

# 國立新竹女子高級中學彈性學習時間學生自主學習計畫申請書

2021/07/03 12:00

申請人	張敏茹	班級/座號	2 年 13 班 14 號
申請學期	10902	申請時數	31
共學同學	無		
計畫名稱	C++基礎自學		
學習類型	學科課程延伸	對應學科屬性	資訊
設備需求	電腦		
指導教師	古佳怡		
自主學習內容概述	在高二接觸資訊課程後，漸漸地對資訊產生興趣。我會利用這個時間將上課的部分重新複習，並利用線上資源做練習，以達到對程式的熟悉並培養更深的樂趣。		
預期效益	希望藉由這次自主學習的機會可以將平常上課不熟的部分重新了解，並且利用線上平台練習題目，之後更可以自學一些上課沒有接觸過的範圍。		
與十二年國教核心素養之關聯	A1 身心素質與自我精進、A2 系統思考與解決問題、B2 科技資訊與媒體素養		
成果展示	同意於校內學習平台提供自主學習成果與資料給其他同學參考		

週次	日期	課程	自學內容	自學場地
2	110/03/02(二)	自主學習	將課堂的程式語言介紹及變數運算及輸入輸出的簡報重新複習	
2	110/03/04(四)	自主學習	練習課堂相對應程式介紹變數運算及輸入輸出的題目	
3	110/03/09(二)	自主學習	將課堂流程控制(IF)的簡報重新複習	
3	110/03/11(四)	自主學習	練習課堂相對應流程控制(IF)的題目	
4	110/03/16(二)	自主學習	將課堂迴圈(for)的簡報重新複習	
4	110/03/18(四)	自主學習	練習課堂相對應迴圈(for)的題目	
5	110/03/23(二)	自主學習	將課堂迴圈(while)的簡報重新複習	
6	110/03/30(二)	自主學習	練習課堂相對應迴圈(while)的題目	
6	110/04/01(四)	自主學習	訂正並檢討筆試的題目(15題)	
7	110/04/06(二)	自主學習	訂正並檢討筆試的題目(15題)	
7	110/04/08(四)	自主學習	訂正並檢討筆試的題目(15題)	
8	110/04/13(二)	自主學習	訂正並檢討筆試的題目(15題)	

8	110/04/15(四)	自主學習	訂正並檢討筆試的題目(15題)	
9	110/04/20(二)	自主學習	訂正並檢討筆試的題目(15題)	
9	110/04/22(四)	自主學習	重新挑戰期末考的程式實作題目(分類帽)	
10	110/04/27(二)	自主學習	重新挑戰期末考的程式實作題目(古靈閣銀行)	
10	110/04/29(四)	自主學習	重新挑戰期末考的程式實作題目(鳳凰會結社)	
11	110/05/04(二)	自主學習	重新挑戰期末考的程式實作題目(禮物選擇遊戲)	
11	110/05/06(四)	自主學習	重新挑戰期末考的程式實作題目(貓貓攻擊遊戲)	
13	110/05/18(二)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
13	110/05/20(四)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
14	110/05/25(二)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
14	110/05/27(四)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
15	110/06/01(二)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
15	110/06/03(四)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
16	110/06/08(二)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
16	110/06/10(四)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
17	110/06/15(二)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
17	110/06/17(四)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
18	110/06/22(二)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	
18	110/06/24(四)	自主學習	在 Green Judge 或 Zero Judge 練習 1-5 個題目	

以下為審查填寫欄，申請者勿填。

審查	<input checked="" type="checkbox"/> 通過 <input type="checkbox"/> 待修正 <input type="checkbox"/> 不通過
	審查意見：

	認證： <span style="border: 1px solid red; padding: 2px;">古佳怡老師</span>
家長簽名	
學校核章	<span style="border: 1px solid red; padding: 5px; display: inline-block;">國立新竹女子高級中學 自主學習小組</span>

# 國立新竹女中彈性學習時間學生自主學習成果表

2021/07/03 11:30

申請人	張敏茹	班級/座號	2 年 13 班 14 號
申請學期	10902	申請時數	31
共學同學	無		
計畫名稱	C++基礎自學		
學習類型	學科課程延伸	對應學科屬性	資訊
設備需求	電腦		
指導教師	古佳怡		
自主學習內容概述	在高二接觸資訊課程後，漸漸地對資訊產生興趣。我會利用這個時間將上課的部分重新複習，並利用線上資源做練習，以達到對程式的熟悉並培養更深的樂趣。		
預期效益	希望藉由這次自主學習的機會可以將平常上課不熟的部分重新了解，並且利用線上平台練習題目，之後更可以自學一些上課沒有接觸過的範圍。		
與十二年國教核心素養之關聯	A1 身心素質與自我精進、A2 系統思考與解決問題、B2 科技資訊與媒體素養		
成果展示	同意於校內學習平台提供自主學習成果與資料給其他同學參考		

週次	日期	自學內容	檢核進度	學習心得	自學場地
2	110/03/02(二)	將課堂的程式語言介紹及變數運算及輸入輸出的簡報重新複習	完全達標	在這當中我了解了關於一直都不太懂得小數與整數，例如哪種除以哪種會有小數點之類的。	
2	110/03/04(四)	練習課堂相對應程式介紹變數運算及輸入輸出的題目	完全達標	在這節我去做關於 Int 和 double 的程式，經過不停的變換組合去更加了解他們之間的關係。 【作品連結網址】 <a href="https://ldrv.ms/w/s!Anpp-cxFty_lgZxigK-1-DJz0aE0jQ?e=8ybnCh">https://ldrv.ms/w/s!Anpp-cxFty_lgZxigK-1-DJz0aE0jQ?e=8ybnCh</a>	
3	110/03/09(二)	將課堂流程控制(IF)的簡報重新複習	完全達標	看完了簡報理解了 if else 的篩網模式和 if 跟 if 之間的獨立性。也看了很多老師的範例。	
3	110/03/11(四)	練習課堂相對應流程控制(IF)的題目	完全達標	寫了兩個程式碼，一個是課堂的隨堂	

