

無尾熊遊戲

Koala 想出了一個新的遊戲，而她要向你挑戰！遊戲開始時，Koala 放了 N 個物品，這些物品的編號為 0 到 $N - 1$ 。接著，她偷偷的將這些物品各自給予其價值，範圍是由 1 到 N ，而且每個物品的價值都不一樣。物品 i 會對應到價值 P_i 。Koala 要你認出序列 $P = P_0, P_1, \dots, P_{N-1}$ 的某些特徵。

為了達成目標，你會要求 Koala 進行一連串的回合。每一回合，你會拿到 W 顆藍色石頭，而 Koala 會拿到 W 顆紅色石頭。首先，你可以將一些石頭 (有可能全放) 放到一些你選擇的物品旁邊。接著，Koala 會根據你的放法，也同樣地將一些石頭 (有可能全放) 放到一些物品旁邊。最後，Koala 可以贏得那些紅色石頭的數量超過藍色石頭的數量的物品。Koala 每次都會想辦法讓自己的贏得的物品總價值越高越好。如果有許多不同的方式可以達成目標，Koala 就會讓贏得的物品數量越多越好。若仍然有不同的方式可以達成的話，她就會選擇其中任何一種方法。

Koala 非常的懶，如果你要求她玩太多回合的話，她就會睡著。因此，你的工作就是盡可能用最少回合來確認 Koala 所設定的序列 P 。

任務

在這個任務中，你必須做出四個函式: `minValue`、`maxValue`、`greaterValue` 和 `allValues`。每種函式要能辨別出序列 P 的一種特徵。**強烈建議** 你使用範例所附的程式作為解決這個任務的出發點。請注意，就算你只想解決某些子任務，你仍 **必須** 要實作出這四個函式 (即使函式內容是空的)。你的程式不能使用標準輸入輸出，也不能藉助任何檔案輸出入。

每個函式都有參數 N 代表這個遊戲中的物品個數，而參數 W 則代表你和 Koala 在每次遊戲中各自拿到的石頭數。

- `minValue(N, W)` --- 這個函數要回傳價值最低的物品編號 i ，也就是說: $P_i = 1$ 。
- `minValue(N, W)` --- 這個函數要回傳價值最高的物品編號 i ，也就是說: $P_i = N$ 。
- `greaterValue(N, W)` --- 這個函數要比較物品 0 以及物品 1 的價值，並回傳價值較大的物品編號。更明白地說，若 $P_0 > P_1$ ，就回傳 0 ，否則就回傳 1 。
- `allValues(N, W, P)` --- 這個函數必須將整個序列 P 給算出來，並存到陣列 P 中: $P[i]$ 代表了物品 i 的價值 P_i ，其中 $0 \leq i \leq N - 1$ 。

在每個測試資料中，`grader` 只會呼叫這些函式中的 **其中一個** 函式 **一次或多次**。每呼叫一次函式都視為一場新的遊戲。至於會呼叫哪個函式、最多會呼叫幾次，則依各個子任務而定 (容後說明)。你可以假設 Koala 會先固定序列 P 再呼叫函式，序列 P 的值在函式執行期間不會更動。在開始下一次函式呼叫前，她才可能會改變序列的值。

實作每個函式時，會用到函式 `playRound` 來得到 Koala 的序列的資訊。

- `playRound(B, R)` --- 要求 Koala 跟你玩一回合

- 陣列 B 代表你放多少顆藍色石頭在每個物品旁邊。明確地說，對於所有的 $0 \leq i \leq N - 1$ ，會有 $B[i]$ 顆藍色石頭會放在物品 i 旁。每個 $B[i]$ 都會是非負整數而且 $B[0] + B[1] + \dots + B[N-1]$ 的值不會超過 W 。
- grader 會填入陣列 R 的值，代表 Koala 要放的紅色石頭的數量。明確地說，對於所有的 $0 \leq i \leq N - 1$ ，Koala 會放 $R[i]$ 顆紅色石頭在物品 i 的旁邊。
- 每個子任務對 **每一場新的遊戲** 呼叫 playRound 的次數都有限制。呼叫的次數越少，你得到的分數越高 (容後說明)。

