Data Structure and Algorithm Final Project
"The Median Winner"
(**Chinese ver.** is offered 有中文版,見第三頁)

Team 15

Member: B04502036 B03607036 R04943174

Problem definition:

In order to choose a leader to protect the Stark family, the family on the legendary Westeros sends a group of candidates to an unknown island to find an ideal person that combine the power of luckiness and courage through a game called **"The Median Winner"**. The rule is: Everyone will get an ID ranged from **1 to 10^6** after arriving at the island. When the time is out. The player possessing the ID which is the "median" of ID sequence of all players will win the game.

In order to become the winner, the player can (1) Introduce other people (have their own ID) to participate this game to change the median or (2) Attack the player having median ID to let him/her out of the game. Now, **design a data structure** that could support the proceeding of the game with following operations and time complexity **O (log n)**.
(We strongly recommend to use a "modified" heap)

1. ***Insert( )***: Insert a player with some ID (different to existing players.) to join the game.
2. ***Find-Median***: Show the median of the ID sequence. If there are even number of players, output the **"average"** of the 2 median IDs and directly **"Round down"** the decimal part.
3. ***Extract-Median***: Remove the median, which means the median player is out due to attack. If there are even number of players, remove the **"smaller"** median.

(Hint: There are some solutions offered on the Internet, please take use of Google!)

Requirement:

Time limit: **5** seconds.
Memory limit: 256MB.

I/O format:

Input format:

(1) The first line will be the total number of operation (1 < Total operation # < **10^6**)

(2) Number and corresponding operation:

1 : Insert (will followed by a number)
2 : Find-Median
3 : Extract-Median

Ex:

```
13           (13 operations)
1 10         (Insert a player with ID 10)
1 4
1 9
1 7
2            (Output the Median)
1 2
1 5
3            (Remove the median (maybe the smaller if there are 2 medians))
1 15
1 1
2
1 3
2
```

Notice:

(1) We will make sure that the **first 20 operation will be Insert( )**, and you do not need to consider the situation of "0 players" during the game.

(2) The **last operation must be "2"**, you need to output the winner's ID and congrats to him/her. ***(If there are 2 medians, the "smaller" median ID will be the winner.)***

Output format:

(The following code is for reference, follow the format of **"Ex:"** if some difference appears)

2 : Printf ("Current median ID:%d\n", Median_ID)

3 : Printf ("Player ID:%d is out!\n", Out_ID)

Last operation 2 :

Printf ("The winner ID: %d!!!\n", Winner_ID) *Notice the space after ID: for the last line

Ex:

```
Current median ID:8
Player ID:5 is out!
Current median ID:7
The winner ID: 4!!!
```

(You need to congrats the winner in the final round. :))

資料結構與演算法期末專案 - 「中位數之爭」

第15組

組員: B04502036 B03607036 R04943174

題目敘述

傳說中維斯特洛上的史塔克家族為了找出保衛家族的領導人，將一群被選定的候選人送到一座不知名的小島，並透過一個「中位數之爭」的遊戲希望找出兼具運氣與勇氣的最佳人選。遊戲規則為：每個人在抵達小島後將會被分配到一個範圍介於1~**10^6** 的ID，當遊戲時間結束時，留在遊戲中且ID恰為中位數的玩家獲勝。

為了在遊戲中獲勝，玩家可以(1)遊說新玩家加入遊戲獲得新ID以改變中位數，或(2)攻擊ID為中位數的玩家使其出局(不可攻擊其他玩家)。現在**請設計一種資料結構**，可以支援以下三種操作(Operations)，且時間複雜度為O(log n)，以利「中位數之爭」的遊戲進行。**(強烈建議使用一個Heap的「變形」結構來解這個問題)**

遊戲中有三種指令，編號分別為1、2、3，各自支援的操作為：

1. ***Insert( )***: 表示將某ID之玩家加入遊戲資料中，1之後會跟著一個數字(1 ID)表示被加入者的ID (不會和當下在遊戲中的玩家重複)。
2. ***Find-Median( )***: 顯示目前在遊戲中的所有玩家ID序列之中位數，如果玩家人數為偶數(有兩個中位數)，**則顯示這兩個中位數的平均(並無條件捨去小數部分)**，表示該玩家將暴露於被其他玩家攻擊的危險中。
3. ***Extract-Median( )***: 表示ID為中位數的玩家被攻擊而出局，將其從遊戲資料中刪除(若如果玩家人數為偶數，則移除兩個中位數中**較小的那個即可**)

(Hint: 網路上有提供一些思路可以解決這個問題，請善用Google)

測試要求

時間限制: **5** 秒
記憶體限制: 256MB.

輸出輸入格式

讀入格式：

(1) 第一行為執行操作的數目(1 < Total operation # < **10^6**)

(2) 數字對應的操作：

1 : (Insert()) 加入，其後會跟著一數字

2 : (Find-Median) 顯示中位數

3 : (Extract-Median) 刪除中位數

範例：

```
13          (13 次操作)
1 10        (將ID為10的玩家加入)
1 4
1 9
1 7
2           (輸出中位數，如果有兩個，則顯示兩個中位數之平均)
1 2
1 5
3           (刪除中位數，如有兩個，則刪除較小那個)
1 15
1 1
2
1 3
2
```

提醒：我們會確保第1到20個操作是加入玩家(1)，因此你**不需要考慮遊戲中玩家數為0的情形**。另外，**最後一次的操作一定為顯示中位數(2)**，請公布獲勝玩家的ID，並恭喜他/她(若有兩個中位數，**ID較小者為贏家**)

輸出格式：(以下程式碼僅供參考，正確格式請按照範例)

2 : Printf (“Current median ID:%d\n”, Median_ID)

3 : Printf (“Player ID:%d is out!\n”, Out_ID)

最後一個步驟的 2:

Printf (“The winner ID: %d!!!\n”, Winner_ID) *這最後一行ID:後有一個空白

範例：

```
Current median ID:8
Player ID:5 is out!
Current median ID:7
The winner ID: 4!!!
```

(請記得恭喜贏家 :))