				<p>練習(hgsh online)，一個是 green judge 上的題目(a011)</p> <p>【作品連結網址】  <a href="https://1drv.ms/w/s!Anpp-cxFty_lgZxh27I-4pneQ1jr7A?e=khA801">https://1drv.ms/w/s!Anpp-cxFty_lgZxh27I-4pneQ1jr7A?e=khA801</a></p>	
4	110/03/16(二)	將課堂迴圈(for)的簡報重新複習	稍有落後	<p>我發現 for 的概念相較於之間更為複雜，而我無法在這堂課完全地將整個簡報看完了解</p>	
4	110/03/18(四)	練習課堂相對應迴圈(for)的題目	稍有落後	<p>for 的複雜性增高，而因為如此花了其他時間把前面的簡報看完並練習了其他 4 個題目，1 個隨堂練習(雙 11)以及 3 個 green judge 上的題目(a026、a027、a029)。</p> <p>【作品連結網址】  <a href="https://1drv.ms/w/s!Anpp-cxFty_lgZxfGEi5tuxvo_In2Q?e=wnlou3">https://1drv.ms/w/s!Anpp-cxFty_lgZxfGEi5tuxvo_In2Q?e=wnlou3</a></p>	
5	110/03/23(二)	將課堂迴圈(while)的簡報重新複習	稍有落後	<p>while 迴圈也較為複雜，花了很多的時間了解，並無法在一節課內看完整份 PPT。因段考的關係，自主學習時間也被壓縮，因此進度稍微落後。</p>	
6	110/03/30(二)	練習課堂相對應迴圈(while)的題目	完全達標	<p>這次都是重新練習隨堂練習的題目，共有兩題(猜拳、期間限定合作)。其中猜拳讓我比較難理解，之中也包含較多概念。</p>	

				<p>【作品連結網址】</p> <p><a href="https://ldrv.ms/w/s!Anpp-cxFty_lgZxkfx9iTTl_gjkILg?e=2nxfJe">https://ldrv.ms/w/s!Anpp-cxFty_lgZxkfx9iTTl_gjkILg?e=2nxfJe</a></p>	
6	110/04/01(四)	訂正並檢討筆試的題目(15題)	完全達標	<p>今天是1-4題，發現打好基礎理解所有的PPT之後作答變得輕鬆許多。這3題的題目都是屬於變數運算及輸出的範疇。</p>	
7	110/04/06(二)	訂正並檢討筆試的題目(15題)	完全達標	<p>今天是5-6題，而這些都是if-else的題目，了解他們的性質後，題目也變得較簡單。</p>	
7	110/04/08(四)	訂正並檢討筆試的題目(15題)	完全達標	<p>今天是7-9，題目開始變得有陷阱且複雜，概念橫跨if-else到for。8和9是我上學期寫考卷時錯的題目。</p>	
8	110/04/13(二)	訂正並檢討筆試的題目(15題)	完全達標	<p>今天是10-11，題目都屬於迴而出現了之前都沒看過的迴圈內迴圈。內迴圈。</p>	
8	110/04/15(四)	訂正並檢討筆試的題目(15題)	稍有落後	<p>今天是12-14，12題的迴圈讓我思考較久，因為有迴圈內有雙迴圈，還結合圖形。13、14是題組，而觀念較直觀，比12好解許多。</p>	
9	110/04/20(二)	訂正並檢討筆試的題目(15題)	完全達標	<p>今天是最後一題，將數字依序帶入最後才發現程式碼有目的，讓我思考若可以提前知道他的目的解題就會快許多。回顧一整個訂</p>	

				正考卷的過程，真的讓我更了解整學期的課程。	
9	110/04/22(四)	重新挑戰期末考的程式實作題目(分類帽)	完全達標	<p>發現徹頭徹尾的把所有東西複習一遍真的很有效。一來，我的錯誤率減低，而無法執行時總能自己找出問題。二來，看到題目比較有想法，不會不知道怎麼下手。這題本來就屬於比較簡單的範圍，令我得心應手。</p> <p>【作品連結網址】  <a href="https://ldrv.ms/w/s!Anpp-cxFty_lgZxmHBu2bruirY2Cmg?e=Cw0mJn">https://ldrv.ms/w/s!Anpp-cxFty_lgZxmHBu2bruirY2Cmg?e=Cw0mJn</a></p>	
10	110/04/27(二)	重新挑戰期末考的程式實作題目(古靈閣銀行)	完全達標	<p>這題在我當初考試的時候解的時候覺得很難，主要是關於幣值轉換我無法理解。但是可能練習的過程也有訓練到邏輯，讓我不費力的解完這題。</p> <p>【作品連結網址】  <a href="https://ldrv.ms/w/s!Anpp-cxFty_lgZxo6zgpca00RXuIGw?e=ivFXdg">https://ldrv.ms/w/s!Anpp-cxFty_lgZxo6zgpca00RXuIGw?e=ivFXdg</a></p>	
10	110/04/29(四)	重新挑戰期末考的程式實作題目(鳳凰會結社)	完全達標	<p>這種題型對我來說比較得心應手，因為之前的PPT也有出現相似的題型，只是題幹比較複雜。在反覆的練習我學到怎麼分析複雜的題型並用程式的語言將他表現出</p>	