範例資料 [0 分]

- 總共有 5 個範例測資。每個測資呼叫前述函數中的其中一個一次。請看後續 **範例** 以得到更詳細的說明。
- $N = 6$.
- $P = 5, 3, 2, 1, 6, 4$.
- 一場遊戲中最多呼叫 playRound 3200 次。

子任務 1 [4 分]

- 在這個子任務中，grader 只會呼叫 minValue。這個函式在每個測試資料中最多只會被呼叫 100 次。
- $N = 100$.
- $W = 100$.
- 一場遊戲中最多呼叫 playRound 2 次。

子任務 2 [最多 15 分]

- 在這個子任務中，grader 只會呼叫 maxValue。這個函式在每個測試資料中最多只會被呼叫 100 次。
- $N = 100$.
- $W = 100$.
- 一場遊戲中最多呼叫 playRound 13 次。
- 每個測試資料的分數會依照在該測資所有遊戲中呼叫 playRound 次數的最大值 C_{\max} 來評分。精確來說：
 - 若 $C_{\max} \leq 4$ ，會得到 15 分。
 - 若 $5 \leq C_{\max} \leq 13$ ，會得到 7 分。

子任務 3 [最多 18 分]

- 在這個子任務中，grader 只會呼叫 greaterValue。這個函式在每個測試資料中最多只會被呼叫 1100 次。
- $N = 100$.

- $W = 100$.
- 一場遊戲中最多呼叫 playRound 14 次。
- 每個測試資料的分數會依照在該測資所有遊戲中呼叫 playRound 次數的最大值 C_{\max} 來評分。精確來說：
 - 若 $C_{\max} \leq 3$ ，會得到 18 分。
 - 若 $C_{\max} = 4$ ，會得到 14 分。
 - 若 $C_{\max} = 5$ ，會得到 11 分。
 - 若 $6 \leq C_{\max} \leq 14$ ，會得到 5 分。

子任務 4 [10 分]

- 在這個子任務中，grader 只會呼叫 allValues。這個函式在每個測試資料中 **被呼叫 1 次**。
- $N = 100$.
- $W = 200$.
- 最多可以呼叫 playRound 700 次。

子任務 5 [最多 53 分]

- 在這個子任務中，grader 只會呼叫 allValues。這個函式在每個測試資料中 **被呼叫 1 次**。
- $N = 100$.
- $W = 100$.
- 最多可以呼叫 playRound 3200 次。
- 每個測試資料的分數會依照在遊戲中呼叫 playRound 的次數 C 來評分。精確來說：
 - 若 $C \leq 100$ ，得到 53 分。
 - 若 $101 \leq C \leq 3200$ ，得到 $\lfloor 53 - 8 \log_2 (C/100) \rfloor$ 分，其中 $\lfloor x \rfloor$ 為小於等於 x 的最大整數。而當 $C = 3200$ ，你會得到 13 分。

評分

- 在每個測試資料中，你的程式必須在不超過時間以及記憶體限制下完成這個任務，這包含 grader 呼叫 playRound 時所消耗的時間以及記憶體空間。在評估時間和記憶體時，你可以假設 sample grader 和評分時用的 grader 有 **相同的功能** 以及 **相似的實作方式**。
- 若在呼叫 playRound 時使用了不合法的陣列 B，或是在測資中的任意一場遊戲中呼叫 playRound 的次數超過了限制，則該測資就會是 Not Correct，分數為 0。
- 若你在一個測試資料的某場遊戲無法正確地算出 Koala 序列 P 的指定特徵，則該測資就會是 Not Correct，分數為 0。
- 子任務 4 和子任務 5 都要求你要實作出函式 allValues，但是 W 給的範圍不一樣。你

