

110 學年度全國資訊學科能力競賽

臺南一中校內複選 試題本

競賽規則

1. 競賽時間：2021/09/24 13:00 ~ 17:00，共 4 小時。
2. 本次競賽試題共 6 題，每題皆有子任務。
3. 為了愛護地球，本次競賽題本僅提供電子檔，不提供紙本。
4. 每題的分數為該題所有子任務得分數加總；單筆子任務得分數為各筆繳交在該筆得到的最大分數。
5. 本次複選比照全國賽不提供記分板。
6. 全部題目的輸入皆為標準輸入。
7. 全部題目的輸出皆為標準輸出。
8. 所有輸入輸出請嚴格遵守題目要求，多或少的換行及空格皆有可能造成裁判系統判斷為答案錯誤。
9. 每題每次上傳間隔為 120 秒，裁判得視情況調整。
10. 所有試題相關問題請於競賽系統中提問，題目相關公告也會公告於競賽系統，請密切注意。
11. 如有電腦問題，請舉手向監考人員反映。
12. 如有如廁需求，須經過監考人員同意方可離場。
13. 不得攜帶任何參考資料，但競賽系統上的參考資料可自行閱讀。
14. 不得自行攜帶隨身碟，如需備份資料，請將資料儲存於電腦 D 槽。
15. 競賽中請勿交談。請勿做出任何會干擾競賽的行為。
16. 如需使用 C++ 的 `std::cin` 或 `std::cout` 可將以下程式碼插入 `main function` 以及將 `endl` 取代為 `'\n'` 來優化輸入輸出速度。唯須注意不可與 `cstdio` 混用。

```
std::ios::sync_with_stdio(false);  
std::cin.tie(nullptr);
```

A. 吉良吉影想平靜地過日子

Problem ID: YoshikageKiraJustWantsToLiveQuietly

Time Limit: 2.0s

Memory Limit: 512MiB



Figure 1: 「在你告訴別人之前，我要先把你收拾掉，讓我今晚也能好好睡一覺。」
出自動畫《JoJo 的奇妙冒險·第四部·不滅鑽石》

杜王町存在一位名為 吉良 的危險人物。在他遭遇到 重清 識破他的身分時，他決定使用他的特殊能力「殺手皇后」消滅對方，因而與重清的「收成者」對峙。「收成者」有如一群「收成者」，他們會一擁而上攻擊目標敵人；吉良的能力「殺手皇后」可以將任何物體變為炸彈，而他決定將一隻隻「收成者」變成炸彈，再引發爆炸消滅他們。

雖然「收成者」會一擁而上，但因為秩序零落、完全不同步，吉良將他們視為一隻隻依序攻擊。吉良會持續引爆多隻「收成者」：他會選擇**最少一隻、最多全部**先湧上的前幾隻一次引爆，未被引爆的再重複選擇前幾隻、一次引爆，並一次次重複選擇、引爆，直到全部「收成者」都被引爆而消滅殆盡。不過，在每次選擇之前，他必須先考慮爆炸的**強度與範圍**。

每隻「收成者」都有自己的耐久度 w_i ，而爆炸的**強度**也隨之影響：當吉良選擇引爆一群「收成者」時，引爆的強度會**恰好**為該群「收成者」的耐久度總和（畢竟過強會炸傷自己、過弱又不能確實消滅對手）。雖然吉良每次都可以製造任意強度的爆炸，但是因為有著完美一致性的強迫症，所以他限制自己**每次爆炸的強度都必須相同**。

爆炸的**範圍**等同於引爆的「收成者」的數量：單次爆炸中的每一隻都需要一單位範圍，而多次爆炸的範圍等同於其中**範圍最大的爆炸**。吉良可以一次引爆所有「收成者」，但是範圍越大就越容易被旁人發現，因此他希望整體的爆炸範圍越小越好。

找出最小的範圍 k ，使得吉良有辦法在接連爆炸下消滅所有「收成者」，且每次爆炸的強度相同、整體範圍在 k 以下。

— 輸入 —

第一行有一個數字 n ，代表收成者的數目。

第二行有 n 個數字，依序代表湧上的「收成者」的耐久度 w 。

— 輸出 —

輸出最小的範圍 k 。

— 輸入限制 —

- $1 \leq n \leq 8000$
- $1 \leq w_i \leq 10^9$

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	20	$n \leq 20$
3	20	$n \leq 500$
4	20	$\sum w_i \leq 10^6$
5	20	$n \leq 800$
6	20	沒有其他限制