				來。 【作品連結網址】 <a href="https://1drv.ms/w/s!Anpp-cxFty_1gZxqvkb61an-E4Z0rQ?e=iF8We3">https://1drv.ms/w/s!Anpp-cxFty_1gZxqvkb61an-E4Z0rQ?e=iF8We3</a>	
11	110/05/04(二)	重新挑戰期末考的程式實作題目(禮物選擇遊戲)	稍有落後	這題我想了很久，還是忍不住用高二下程式課程學的陣列，實在想不出不用陣列要怎麼寫。看完老師的解答後有豁然開朗之感，恍然大悟說程式以分層的方式可以解決很多問題。 【作品連結網址】 <a href="https://1drv.ms/w/s!Anpp-cxFty_1gZxuHmPrKvSrS_PqpA?e=kJQCWo">https://1drv.ms/w/s!Anpp-cxFty_1gZxuHmPrKvSrS_PqpA?e=kJQCWo</a>	
11	110/05/06(四)	重新挑戰期末考的程式實作題目(貓貓攻擊遊戲)	完全達標	這題也很難，且靠近期中考進度落後。看完老師的解答依然發現若用分層的方式(先用 If else 篩選)寫就很簡單。我還是要學習如何跳出自己思維的框架，不能一直卡在一個思維出不去，這樣反而沒辦法想到答案。 【作品連結網址】 <a href="https://1drv.ms/w/s!Anpp-cxFty_1gZxsRS-WpsgI8yzdXw?e=ztvMHw">https://1drv.ms/w/s!Anpp-cxFty_1gZxsRS-WpsgI8yzdXw?e=ztvMHw</a>	
13	110/05/18(二)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	這邊改掉原先的計畫，去籌備自己設計第一個程式，這樣比較能呈現自己真的是否學會。(因	

				<p>改掉計畫，所以進度都設無)</p> <p>第一步為尋找可行的題材。在偶然翻到生物課本的轉錄轉譯，我就決定以它為我撰寫的程式。</p>	
13	110/05/20(四)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	<p>目標是使用者給一串 DNA 密碼，而之後會轉錄轉譯最後成為蛋白質序列，為此我準備了架構圖。</p> <p>【作品連結網址】  <a href="https://ldrv.ms/w/s!Anpp-cxFty_1gZxTE25fVqdsjhTgPw?e=pbDA75">https://ldrv.ms/w/s!Anpp-cxFty_1gZxTE25fVqdsjhTgPw?e=pbDA75</a></p>	
14	110/05/25(二)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	<p>開始撰寫程式，真的比想像中困難，一次就 200 多行程式，而迴圈與條件式之間的邏輯也要很清晰。每次跑程式的時候都有 bug。除了這節課我還花很多時間在上面，並開啟了另一個程式視窗，是負責釐清我欠缺的觀念。</p>	
14	110/05/27(四)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	<p>依然是修程式 debug，主要的架構已經出來了，是以 while 包圍了大部分的程式碼。我也對整體做了一大修改，著重於整個邏輯層面，盡可能想讓這</p>	

				<p>個程式碼貼近原始的轉錄轉譯，也在過程中加入之前沒注意到的問題。</p> <p>【作品連結網址】  <a href="https://1drv.ms/w/s!Anpp-cxFty_1gZxQV52FUiTjB5kg0A?e=2aHxHb">https://1drv.ms/w/s!Anpp-cxFty_1gZxQV52FUiTjB5kg0A?e=2aHxHb</a></p>	
15	110/06/01(二)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	<p>整個程式碼可以運作了，而我在這堂課試圖找到更多可以加強的地方。我發現整個程式碼是不需要 while 包圍了，反而幾個條件式就可以取代之，而這堂課我也著手於簡化我原先的程式碼。</p> <p>【作品連結網址】  <a href="https://1drv.ms/w/s!Anpp-cxFty_1gZxyWmSZiXL_xGsTZg?e=i44aj3">https://1drv.ms/w/s!Anpp-cxFty_1gZxyWmSZiXL_xGsTZg?e=i44aj3</a></p>	
15	110/06/03(四)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	<p>把我的程式碼配合 word 報告完整的表現出來，包括整個轉錄轉譯的基本概念、輸入輸出的指示、程式碼，寫成一個報告。</p> <p>【作品連結網址】  <a href="https://1drv.ms/w/s!Anpp-cxFty_1gZwZx0goOfBuD9XZeg?e=nD128B">https://1drv.ms/w/s!Anpp-cxFty_1gZwZx0goOfBuD9XZeg?e=nD128B</a></p>	
16	110/06/08(二)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	<p>這天開始設計自己的第二個程式，一樣在找尋設計的題材。一樣是往生物的方向發展，鎖定在旁氏表以及疾病</p>	

				機率計算。	
16	110/06/10(四)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	初步設計草稿，查詢我要的疾病(亨丁頓舞蹈症、鐮刀型紅血球疾病)的特徵，包括他們是不是遺傳疾病、是顯性還是隱性、病狀等	
17	110/06/15(二)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	大概釐清會需要用到二維陣列以及很多的條件式，就開始我的程式撰寫(未完成)。	
17	110/06/17(四)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	繼續我的程式撰寫，我不只想要幫使用者預測小孩的得病機率，還想要呈現出旁氏表，讓他們知道狀況。完成了我的程式撰寫，至少可以執行。	
18	110/06/22(二)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	修整了我的程式問題，像是旁氏表呈現的方式可能會是小寫在前面(但是一般是大寫要在前面)，再盡可能把程式精簡化 【作品連結網址】 <a href="https://1drv.ms/w/s!Anpp-cxFty_1gZx095uWKDBkhNk6uA?e=UrH15U">https://1drv.ms/w/s!Anpp-cxFty_1gZx095uWKDBkhNk6uA?e=UrH15U</a>	
18	110/06/24(四)	在 Green Judge 或 Zero Judge 練習 1-5 個題目	無	把整個程式配合 word 檔呈現出來，包括輸入輸出和病狀的介紹。 【作品連結網址】 <a href="https://1drv.ms/w/s!Anpp-cxFty_1gZwnON10fFukTLbq1Q?">https://1drv.ms/w/s!Anpp-cxFty_1gZwnON10fFukTLbq1Q?</a>	