可以利用這個條件來區分這兩個子任務 -- 若要得知更多細節，請參見所附範例的實作。

- 你最多可以提交 **60** 次，兩次提交中間至少要間隔 2 分鐘。

範例

以如下序列 P 為例。

i	0	1	2	3	4	5
P_i	5	3	2	1	6	4

下面是一連串呼叫 `playRound` 的範例，以及 `grader` 針對每次呼叫時會做的合法回應 (請注意同樣的呼叫可能有不只一種合法回應)。

W	程式呼叫	grader 可能的回應	說明
6	<code>playRound([0, 3, 0, 2, 1, 0], R)</code>	$R = [1, 1, 1, 0, 2, 1]$	你分別放了 3、2 和 1 顆藍色石頭在物品 1、3 和 4 旁邊，而物品 0、2 和 5 並沒有放石頭。Koala 在物品 0、1、2 和 5 各放了 1 顆紅色石頭並且放了 2 顆紅色石頭在物品 4，而物品 3 沒有放石頭。因此，她贏得了物品 0、2、4 和 5，而總價值為 $5 + 2 + 6 + 4 = 17$ ，是在這個情況中可以獲得的最大價值。
6	<code>playRound([1, 2, 3, 1, 2, 0], R)</code>	不合法的呼叫。你的程式會終止並被視為 Not Correct，在此測試資料中，你的分數為 0。	你放了 $1 + 2 + 3 + 1 + 2 = 9 > 6 = W$ 顆石頭，是不合法的。
12	<code>playRound([1, 2, 3, 1, 2, 0], R)</code>	$R = [2, 3, 0, 2, 3, 1]$	你不一定要把 W 顆石頭全部放到物品旁，而 Koala 也不一定會把 W 顆石頭全部放下去。
6	<code>playRound([0, 1, 0, 0, 1, 0], R)</code>	$R = [1, 0, 1, 1, 2, 1]$	如果有許多不同的方法可以讓 Koala 贏到最多的價值，那麼她會選擇可以拿到最多的物品的方法，因此本例中 $R = [1, 2, 0, 0, 2, 1]$ 不是一個合法的回應。

上傳解答時，下列五個測試範例的回饋會依序列在 "Sample Data" 之下。每個測試資料最多可呼叫 `playRound` 3200 次。

·
·
·
·
·

#	Grader calls	預期的回傳值	說明
1	minValue(6, 6)	3	$P_3 = 1$ ，代表物品 3 的價值最低。
2	maxValue(6, 6)	4	$P_4 = 6 = N$ ，代表物品 4 的價值最高。
3	greaterValue(6, 6)	0	$P_0 = 5 > 3 = P_1$ ，代表物品 0 的價值比物品 1 高。
4	allValues(6, 12, P)	無， $P = [5, 3, 2, 1, 6, 4]$	函式 allValues 並沒有回傳值，而是要把正確數值放到陣列 P 中。
5	allValues(6, 6, P)	無， $P = [5, 3, 2, 1, 6, 4]$	同上。

評分範例

sample grader 會從標準輸入中讀入以下格式：

- 第一行：兩個整數 F, G ；
- 第二行至第 $G + 1$ 行：每一行有兩個整數 N, W ，接下來有 N 個整數 P_0, P_1, \dots, P_{N-1} ，代表一場遊戲。

整數 F 代表 sample grader 要呼叫哪種函式：

F	呼叫的函式
1	minValue
2	maxValue
3	greaterValue
4	allValues

整數 G 代表要呼叫函式幾次。接下來每一行都代表一場遊戲中 Koala 設定的序列。

sample grader 會對每次的函式呼叫在標準輸出中輸出兩行資訊。第一行包含了 playRound 被呼叫的次數。

- 若 $F = 4$ ，第二行就會包含了藉由你的函式 allValues 算出來的序列 P 的內容。
- 如果 ($F = 1, 2$) 或 3 ，第二行就只有一個整數：代表你的相對應函式的回傳值。

例如，前面第四個範例測試資料 (呼叫 allValues，而且 $N = 6$ 、 $W = 12$) 就可以表示成以下的輸入檔。

```
4 1
6 12 5 3 2 1 6 4
```