— 範例輸入 1 —

5
1 2 2 1 3

— 範例輸出 1 —

2

— 範例說明 1 —

吉良引爆了 3 次，每次引爆的「收成者」的耐久度依序為 $\langle 1, 2 \rangle$, $\langle 2, 1 \rangle$, $\langle 3 \rangle$ 。爆炸強度為 3，需要 2 單位爆炸範圍。

— 範例輸入 2 —

9
8 3 3 5 2 1 5 3 3

— 範例輸出 2 —

4

— 範例說明 2 —

吉良引爆了 3 次，每次引爆的「收成者」的耐久度依序為 $\langle 8, 3 \rangle$, $\langle 3, 5, 2, 1 \rangle$, $\langle 5, 3, 3 \rangle$ 。爆炸強度為 11，需要 4 單位爆炸範圍。

— 範例輸入 3 —

5
1 2 4 8 16

— 範例輸出 3 —

5

— 範例說明 3 —

吉良引爆了 1 次，引爆的「收成者」的耐久度依序為 $\langle 1, 2, 4, 8, 16 \rangle$ 。爆炸強度為 31，需要 5 單位爆炸範圍。

－ 範例輸入 4 －

7

999999999 999999999 999999999 999999998 999999997 999999996 6

－ 範例輸出 4 －

4

－ 範例說明 4 －

吉良引爆了 2 次，每次引爆的「收成者」的耐久度依序為 $\langle 999999999, 999999999, 999999999 \rangle$, $\langle 999999998, 999999997, 999999996, 6 \rangle$ 。爆炸強度為 2999999997，需要 4 單位爆炸範圍。

B. 學測分發

Problem ID: Kyoya

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 橋場恭也收到兩間學校的錄取通知書，出自動畫《我們的重製人生》

又來到了高中學測的季節，而今天是放榜的日子，橋場恭也這次回到了十年前，重新選擇了大中藝術大學。

不過說到放榜，首先就要了解學校分發的方式。

每個學生經過考試後都會得到一個總成績，並且可以填寫六個志願序。填寫完志願序後，每間學校會得到一份學生的清單，接下來將學生依照成績由高至低排序。只要還在名額內的學生都一定可以錄取，剩餘的學生則列為備取，當正取生放棄該名額，則會依序由備取名額補足。

假如每間學校都只能收 k 位學生，告訴你 n 個學生填寫的志願序，以及每間學校每個申請者成績由高到低的排名，並且假設每個學生都會盡可能選擇志願序越前面的學校，你能知道這 m 間學校的最終錄取人數以及錄取的學生編號分別為何嗎？

— 輸入 —

第一行有三個整數 n, m, k ，代表有 n 位學生， m 間學校，每間學校收 k 個人。

接下來有 n 行，每行有六個數字，第 i 行代表編號為 i 的學生選填的六個志願。

接下來有 m 行，第 i 行的第一個數字 a_i 代表選填第 i 間學校的人數，接下來的 a_i 個數字代表申請者成績由高到低的排名。

— 輸出 —

輸出 m 行，每行的第一個數字 b_i 代表進入第 i 間學校的人數，接下來輸出 b_i 個數字，代表進入第 i 間學校的學生編號，編號請由小到大排序。

— 輸入限制 —

- $1 \leq n \leq 100000$
- $6 \leq m \leq 1000$
- $1 \leq k \leq N$
- 保證每個選填一間學校的學生都會出現在該校的排名結果中，並且沒填該校的不會出現。

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	25	若對於大學 a 來說，第 i 人排名在第 j 人之前，則對於 i 與 j 同時申請的所有大學來說，第 i 人排名在第 j 人之前。
3	30	$n \leq 8$
4	30	$n \leq 10^3$
5	50	無額外限制

— 範例輸入 1 —

```
3 7 2
1 2 3 7 5 6
1 3 7 4 6 5
1 7 4 6 2 3
3 2 1 3
2 1 3
3 3 2 1
2 2 3
2 1 2
3 2 3 1
3 2 3 1
```

