

Programming assignment #2

寶可夢戰鬥力排行榜

Objective

1. To understand how to design an algorithm.
2. To exercise the concept of recursion.
3. To learn how to use tree.

Problem

Adar 老師需要一個寶可夢即時戰鬥力排行榜，戰鬥力數值越高的排行越前面，而該排行榜又有兩種模式：(1)戰鬥力數值相同的排名相同 (排名=戰鬥力數值大於自己的個數+1)；(2)戰鬥力數值相同時，按照 ID 大小升序排列(排名=戰鬥力數值大於自己的個數+戰鬥力數值相同且 ID 數值比自身 ID 數值小的個數+1)

1. “I”：將特定 ID 與戰鬥力的寶可夢輸入到當前排行榜中。
2. “D”：從當前排行榜中移除特定 ID 與戰鬥力的寶可夢。
3. “r”：輸出當前排行榜中模式一特定排名的 ID 與戰鬥力。
4. “R”：輸出當前排行榜中模式二特定排名的 ID 與戰鬥力。
5. “v”：輸出當前排行榜中模式一特定戰鬥力數值中的排名。
6. “V”：輸出當前排行榜中模式二特定戰鬥力數值中最高/最低的排名。
7. “K”：輸出與特定戰鬥力數值距離絕對值前 N 個近戰鬥力數值之寶可夢。

補充說明：

K 指令：

假設目前排行榜上有 5 個寶可夢，排名如下：

<ID>	<Mode1 RK>	<Mode2 RK>	<Value>
1	1	1	10000.5
7	1	2	10000.5
3	3	3	9854.7
4	4	4	6453.1
9	5	5	3551.6

收到 K 指令為：

K 9000.0 2，則輸出 K 3 1

K 9000.0 3，則輸出 K 3 1 7

也就是說，當遇到絕對值距離相同時，以 ID 大小來決定誰近(小的比較近)

Input

排行榜中的各指令如下表所示：

I <ID> <value>	// 將 ID 為<ID>、戰鬥力為<value>的寶可夢輸入 // 到排行榜中 (不會輸入已存在排行榜中的 ID)
D <ID> <value>	// 將 ID 為<ID>、戰鬥力為<value>的寶可夢由排 // 行榜中移除 (此寶可夢必存在排行榜中)
r <rank>	// 輸出排名為<rank>的 ID (排名必存在) // 模式一
R <rank>	// 輸出排名為<rank>的 ID (排名必存在) // 模式二
v <value>	// 輸出戰鬥力為<value>的排名 (數值必存在) // 模式一
V <value>	// 輸出戰鬥力為<value>的最高/最低排名 (數值必存在) // 模式二
K<value> <number>	//輸出與特定戰鬥力數值距離前 N 個近戰鬥力 //數值之寶可夢(數值不一定存在)

<ID、rank、number 皆是 int，比 0 大的正整數，ex : 2、7、1490>

<value 則為 float，比 0 大的數值且固定小數點後一位，ex : 500.0、103.7>

在 input file 中，一行代表一個指令，指令由前至後依序執行。Input file

I 5 17.2	//將 ID 為 5 戰鬥力為 17.2 的寶可夢輸入到排行榜中
I 2 8.6	
I 16 12.5	
I 11 4.1	
I 4 4.1	
r 4	//輸出模式一排名第 4 的 ID 與戰鬥力
D 16 12.5	//將 ID 為 16 戰鬥力為 12 的寶可夢從排行榜中移除
R 4	//輸出模式二排名第 4 的 ID 與戰鬥力
I 6 15.9	
I 7 0.1	
v 0.1	//輸出戰鬥力為 0.1 的模式一排名
V 4.1	//輸出戰鬥力為 4.1 最高/最低的模式二排名
V 8.6	//輸出戰鬥力為 8.6 最高/最低的模式二排名
I 16 4.1	
I 17 4.1	
I 18 8.6	
K 4.5 5	//輸出戰鬥力絕對值距離 4.5 前 5 近的寶可夢的 ID

Output

在“r”中，當遇到該 rank 有許多戰鬥力相同的 ID 時，則輸出 ID 最小的寶可夢；
在“K”中，則是以距離由小至大排列；而“V”中，該戰鬥力的寶可夢只有一隻
時，則其輸出之最高/最低排名皆為同一隻寶可夢。

Output file example: (“//” is comment and will not appear in output file)

```
r 4 4.1
R 11 4.1
v 6
V 4 5
V 3 3
K 4 11 16 17 2
```

Compile & Execute

Compile command: `$ g++ -std=c++11 <hw2.cpp> -O3`

Execute command: `$./<execute file> <input file> <output file>`

(Ex : `$./a.out input.txt output.txt`)

Note that input and output file should be the arguments of program. Please make sure your code can be compiled and executed.

Program Submission

1. Please use **C++** language and your program **must be written in only one source file**.
2. Your source file must be named as “**Student_ID_number_hw2.cpp**” and please make sure that all characters of the filename are in **lower case**. For example, if your student number is 0710100, the name of your program file should be “**0710100_hw2.cpp**”.
3. **Do not print** any words on terminal.
4. Upload your **report** and **program** to the new E3 by deadline. (不要壓縮)

未按照繳交格式繳交作業的人會得到 0 分！！！！

Report

1. No more than 3 pages.
2. Your report must contain:
 - a. The flow chart or the pseudo code of you program.
 - b. The experimental results and analysis. (Time complexity)
3. The report file name must be “**Student_ID_number_hw2.pdf**” and please make sure that all characters of the filename are in **lower case**. For example, if your student number is 0710100, the name of your program file should be “**0710100_hw2.pdf**”.

Grading

You need to submit both your source code and report. Remember the submission rules mentioned above, or **you will get zero point!!!!**

- Mode I-D-R-r (×3) 15%
- Mode I-D-V-v (×3) 15%
- Mode I-D-K (×3) 15%
- *** Time limit: 1 minute for small case**
- Big case (1 個) 35%
- Correct answer 15%
- Timing performance (If the answer is correct) 20%
- *** Time limit: 5 minute for big case**
- Report 20%

Due Date

- **Due Date : 2021/4/22 23:55:00**
- **Plagiarism is forbidden.**
- **遲交，第一天分數打 9 折 第二、三天打 8 折 第四、五天打 7 折 第六天打 6 折，之後則不予繳交**