

程式設計練習 Final

模擬：龜兔賽跑

在這個習題中，讀者將建立龜兔賽跑的遊戲，透過產生亂數模擬這個故事。遊戲中，競賽者由 70 個方格中的「方格 1」開始比賽。每個方格代表比賽路線上的可能位置。終點線在第 70 格。第一個到達或通過第 70 格的競賽者，會得到一桶新鮮的胡蘿蔔和萵苣作為獎賞。競賽路線沿著滑溜的山坡向山頂進行，有時候，競賽者會失足跌落地面。模擬中，有個時鐘每秒滴答一次。時鐘每滴答一次，程式應按圖 8.21 的規則，使用函式 `moveTortoise` 和 `moveHare` 調整動物的位置。這些函式應該以指標為基礎，使用傳參考呼叫來調整烏龜和兔子的位置。

動物	移動方式	時間百分比	實際動作
烏龜	快走	50%	往右移 3 格
	滑倒	20%	往左移 6 格
	慢走	30%	往右移 1 格
野兔	睡覺	20%	不移動
	大跳躍	20%	往右移 9 格
	大滑倒	10%	往左移 12 格
	小跳躍	30%	往右移 1 格
	小滑倒	20%	往左移 2 格

圖 8.21 移動烏龜和野兔的規則

請使用變數來追蹤這些動物的位置 (即 1-70 的位置編號)。每隻動物由位置 1 開始起跑。若動物在第 1 格就向左移動，則將動物移回方格 1。

請用介於 1 到 10 之間的隨機整數 i 產生表格內的比率。對烏龜來說，當 $1 \leq i \leq 5$ 時，會快速移動；當 $6 \leq i \leq 7$ 時，會滑一跤；當 $8 \leq i \leq 10$ 時，慢速移動。兔子也以相同方式移動。

當時鐘每滴答一下 (即重複執行一次迴圈)，印出代表 70 個位置的直線，以字母 T 代表烏龜的位置，以字母 H 顯示兔子的位置。除了 T、H，其它的位置都留白。

每當印出一行，請測試是否有動物抵達或超過第 70 格。假如有，請印出獲勝者，然後結束這個模擬程式。如果烏龜贏了，印出 `TORTOISE WINS!!!YAY!!!` 野兔贏了，請列印 `Hare wins. Yuch.` 如果兩隻動物在相同的時鐘滴答下贏了，你可讓烏龜贏，或列印 `"It's a tie!"`。

- (1) 請完成檔案 `step1.cpp` 中的函式 `void printCurrentPositions(const int * const snapperPtr, const int * const bunnyPtr)`，使程式執行後印出兔子與烏龜的位置 (即 `main` 中的 `hare` 與 `tortoise`)，如下圖所示。

-----	-----	-----	-----	-----	-----	-----
					H	
		T				

- (2) 請完成檔案 `step2.cpp` 中的函式 `void moveTortoise(int * const turtlePtr)`，使程式執行後可模擬烏龜的移動 (即在此函式中改變烏龜的位置)。請參考圖 8.21 的機率來產生烏龜的移動，若移動後的位置小於 1，則設定為 1，若移動後的位置大於 70，則設定為 70。main 函式迴圈中的 `Sleep(n)` 將暫停程式 n 毫秒，`Clr()` 將清除畫面。請將上題中所完成的函式複製至此程式。