— 範例輸出 1 —

```
2 1 2
0
0
0
0
0
1 3
```

— 範例輸入 2 —

```
7 6 1
1 3 2 4 6 5
1 2 3 4 5 6
1 4 2 3 5 6
1 6 5 4 3 2
2 3 1 5 4 2
6 5 4 3 2 1
3 2 4 5 6 1
7 7 5 4 3 1 6 2
7 7 4 3 5 1 6 2
7 6 1 4 7 5 3 2
7 7 6 4 3 1 5 2
7 7 5 3 4 1 6 2
7 1 3 4 5 7 6 2
```


－ 範例輸出 2 －

1 5

1 7

1 1

1 3

1 6

1 4

C. 為美好的地牢獻上爆擊

Problem ID: Kazuma

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 和真正在向其他人炫耀自己獲得的新技能，出自動畫《為美好的世界獻上祝福》

在某次的公會任務當中，和真意外獲得了新的被動技能：ultimate boss 3liminator - 3，簡稱 ub33。顧名思義，這是一種用來消滅魔物的強力技能，而且還不需要消耗魔力。詳細來說，這個技能會在擁有者遇到魔物後強制進入戰鬥，在受到魔物的攻擊後，無論是否受到傷害，下一次我方攻擊時必定會消滅對手。雖然是個像是 bug 的技能，不過因為會強制加入戰鬥，還需要先吃一記傷害後才能發動，要是沒掌握好魔物出現的位置與自己的血量，那可能很快就會上西天了。

雖然聽起來很危險，不過和真已經迫不及待想要到地牢當中試試 ub33 的效果了！地牢是一個 $n \times m$ 的地圖，左上角與右下角座標分別為 $(1, 1)$ 與 (n, m) 。有許多的魔物散布在地牢中，每個魔物的攻擊力為 w_{ij} ，每次我們可以選擇上/下/左/右其中一個方向前進一格。值得一提的是，在進到地牢的那一刻戰鬥就開始了，也就是說，如果起點就存在魔物，那就會直接進入戰鬥。

在地牢當中可能會有稀有的道具 Amazing Damage Decreaser，簡稱 ADD，有了 ADD，每次遇到魔物時受到的傷害都會減少 w_{ij} 點，不過最多就只是不會受到傷害，並沒有回血的功能。而因為 ADD 十分稀有，因此在地牢當中最多就只會出現一個。

和真想知道，在走到地牢出口之前他最少承受的傷害是多少，要是因為太過興奮就被幹掉，那就糟糕了。你能告訴和真承受的傷害最少是多少嗎？

— 輸入 —

輸入第一行包含兩個數字 n, m 表示地圖大小。輸入第二行包含兩個數字 s_x, s_y 表示起點座標。輸入第三行包含兩個數字 e_x, e_y 表示終點座標。接下來會有 n 行輸入，每行 m 個數字，表示魔物攻擊力大小 w_{ij} 。如果 $w_{ij} < 0$ 則表示稀有道具 ADD，其數值表示 ADD 減少的傷害量。

— 輸出 —

輸出一行數字，表示和真在地牢中受到的最少傷害值。

— 輸入限制 —

- $1 \leq n, m \leq 10^3$
- $1 \leq s_x, e_x \leq n$
- $1 \leq s_y, e_y \leq m$
- $-10^9 \leq w_{ij} \leq 10^9$

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	20	沒有 ADD
3	30	除了 ADD 以外所有 w 相等
4	50	$n, m \leq 4$
5	50	無額外限制

— 範例輸入 1 —

3 4
1 1
3 4
6 8 3 5
5 1 6 3
9 4 1 7

— 範例輸出 1 —

24

— 範例輸入 2 —

3 4
1 1
3 4
6 8 -3 5
5 1 6 3
9 4 1 7

— 範例輸出 2 —

19

D. 本田小狼

Problem ID: Cub

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 小熊正騎著本田小狼 50 兜風，出自動畫《本田小狼與我》

今天小熊騎著本田小狼 50 兜風，剛好到了一個絕佳的制高點，在這裡整個小鎮一覽無遺。距離黃昏還有一段時間，因此小熊打算再到附近晃晃，考慮到小狼剩餘的油量，因此能去的地方是有限的。