## 成果說明：文字

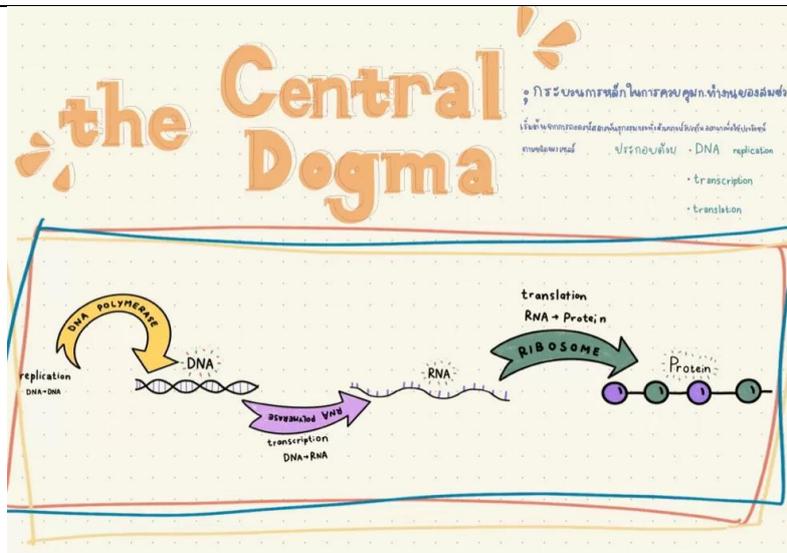
目標是將高二上的程式課程完整的做一次複習及加深加廣的練習。由於高二上初期還未進入狀況，程式的疑問一再積累，導致到後期完全跟不上進度。藉由這次自主學習，我將以學好程式打好基礎為目標，重新複習上學期老師的 PPT，再以線上的資源如 GreenJudge 作為練習。不只如此，後期還將會重新寫高二上學期的期末考(包括筆試與實作)，作為學習成果的檢核。而最終在學期末，我改了先前的計畫，變為自己設計兩個程式，不但可以在之中學習，也可以當作一個自我學習成果。

## 成果說明：照片

說明：第一個程式撰寫——轉錄與轉譯

## 轉錄與轉譯

### 題目說明

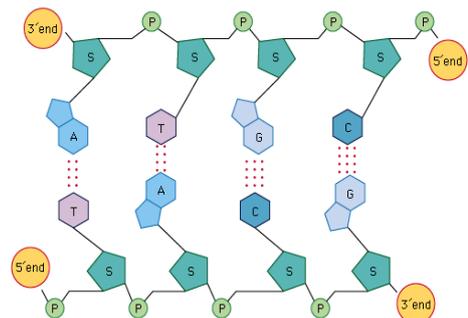


如何讓 DNA 上的鹼基對 A、T、C、G 變成可以表現的蛋白質？分子生物學的中心法則(central dogma of molecular biology)是在 1957 年由英國生物物理學家 Francis Crick 提出：「DNA 會將訊息傳遞給 RNA，而 RNA 再將訊息傳遞給蛋白質。」

### DNA 與 RNA 的方向：

一條 DNA 或 RNA 有分是 5' 端還是 3' 端，何謂 5' 端和 3' 端呢？方向是由五碳糖決定，與含氮鹼基連接的碳為第 1 個碳；與磷酸基連接的碳為第 5 個碳。

若聚核苷酸鏈的一端為第 5 個碳，此端稱為 5' 端；另一端則是五碳糖的第 3 個碳，此端為 3' 端。一般來說如果沒有特別標明的話，一個序列的左邊為 5' 端、右邊為 3' 端。舉例來說，AAACC=5' AAACC 3'。



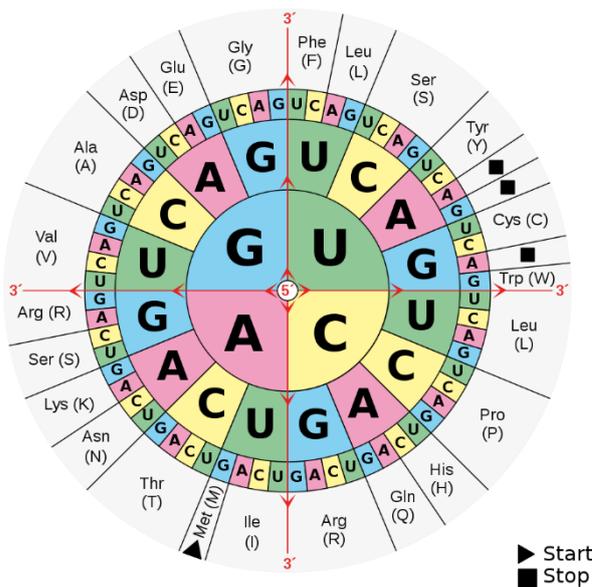
### 轉錄：

以 DNA 的一股為模板，由 RNA 聚合酶催化成 RNA，分為起始、延長、終止三個步驟，而 DNA 模板上的 A、T、C、G 會分別變成 U、A、G、C。此轉錄的過程是由模板股的 3' 到 5' 端，而 mRNA 的合成方向為 5' 端到 3' 端。所以如果一個 DNA 序列從 5' 端到 3' 端為 ACCG，那轉錄後的 mRNA 序列從 5' 端到 3' 端就是 CGGU。mRNA 之後可以帶著這串訊息去核糖體轉譯成蛋白質。

### 轉譯：

以 mRNA 為模板，到粗糙內質網上的核糖體可進行轉譯。轉譯可以讀 mRNA 上的序列並將其轉為特定的胺基酸，而以肽鍵連接的胺基酸可形成多肽鏈(修飾後成為蛋白質)。mRNA 上 3 個核苷酸可構成 1 組密碼子，而每種密碼子可對應到一種胺基酸 (一種胺基酸卻可對應到數種密碼子)，如下表。而每一段基因轉譯都需要啟動子，而 RNA

