Snowman

製作者:高培倫

題目意思

先輸入一個天數，小雪一天都會做一顆雪人，而這雪人會因為溫度而融化，所以我們快速地看看第一個測資

雪人高度 :20 5 7 9 1

溫度 :3 6 4 5 1

所以第一個雪人大小20，經過了5天，大小變成20-3-6-4-5-1 = 1，因為過五天還沒融化，所以輸出-1

第三個雪人大小7，第三天少4，第四天少5，這時侯雪人就消失了，輸出4

思路(1)

第一個最快的想法就是硬爆, 跑一個雪人的for, 每個雪人從當前天數再跑一個溫度的for, 就這麼簡單. 時間複雜度O(n^2)

python code

```
for i in snowman:
    for j in range(0,len(temperature)):
        i-= temperature[j]
        if i <= 0 : print j+1
```

思路(2)

然後恭喜你, 就會發現有幾個紅色的, 上面錯誤訊息寫超時.

所以, 該怎麼做呢？因為這個區塊都是二元搜尋, 所以就往二元搜尋的方向.

二元搜尋最重要的關鍵:資料要排序！

現在有兩個陣列, snowman以及temperature. 誰要排序呢？稍微思考一下.

如果想不出來, 再看一下一張投影片.

思路(3)

大概的方向是這樣子的，對雪人來說，後面的天數只會一直融化.

對第四個雪人與第五個雪人而言，如果他們都能撐到第九十九夜，那他們會有5~99這幾天融化的大小一樣，這樣做了太多重複的事情了.

所以，我們把溫度給累加，3 6 4 5 1 變成 3 9 13 18 19，這樣第一個雪人就只要減

temperature[4]了，20-19 = 1 > 0，我們知道他在給定的天數不會融化，第二層for就不用了，YA！

你可能會問，那第二天做出來的雪人怎麼辦？她各天會融化的大小可不是9 13 18 19. 好，這問題你自己想想看. **第二天後的雪人們在往後的融化大小如何推？**

思路(4)

只要你會上一個問題，就完全理解怎麼找了.

又因為每天的temperature至少有1，所以更改過後的temperature陣列一定會遞增.

所以對temperature做二搜，得到比snowman大一點點或剛好的那個temperature，所對應的key，就是雪人死掉的那一天(QAQ).

注意事項

這題蠻多陷阱的，先預告幾個

1.數量很大，需要long long int

2.二元搜尋給定太多條件反而會錯

大概就這樣了，大家加油吧～