包含最多類型的judge，大部分的judge像是codeforces或是tpoj之類的都是只有基本的special judge雖然出題者可以以此來做到大部分的功能，但這會非常考驗出題者的實力和精力，因此大部分就算有special judge的系統也不會因此而擁有這麼多類型的題目 。

而我們的Judge除了以上所說的優點外我各人覺得他最大的發展機會是他以c++所寫的因此每個拿到他的人都可以將自己的巧思加進去並且發展出另一套集聚個人特色的judge系統並且能繼承之前所寫下的函式發展出新的題目類型。

柒、結論:

哵、參考資料:

1.Renewable Energy UK(n.d).Solar Declination,

from：<http://www.reuk.co.uk/wordpress/solar/solar-declination/>