聚合酶便會結合到啟動子上並開始轉譯。轉譯讀取 mRNA 的方向是從 5' 到 3'，多肽鏈的合成方向為 N 端 (NH<sub>3</sub>)到 C 端(COOH)。每個密碼子序列的開頭都需要有起始密碼子(DNA 序列為 CAT；mRNA 序列為 AUG；胺基酸為 Met)和終止密碼子(DNA 序列為 TTA/CTA/TCA；mRNA 續列為 UAA/UAG/UGA；不連接胺基酸)。但是 mRNA 序列中間若出現起始密碼子或終止密碼子，這段序列就不能合成出完整的多肽鏈。





## 至少兩筆範例測資

### 輸入範例 1

(輸入可轉錄：符號為 DNA 核甘酸序列的符號[A T C G]。)

可轉譯：

- ✓ mRNA 核甘酸的數量為三的倍數、
- ✓ mRNA 序列的開頭有起始密碼子結尾有終止密碼子、
- ✓ 中間無起始密碼子或終止密碼子)

24

```
T C A C A C T G A T T G C A A  
G A C C C C C A T
```

### 輸出範例 1

mRNA: 5' [AUGGGGGUCUUGCAAUCAGUGUGA]3'  
protein: N 端[Met-Gly-Val-Leu-Gln-Ser-Val]C 端

### 輸入範例 2

(輸入可轉錄：符號為 DNA 核甘酸序列的符號[A T C G]。)

不可轉譯：

- × mRNA 核甘酸的數量為三的倍數、
- × mRNA 序列的開頭有起始密碼子結尾有終止密碼子、
- ✓ 中間無起始密碼子或終止密碼子)

16

```
T C A A A A G C G C G A C A G  
A
```

### 輸出範例 2

mRNA: 5' [UCUGUCGCGCUUUUGA]3'  
can't be translated

### 輸入範例 3

(輸入不可轉錄：之中有符號與 DNA 核甘酸序列的符號[A T C G]不相符。)

不可轉譯(但符合轉譯條件)：

- ✓ mRNA 核甘酸的數量為三的倍數、
- ✓ mRNA 序列的開頭有起始密碼子結尾有終止密碼子、
- ✓ 中間無起始密碼子或終止密碼子)

18

```
C T A A A A G A C C G C U A C  
A T A C A T
```

### 輸出範例 3

can't be transcribed  
can't be translated

#### 輸入範例 4

(輸入不可轉錄：之中有符號與 DNA 核甘酸序列的符號[A T C G]不相符。

不可轉譯(也不符合轉譯條件)：

- ✓ mRNA 核甘酸的數量為三的倍數、
- ✓ mRNA 序列的開頭有起始密碼子結尾有終止密碼子、
- × 中間無起始密碼子或終止密碼子)

15

T T A 6 A C C A T A C C C A T

#### 輸出範例 4

can't be transcribed  
can't be translated

至少一種解法(需含註解) p.s. 如果先把程式碼貼到 rep1，再複製貼到 word，會自動上色喔！

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

int main()
{
    //宣告陣列 X(去紀錄使用者輸入的 DNA 核甘酸序列)、陣列 Y(轉錄後的 mRNA 核甘酸序列)、N
    為核甘酸的數量
    string x[100], y[100];
    int N;

    cin >> N;
    //將使用者輸入的 DNA 核甘酸序列存到 x 陣列裡面
    for (int i=0 ; i < N ; ++i)
    {
        //輸入是從 5' 輸入到 3'
        cin >> x[i];
    }

    //將陣列 X(DNA 核甘酸序列)轉為陣列 Y(mRNA 核甘酸序列)
    //因為方向的不同，所以要倒過來

    for (int i=0 ; i < N ; ++i )
    {
        if(x[i]=="A") y[N-(i+1)]="U";

        if(x[i]=="T") y[N-(i+1)]="A";
```

```

    if(x[i]=="C") y[N-(i+1)]="G";

    if(x[i]=="G") y[N-(i+1)]="C";

}
//用變數 num 去確認使用者輸入的東西是否符合 DNA 核甘酸序列(A T C G)
int num=0;
for(int i=0; i<N; ++i)
{
    if((x[i]=="A") || (x[i]=="T") || (x[i]=="C") || (x[i]=="G"))
    {
        ++num;
    }
}
//如果符合的話就輸出轉錄後的 mRNA 序列(從 5' 端到 3' 端)
if(num==N)
{
    cout<<"mRNA: 5'[";

    for(int j=0 ; j< N ; ++j)
    {
        cout<<y[j];
    }
    cout<< "]"3'"<<endl;
}
//不符合的話就顯示"can't be transcribed"
else
{
    cout<<"can't be transcribed"<<endl;
}

//開始轉譯：宣告 pro 陣列(蛋白質序列)
//acc 變數是用來檢查 mRNA 中間有沒有出現不該出現的 "起始密碼子" 與 "終止密碼子"
string pro[100];
int acc=0;
//因為密碼子是三個一組變成一個蛋白質，如果密碼子不是三的倍數的話就不能被轉譯
//且不能被轉錄就不能被轉譯
if(N%3 != 0 || num!=N)

```