如果將所有的景點、店家等等都當作是一個節點，並且依序將節點從 $1 \sim n$ 編號，制高點編號為 1。接著依序把所有節點與未連接過的節點中，高度比該節點低，且距離最近的節點連上一條邊。並且保證每個節點都一定可以透過邊連接到。如此一來就可以更方便規劃接下來的行程了。

小熊希望接下來都是往山下前進，所以絕對不會往距離起點較近的節點前進，並且同一個節點也不會重複經過。假設走過一條邊需要花費 1 單位的油量，小熊想知道如果她現在在節點 p ， k 單位的油量可以到達的節點包含 p 總共有多少個呢？

— 輸入 —

輸入第一行有一個數字 n ，表示包含起點有 n 個節點。

接下來有 $n - 1$ 行，每行兩個數字，表示兩個節點之間有一條邊。

接下來有一個數字 q ，表示有 q 次詢問。

接下來有 q 行，每行包含兩個數字 p, k ，表示詢問從節點 p 開始，並且有 k 單位的油量。

— 輸出 —

輸出 q 行，每行一個數字，表示從 p 點開始， k 單位的油量可以到達的節點總數。

— 輸入限制 —

- $1 \leq n, q \leq 10^6$
- $1 \leq p \leq n$
- $0 \leq k \leq n$

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	20	$n, q \leq 5000$
3	30	$n, q \leq 10^5, k \leq 100$
4	50	所有 k 相同
5	50	$n, q \leq 10^5$ 且與制高點距離相同的節點不超過 50 個
6	50	$n, q \leq 10^5$
7	50	無額外限制

— 範例輸入 1 —

3
1 2
1 3
2
1 0
1 1

— 範例輸出 1 —

1
3

— 範例說明 1 —

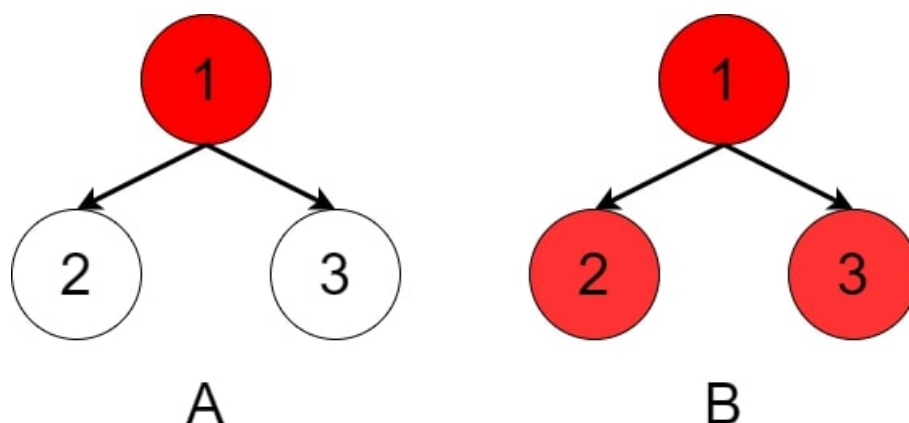


Figure 2: 範例說明 1

- 第一筆輸出如圖 A，從編號 1 的節點消耗 0 個單位的油量只能停留在原點，因此能到達的節點總數為 1。
- 第二筆輸出如圖 B，從編號 1 的節點出發，消耗 0 個單位的油量可以到達節點 1；消耗 1 單位的油量可以到達節點 2、3。因此能到達的節點總數為 3。

