20230512 計算機概論 Practice 13

**題目：**

一年一度的施伯伯跑跑車大賽即將開始，賽季規則如下：

1. 在筆直的賽道上，賽道長度為2000公尺，最先抵達終點的車獲得冠軍，當冠軍得主出現時比賽就結束。秒數請以1秒為單位計算，當第n秒時有任何車跑的距離大於等於2000公尺，則比賽立即結束。
2. 大賽規定，只能使用以下三種車種：

|  |  |  |
| --- | --- | --- |
| 編號 | 名稱 | 車子原始速度(speed) |
| 1 | Benz | 1秒14公尺 |
| 2 | BMW | 1秒8公尺 |
| 3 | Mazda | 1秒11公尺 |

表一、大賽規定車種

熟悉賽車的你知道好的駕駛會讓車子跑得更快，所以你分析了每位駕駛開車時能讓原本的汽車原始速度增加多少，稱其為「駕駛速度增值(driver’s boost)」，並整理成下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 駕駛名稱 | 駕駛速度增值 | 駕駛口頭禪 |
| A | 律文 | 1秒多3公尺 | 速度可以殺人，也可以救人。 |
| B | 桃宇 | 1秒多2公尺 | 你覺得我們兩個，誰會先踩煞車？ |
| C | 餅則 | 1秒多1公尺 | 寧可粉身碎骨也不能放慢速度。 |
| D | 凱西 | 1秒多3公尺 | 沒有過不去的彎道，只有過不去的心情。 |
| E | 丕老大 | 1秒多10公尺 | 我沒有朋友，我只有家人。 |
| F | 韶恩 | 1秒多2公尺 | 我從不害怕死亡，只害怕無疾而終。 |
| G | 韓哥 | 1秒多1公尺 | 叫我韓好了，兄弟都叫我韓。 |

表二、駕駛基本資料

舉例來說，如果律文開Benz，那他一秒的速度會是車子原始速度+駕駛速度增值=14+3=17公尺。

熟悉物件導向的你決定利用Java的Inheritance寫一個程式，能夠輸入各個參賽手使用的車種，並預測比賽結果。

**題目要求：**

Class **BaseCar**中需包含有以下元件(程式中必須使用此class 否則0分)

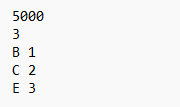
|  |  |  |
| --- | --- | --- |
| 名稱 | 型態 | 說明 |
| driver | string | 駕駛的編號 |
| speed | int | 汽車速度 |
| driver\_boost | int | 駕駛速度增值 |

注意，這些參數可以增加不能減少！

除此之外，你需要透過Inheritance的方式去繼承上面的BaseCar，產生Class Benz, Class BMW, 以及Class Mazda，至於這些class繼承後要增加/更改什麼功能就留給同學自由發揮。

**輸入/輸出格式：**

輸入的第一行會給你駕駛的數量，接著每一行都是那位駕駛與他開的車種，如下面所示：



駕駛編號(英文大寫)

車種編號(整數)

駕駛數量(整數)

輸入不會有不符合格式的情況發生，不用設計防呆。

你的程式需要輸出誰是最後冠軍，輸出規則如下：

1. 比賽從0秒開始，當經過的時間為10的倍數時，印出在那秒開始時(移動前)每台車的位置，如：”B:34m C:25m E:24m ”，按照input順序印出(最後有換行)。
2. 大會評審每一秒才會看一次賽道狀況，所以不用考慮微秒時的賽道的狀況。只要任何人跑的距離大過賽道長度，就印出駕駛編號(大寫) + “ wins! ”，以及所有人當時的位置，按照input順序印出。
3. 如果在比賽結束的那一秒有大於1台車超過終點線，就印出”More than 1 winner!”的字串，以及所有人當時的位置，按照input順序印出。

The actual input and output for running your program is something that looks like the following：

**例一：**

# input自此始，但不包括此行

2

B 2

C 1

# input至此止，但不包括此comment

# output自此始，但不包括此行

When 10sec starts, B:100m C:150m

When 20sec starts, B:200m C:300m

When 30sec starts, B:300m C:450m

When 40sec starts, B:400m C:600m

When 50sec starts, B:500m C:750m

When 60sec starts, B:600m C:900m

When 70sec starts, B:700m C:1050m

When 80sec starts, B:800m C:1200m

When 90sec starts, B:900m C:1350m

When 100sec starts, B:1000m C:1500m

When 110sec starts, B:1100m C:1650m

When 120sec starts, B:1200m C:1800m

When 130sec starts, B:1300m C:1950m

C wins! B:1340m C:2010m

# output至此止，但不包括此comment

**例二：**

# input自此始，但不包括此行

3

E 2

A 1

C 1

# input至此止，但不包括此comment

# output自此始，但不包括此行

When 10sec starts, E:180m A:170m C:150m

When 20sec starts, E:360m A:340m C:300m

When 30sec starts, E:540m A:510m C:450m

When 40sec starts, E:720m A:680m C:600m

When 50sec starts, E:900m A:850m C:750m

When 60sec starts, E:1080m A:1020m C:900m

When 70sec starts, E:1260m A:1190m C:1050m

When 80sec starts, E:1440m A:1360m C:1200m

When 90sec starts, E:1620m A:1530m C:1350m

When 100sec starts, E:1800m A:1700m C:1500m

When 110sec starts, E:1980m A:1870m C:1650m

E wins! E:2016m A:1904m C:1680m

# output至此止，但不包括此comment

**評分標準：**

For each input (test data) file, your program must produce an output file with a content that is EXACTLY THE SAME AS the expected output. This is the only way your program can "pass" any particular test.