```

{

}
else
{
    //將 mRNA 轉成蛋白質
    //因為 mRNA 上的密碼子是三個一組，所以用 i=i+3 去讀 y 陣列裡的資料；
    //因為 pro 的陣列是造順序填入的，所以用++j 去填入
    for(int i=0,j=0 ; i < N ; i=i+3,++j)
    {
        if((y[i]=="U" && y[i+1]=="U" && y[i+2]=="U") || (y[i]=="U" &&
y[i+1]=="U" && y[i+2]=="C"))    pro[j]="Phe";
        if((y[i]=="U" && y[i+1]=="U" && y[i+2]=="A") || (y[i]=="U" &&
y[i+1]=="U" && y[i+2]=="G"))    pro[j]="Leu";
        if((y[i]=="C" && y[i+1]=="U" && y[i+2]=="U") || (y[i]=="C" &&
y[i+1]=="U" && y[i+2]=="C"))    pro[j]="Leu";
        if((y[i]=="C" && y[i+1]=="U" && y[i+2]=="A") || (y[i]=="C" &&
y[i+1]=="U" && y[i+2]=="G"))    pro[j]="Leu";
        if((y[i]=="A" && y[i+1]=="U" && y[i+2]=="U") || (y[i]=="A" &&
y[i+1]=="U" && y[i+2]=="C") || (y[i]=="A" && y[i+1]=="U" && y[i+2]=="A"))
pro[j]="Ile";
        if((y[i]=="A" && y[i+1]=="U" && y[i+2]=="G"))
        {
            pro[j]="N 端[Met]";
            //acc 加一以記錄起始與終止密碼子出現的次數
            ++acc;

        }
        if((y[i]=="G" && y[i+1]=="U" && y[i+2]=="U") || (y[i]=="G" &&
y[i+1]=="U" && y[i+2]=="C"))    pro[j]="Val";
        if((y[i]=="G" && y[i+1]=="U" && y[i+2]=="A") || (y[i]=="G" &&
y[i+1]=="U" && y[i+2]=="G"))    pro[j]="Val";
        if((y[i]=="U" && y[i+1]=="C" && y[i+2]=="U") || (y[i]=="U" &&
y[i+1]=="C" && y[i+2]=="C"))    pro[j]="Ser";
        if((y[i]=="U" && y[i+1]=="C" && y[i+2]=="A") || (y[i]=="U" &&
y[i+1]=="C" && y[i+2]=="G"))    pro[j]="Ser";
        if((y[i]=="C" && y[i+1]=="C" && y[i+2]=="U") || (y[i]=="C" &&
y[i+1]=="C" && y[i+2]=="C"))    pro[j]="Pro";
        if((y[i]=="C" && y[i+1]=="C" && y[i+2]=="A") || (y[i]=="C" &&
y[i+1]=="C" && y[i+2]=="G"))    pro[j]="Pro";
        if((y[i]=="A" && y[i+1]=="C" && y[i+2]=="U") || (y[i]=="A" &&
y[i+1]=="C" && y[i+2]=="C"))    pro[j]="Thr";
    }
}

```

```

        if((y[i]=="A" && y[i+1]=="C" && y[i+2]=="A") || (y[i]=="A" &&
y[i+1]=="C" && y[i+2]=="G"))    pro[j]="Thr";
        if((y[i]=="G" && y[i+1]=="C" && y[i+2]=="U") || (y[i]=="G" &&
y[i+1]=="C" && y[i+2]=="C"))    pro[j]="Ala";
        if((y[i]=="G" && y[i+1]=="C" && y[i+2]=="A") || (y[i]=="G" &&
y[i+1]=="C" && y[i+2]=="G"))    pro[j]="Ala";
        if((y[i]=="U" && y[i+1]=="A" && y[i+2]=="U") || (y[i]=="U" &&
y[i+1]=="A" && y[i+2]=="C"))    pro[j]="Tyr";
        if((y[i]=="U" && y[i+1]=="A" && y[i+2]=="A") || (y[i]=="U" &&
y[i+1]=="A" && y[i+2]=="G") || (y[i]=="U" && y[i+1]=="G" && y[i+2]=="A"))
        {
            pro[j]="C端";
            //acc 加一以記錄起始與終止密碼子出現的次數
            ++acc;
        }
        if((y[i]=="C" && y[i+1]=="A" && y[i+2]=="U") || (y[i]=="C" &&
y[i+1]=="A" && y[i+2]=="C"))    pro[j]="His";
        if((y[i]=="C" && y[i+1]=="A" && y[i+2]=="A") || (y[i]=="C" &&
y[i+1]=="A" && y[i+2]=="G"))    pro[j]="Gln";
        if((y[i]=="A" && y[i+1]=="A" && y[i+2]=="U") || (y[i]=="A" &&
y[i+1]=="A" && y[i+2]=="C"))    pro[j]="Asn";
        if((y[i]=="A" && y[i+1]=="A" && y[i+2]=="A") || (y[i]=="A" &&
y[i+1]=="A" && y[i+2]=="G"))    pro[j]="Lys";
        if((y[i]=="G" && y[i+1]=="A" && y[i+2]=="U") || (y[i]=="G" &&
y[i+1]=="A" && y[i+2]=="C"))    pro[j]="Asp";
        if((y[i]=="G" && y[i+1]=="A" && y[i+2]=="A") || (y[i]=="G" &&
y[i+1]=="A" && y[i+2]=="G"))    pro[j]="Glu";
        if((y[i]=="U" && y[i+1]=="G" && y[i+2]=="U") || (y[i]=="U" &&
y[i+1]=="G" && y[i+2]=="C"))    pro[j]="Cys";
        if((y[i]=="U" && y[i+1]=="G" && y[i+2]=="G"))    pro[j]="Trp";
        if((y[i]=="C" && y[i+1]=="G" && y[i+2]=="U") || (y[i]=="C" &&
y[i+1]=="G" && y[i+2]=="C"))    pro[j]="Arg";
        if((y[i]=="C" && y[i+1]=="G" && y[i+2]=="A") || (y[i]=="C" &&
y[i+1]=="G" && y[i+2]=="G"))    pro[j]="Arg";
        if((y[i]=="A" && y[i+1]=="G" && y[i+2]=="U") || (y[i]=="A" &&
y[i+1]=="G" && y[i+2]=="C"))    pro[j]="Ser";
        if((y[i]=="A" && y[i+1]=="G" && y[i+2]=="A") || (y[i]=="A" &&
y[i+1]=="G" && y[i+2]=="G"))    pro[j]="Arg";
        if((y[i]=="G" && y[i+1]=="G" && y[i+2]=="U") || (y[i]=="G" &&
y[i+1]=="G" && y[i+2]=="C"))    pro[j]="Gly";
        if((y[i]=="G" && y[i+1]=="G" && y[i+2]=="A") || (y[i]=="G" &&
y[i+1]=="G" && y[i+2]=="G"))    pro[j]="Gly";
    }

```