— 範例輸入 2 —

```

7
1 2
1 3
2 4
2 5
3 6
4 7
3
1 2
3 3
2 1

```

— 範例輸出 2 —

```

6
2
3

```

— 範例說明 2 —

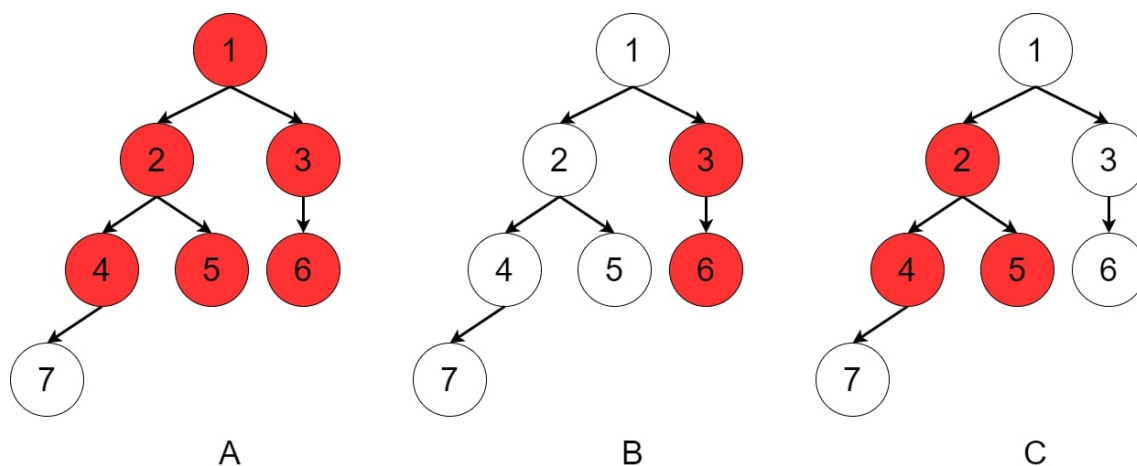


Figure 3: 範例說明 2

- 第一筆輸出如圖 A，從編號 1 的節點出發，節點 7 需要 3 單位油量才能到達，其餘節點皆能在 2 單位以內的油量到達。因此能到達的節點總數為 6。
- 第二筆輸出如圖 B，從編號 3 的節點出發，消耗 0 個單位的油量可以到達節點 3; 消耗 1 單位的油量可以到達節點 6。因此能到達的節點總數為 2。
- 第二筆輸出如圖 C，從編號 2 的節點出發，消耗 0 個單位的油量可以到達節點 2; 消耗 1 單位的油量可以到達節點 4、5。因此能到達的節點總數為 3。