```
}

//如果符合這三項，輸出蛋白質序列
if((pro[0]=="N端[Met]"&& (pro[(N/3)-1]=="]C端")&& acc==2 )
{
    cout<<"protein: ";
    for(int i=0; i< (N/3)-2; ++i)
    {
        cout<<pro[i]<<"-";
    }
    //因為最後的終止密碼子還有倒數第二個後面都不要"-",所以獨立出來輸出
    cout<<pro[(N/3)-2];
    cout<<pro[(N/3)-1];

}
//不然就輸出"can't be translated"
else
{
    cout<<"can't be translated"<<endl;
}

}
```

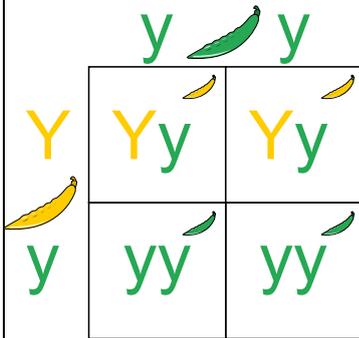
## 遺傳疾病機率計算

### 題目說明



經過 8 年的豌豆雜交實驗，人稱「遺傳學之父」的孟德爾發表了他的論文：植物雜交實驗。他的實驗結論可稱為「孟德爾定律」，主要可以分為兩大部分：

1. **分離律**：細胞中有成對的基本遺傳單位，在雜交的生殖細胞中，成對的遺傳單位一個來自雄性，一個來自雌性，形成配子時這些遺傳單位彼此分離。
2. **獨立分配律**：兩種遺傳因子的分離之間不會有影響，單一對遺傳因子依然會遵從分離律，且成對的因子在分離後會進行自由配對。



**旁氏表(Punnett square)**，又稱為棋盤法，以發明者 Reginald C. Punnett 的姓氏來命名，是用來計算基因型的概率的方法。而這個旁氏表也完全符合孟德爾提出的兩大定律，跟孟德爾作的豌豆實驗也相符，日後成為大家計算基因型的簡便快速方法。

### 亨丁頓舞蹈症(Huntington's Disease)

亨丁頓舞蹈症是一種體染色體顯性遺傳所造成的腦部退化疾病，是因為第四條染色體上的鹼基對序列發生異常所引發的疾病。症狀是不自主運動、臉部輕微抽搐、坐立不安核對於運動缺發整合協調性。

### 鐮型血球貧血症 (Sickle Cell Anemia)

鐮行血球貧血症是體染色體隱性遺傳的疾病。患者是由於位於第 11 對染色體的β 胜肽鏈的 DNA 序列的第 20 個核苷酸發生突變，導致血紅素的結構發生改變，讓本來是雙凹型圓盤的紅血球扭曲成鐮刀型。這不但會讓紅血球攜氧量降低，也會讓血液的粘滯度提高，造成微血管阻塞。

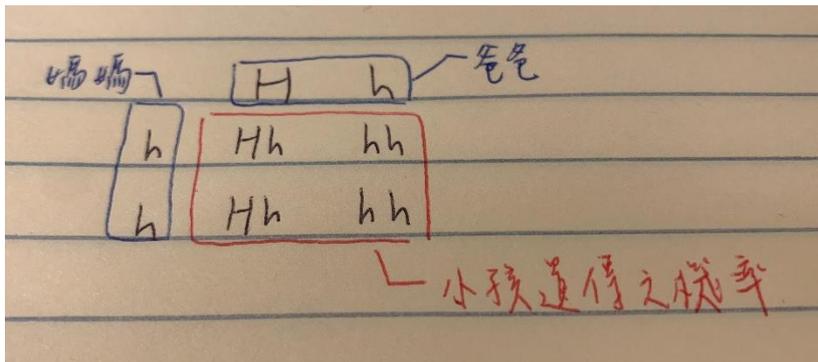
由於兩種疾病是在不同對的染色體上，並不會發生連鎖與互換的現象，所以機率概算可以用旁氏表。

## 輸入說明

1. 分別照著指示輸入爸爸媽媽對於兩種疾病的基因型：
  - 亨丁頓舞蹈症 (顯性基因 H 隱性基因 h)
    - ◆ 可輸入 HH/Hh/hh (記得兩個等位基因中間要空格)
  - 鎌刀型貧血症 (顯性基因 S 隱性基因 s)
    - ◆ 可輸入 SS/Ss/ss (記得兩個等位基因中間要空格)
2. 照著指示填入 yes 或 no (有沒有表現這個病症) , 表示想要算的機率的組合

## 輸出說明

1. 首先輸出的是分別兩種疾病的旁氏圖 , 爸爸是在上面 , 媽媽是在左邊 , 而只有顯示最後小孩遺傳機率的結果 , 如下圖



2. 在使用者輸入完 yes 或 no 後就會出現這個組合所發生的機率。

## 至少兩筆範例測資

### 輸入範例 1

H H  
h h  
S s  
s s  
yes  
yes

### 輸出範例 1

爸爸對亨丁頓舞蹈症的基因型 : H H

媽媽對亨丁頓舞蹈症的基因型 : h h

爸爸對鎌型血球貧血症的基因型 : S s

媽媽對鎌型血球貧血症的基因型 : s s

-----  
旁氏表 :

Hh Hh

Hh Hh

Ss ss

Ss ss

-----  
計算小孩遺傳機率

小孩有無亨丁頓舞蹈症 (請輸入 yes 或 no) :

yes

小孩有無鎌型血球貧血症 (請輸入 yes 或

no) : yes

0%

<p><b>輸入範例 2</b></p> <p>H h h h S s s s no no</p>	<p><b>輸出範例 2</b></p> <p>爸爸對亨丁頓舞蹈症的基因型：H h 媽媽對亨丁頓舞蹈症的基因型：h h 爸爸對鎌型血球貧血症的基因型：S s 媽媽對鎌型血球貧血症的基因型：s s</p> <p>-----</p> <p>旁氏表：</p> <p>Hh hh Hh hh</p> <p>Ss ss Ss ss</p> <p>-----</p> <p>計算小孩遺傳機率</p> <p>小孩有無亨丁頓舞蹈症(請輸入 yes 或 no)： no 小孩有無鎌型血球貧血症(請輸入 yes 或 no)：no 50%</p>
---	--

### 至少一種解法(需含註解)

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

int main()
{
    //宣告四個陣列分別記錄媽媽和爸爸在兩個不同疾病的基因型
    string mh[10], dh[10], ms[10], ds[10];
    //讓陣列結束的未知數，為 2，因為一個基因型裡面只有兩個等位基因
    int N=2;

    //以下都是讓使用者輸入爸爸與媽媽的基因型到對應的陣列中
    cout<<"爸爸對亨丁頓舞蹈症的基因型：";

    for(int i=0; i<N; ++i)
    {
        cin>>dh[i];
    }

    cout<<"媽媽對亨丁頓舞蹈症的基因型：";

    for(int i=0; i<N; ++i)
```

```

{
    cin>>mh[i];
}

cout<<"爸爸對鎌型血球貧血症的基因型：";

for(int i=0; i<N; ++i)
{
    cin>>ds[i];
}

cout<<"媽媽對鎌型血球貧血症的基因型：";

for(int i=0; i<N; ++i)
{
    cin>>ms[i];
}

//為了要呈現旁氏表以及算機率，開兩個二維陣列分別記錄兩個不同的疾病
string h[2][2],s[2][2];
//將爸爸與媽媽的基因型變成旁氏格 以二維陣列呈現
for(int i=0; i<N; ++i)
{
    for(int j=0 ; j<N ; ++j)
    { //把前面的 dh、mh 陣列合併變成 h 二維陣列
        h[i][j]= mh[i]+dh[j];
    }
}

for(int i=0; i<N; ++i)
{
    for(int j=0 ; j<N ; ++j)
    { //把前面的 ds、ms 陣列合併變成 s 二維陣列
        s[i][j]=ms[i]+ds[j];
    }
}

//這邊是讓格式變好看
cout<<"-----"<<endl;
cout<<"旁氏表："<<endl;
cout<<endl;

//把亨丁頓舞蹈症的旁氏表輸出