E. 遞迴呀遞迴

Problem ID: Rerecursion

Time Limit: 1.0s

Memory Limit: 512MiB

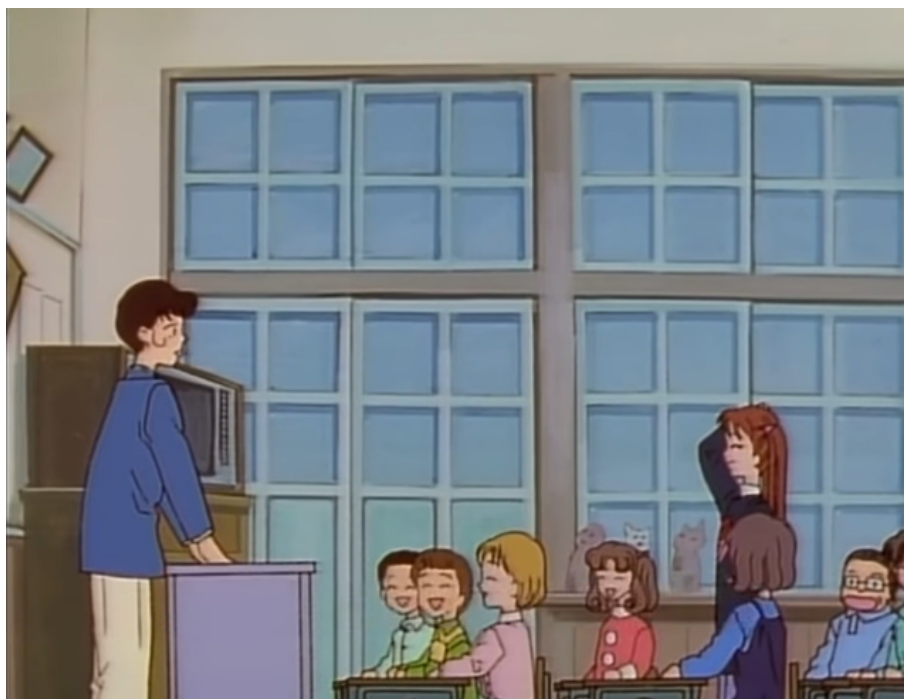


Figure 1: 倉田紗南參加節目「善次郎的玩偶遊戲」的錄影，出自動畫《玩偶遊戲》

「這裡好熱啊～」學生 A 抱怨道。

「提醒大家夏天到了要多補充水分唷。」善次郎老師乘機宣導大家。

「用什麼方式最能降暑呢？大家來分享看看。」善次郎緊接著開啟新的話題。

「舉例來說，老師我最喜歡冰淇淋了。」見大家都沒有回應，善次郎先發表自己的意見。

「正好暑假要到了，我們家要去海邊玩。」學生 B 終於開心地分享。

「快點到啊，真希望暑假快點到。」學生 C 喃喃自語。

「素麵！我們家今天中午吃冰冰涼涼的素麵。」學生 D 繼續分享家裡應對酷暑的方式。

「蜜茶！我媽都會泡我最喜歡的蜜茶給我喝。」學生 E 也分享自己喜愛的食物。

善次郎老師這時才注意到有一個人還沒來。

「先等一下，今天紗南好像還沒到啊？」善次郎問大家。

「抱歉，我遲到了！」紗南衝了進來。

「怎麼了，紗南，妳又睡過頭了嗎？」善次郎問。

「其實是因為我搭的電車突然爆胎了。哈哈哈哈哈～」紗南回應。

「換輪胎花了不少時間是嗎？」善次郎隨著紗南說笑。

「電車哪來的輪胎，少來。」緊接著一個吐槽。

「作為妳遲到的懲罰，這裡有一個問題要問妳。」善次郎說。

給定 a, b, c, r_0, r_1, n, p 和遞迴關係式 $r_{i+2} = a \cdot r_{i+1} \cdot r_i + b \cdot r_{i+1} + c \cdot r_i$ ($i \geq 0$)。
求 $r_n \pmod{p}$ 。

— 輸入 —

第一行有三個數字 a, b, c 。

第二行有二個數字 r_0, r_1 。

第三行有二個數字 n, p 。

— 輸出 —

輸出 $r_n \pmod{p}$ 。

— 輸入範圍 —

- $-10^9 \leq a, b, c, r_0, r_1 \leq 10^9$
- $1 \leq n \leq 10^{18}$
- $1 \leq p \leq 10^4$

— 子任務 —

編號	分數	額外限制
0	0	範例輸入輸出
1	4	$n \leq 10^6$
2	1	$a = c = 0$
3	4	$a = b = 0$
4	4	$a = 0$
5	8	$a = 1, b = c = 0$
6	6	$b = c = 0$
7	5	$p = 2$
8	3	無任何限制

－ 範例輸入 1 －

1 5 -1

0 -1

9 21

－ 範例輸出 1 －

12

－ 範例輸入 2 －

3 -3 5

6 6

10 72

－ 範例輸出 2 －

30

－ 範例輸入 3 －

-3 -5 2

9 9

4 35

－ 範例輸出 3 －

20

F. Hello, Weird Music Sheet!

Problem ID: HelloWeirdMusicSheet

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: Hello, Happy World! 團體照片，出自遊戲《BanG Dream! 少女樂團派對》

Hello, Happy World! 在弦卷心的帶領下，總是有很多不可思議的事情會發生。這次心突發奇想，想透過電腦創造一種符合特殊規則的樂譜。

規則是這樣的，首先我們先決定好整個樂譜使用到的音符數量，並且選定其中一個八度，接下來樂譜當中所有使用到的音符都會在選定的八度內。音符的音名依照音高由低到高依序為 C、C#、D、D#、E、F、F#、G、G#、A、A#、B。第 i 個音符的音高表示為 a_i ，那麼樂譜當中的所有音符需要符合 $a_1 \leq a_2 \geq a_3 \leq a_4 \geq a_5 \dots$ ，並且 $0 \leq a_i \leq 11$ ($1 \leq i \leq n$)。

現在給你一個樂譜音符數量 n ，你能知道有多少種符合這個規則的樂譜嗎？答案可能會很大，因此只需要輸出答案 mod 1000000007 的結果即可。

— 輸入 —

輸入只有一行，一個正整數 n 表示樂譜當中音符的總數。

— 輸出 —

輸出一個整數，表示所有可能的樂譜數量 mod 1000000007 後的結果。

— 輸入限制 —

- $1 \leq n \leq 10^{18}$

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	20	$n \leq 4$
3	30	$n \leq 10^4$
4	30	$n \leq 10^6$
5	30	$n \leq 10^9$
6	50	無額外限制

— 範例輸入 1 —

2

— 範例輸出 1 —

78

— 範例輸入 2 —

3

— 範例輸出 2 —

650

— 範例輸入 3 —

198964

— 範例輸出 3 —

558701749