```

```

for(int i=0; i<N; ++i)
{
    for(int j=0 ; j<N ; ++j)
    {
        //但是基因的代表法不會是小寫在前面，所以出線的話要校正
        if(h[i][j]=="Hh" || h[i][j]=="hH")
        {
            cout<<"Hh"<<" ";
        }
        else if(h[i][j]=="HH")
        {
            cout<<"HH"<<" ";
        }
        else
        {
            cout<<"hh"<<" ";
        }
    }
    //此時換行才會像一個表格
    cout<<endl;
}

//格式空行變好看
cout<<endl;

////把鐮型血球貧血症的旁氏表輸出
for(int i=0; i<N; ++i)
{
    for(int j=0 ; j<N ; ++j)
    {
        // //但是基因的代表法不會是小寫在前面，所以出線的話要校正
        if(s[i][j]=="Ss" || s[i][j]=="sS")
        {
            cout<<"Ss"<<" ";
        }
        else if(s[i][j]=="SS")
        {
            cout<<"SS"<<" ";
        }
        else
        {
            cout<<"ss"<<" ";
        }
    }
}

```

```

    }
    //此時換行才會像一個表格
    cout<<endl;
}

//格式好看
cout<<"-----"<<endl;

//宣告，計算基因型出現的次數
int numh=0,nums=0;

for(int i=0; i<N; ++i)
{
    for(int j=0 ; j<N ; ++j)
    {
        //因為這個是顯性遺傳疾病，所以只要有帶顯性基因都會發病
        if(h[i][j]=="Hh" || h[i][j]=="HH")
        {
            ++numh;
        }
    }
}

for(int i=0; i<N; ++i)
{
    for(int j=0 ; j<N ; ++j)
    {
        //因為這個是隱性遺傳疾病，所以要兩個都帶隱性基因才會發病
        if(s[i][j]=="ss")
        {
            ++nums;
        }
    }
}

//讓使用者輸入，看他們想看哪種組合的機率
string hcin,scin;

cout<<"計算小孩遺傳機率"<<endl;

cout<<"小孩有無亨丁頓舞蹈症(請輸入 yes 或 no) : ";
cin >>hcin;

```

```
cout<<"小孩有無鎌型血球貧血症(請輸入 yes 或 no) : ";
cin >>scin;

//宣告變數，是 double 是因為算機率要有小數
double possh, posss, poss;

//因為旁氏格有四個可能性，所以出現次數要除以四
//為了讓 possh 是小數，要除 4.0 而不是 4
if(hcin=="yes")
{
    possh=numh/4.0;
}
else
{
    possh=(4-numh)/4.0;
}

if(scin=="yes")
{
    posss=nums/4.0;
}
else
{
    posss=(4-nums)/4.0;
}

//最後想呈現的是百分比而非小數
poss=possh*posss;
cout<<poss*100<<"%"<<endl;
}
```

中華民國 110 年 7 月 3 日