

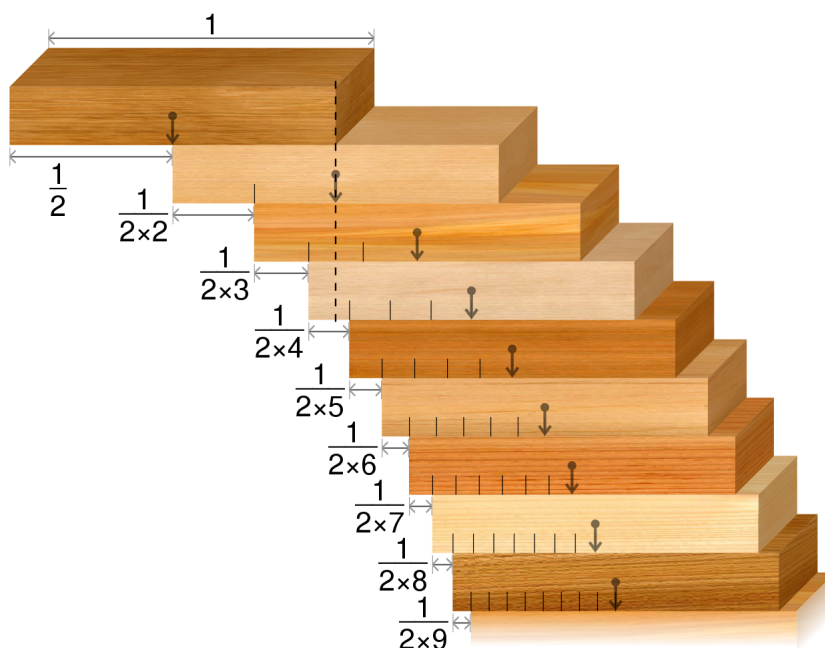
A. 方塊王 (Tower)

問題敘述

物理有一個有名的積木堆積問題，敘述如下：

「有 n 個等長等重的均勻長方體方塊，如果把他們一個個堆疊在桌面上，最遠可以伸出桌子邊緣多遠？」

事實上，這個最遠的距離恰好是 $\sum_{i=1}^n \frac{1}{2i}$ ，方塊數量夠多的話可以伸長無窮遠。可以參考下面附圖：



(圖片取自維基百科，由 cmglee, Anonimski 上傳，Block stacking problem.svg，以創用 CC 姓名標示-相同方式分享 4.0 國際授權條款釋出。)

方塊王覺得這題太水了，於是拿起了手邊等長但是不等重的方塊，想要知道最遠可以伸出桌子多遠。

具體來說，有 n 個長度為 L 的均勻長方體方塊，第 i 個方塊重量為 w_i 。每個方塊必須疊在另外一個方塊或桌面上，且桌面與每個方塊正上方至多放置一個方塊。你可以用任意順序疊方塊，而每一個方塊長度為 L 的邊的方向皆需要與桌緣垂直。桌子可以想像成一個長寬皆無限長的平面。

假設最上面 x 個方塊的重心不在由上往下數第 $x+1$ 個方塊上方，那這些方塊就會傾倒，也就是說如果第 $x+1$ 個方塊的左側座標是 0，從上往下數第 j 個方塊重心位於 p'_j 、重量是 w'_j ，則須符合

$$0 \leq \frac{\sum_{j=1}^x p'_j w'_j}{\sum_{j=1}^x w'_j} \leq L$$

當然，所有方塊合起來的重心也要在桌面之上。假設桌子的左端座標為 0，則方塊往左疊的最左端的位置距離桌邊最遠的距離為何？輸出的答案若與正確答案的絕對或相對誤差在 10^{-9} 以內即視為正確。

輸入格式

$$n \ L$$

$$w_1 \ w_2 \ \dots \ w_n$$

- n 代表方塊的數量。
- L 代表每個方塊的長度。
- w_i 為方塊 i 的重量。

輸出格式

$$ans$$

- ans 為方塊塔的端點和桌面的最遠距離。答案的絕對或相對誤差在 10^{-9} 以內即算為正確。

測資限制

- $1 \leq N \leq 3 \times 10^5$
- $1 \leq L \leq 10^9$
- $1 \leq w_i \leq 10^9$
- 輸入皆為整數

範例測試

Sample Input	Sample Output
1 1 1	0.5
3 12 1 2 3	13

評分說明

本題共有 3 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	25	$n \leq 9$
2	25	$w_i = 1, \forall 1 \leq i \leq n$
3	50	無額外限制

B. 更加 Trivial 的題目 (Quadrivial)

問題敘述

喵喵是一位熱愛演算法的競程選手，不過現在的他也正為了出題而煩惱著。

「到底要如何出一道不 trivial 的題目呢？」日復一日，喵喵困擾著。

於是，喵喵跑去到處翻論文，學了很多新科技，也在跟其他人聊天的時候聽到各種神奇的演算法們（像是 $O(n+m)$ 的四連通分量）。但就算是這樣，喵喵也還是不知道要怎麼生出不 trivial 的題目。

在死線的前一天，一個月黑風高的夜晚，看著可愛的拉菲，喵喵突然靈光一閃：為什麼不把自己出題的歷程寫成題目呢？

喵喵跟拉菲合力將想過的每一道題目、遇到的每一個科技都變成一個個的整數序列。一次的思考是將這些序列們以任意順序排列並拼接起來；一次的出題則是在這拼接起來的序列中尋找最大的非空區間和。喵喵想知道的當然就是在所有思考中能生出最好的題目有多好。

正式地說，喵喵跟拉菲想過的題目或遇到的科技總共有 n 個，其中第 i 個可以被視為一個序列 $a_{i,1}, a_{i,2}, \dots, a_{i,k_i}$ ，喵喵想要決定一個 1 到 n 整數的排列 p_1, p_2, \dots, p_N ，並把第 p_1, p_2, \dots, p_N 個序列依序接起來後，從新的序列選出一個非空的連續區間，他想知道這個區間裡的數字總和最大可以是多少。

喵喵馬上就發現了，這道由許多 trivial 的題目跟科技所拼湊而成的題目並沒有讓這題變得困難，反而是更加的 trivial。但由於為了出這題已經花掉了僅剩的時間，最後喵喵也只能飲恨吞敗。對了，既然這題變得更加 trivial 了，不如就叫它“A Quadrivial Problem”吧！

輸入格式

n
$k_1 \ a_{1,1} \ a_{1,2} \ \cdots \ a_{1,k_1}$
$k_2 \ a_{2,1} \ a_{2,2} \ \cdots \ a_{2,k_2}$
\vdots
$k_n \ a_{n,1} \ a_{n,2} \ \cdots \ a_{n,k_n}$

- n 代表整數序列的數量。
- k_i 代表第 i 個整數序列的大小 ($1 \leq i \leq n$)。
- $a_{i,j}$ 代表第 i 個整數序列的第 j 個元素 ($1 \leq i \leq n, 1 \leq j \leq k_i$)。

輸出格式

ans

- ans 代表在所有思考中能生出最好的題目有多好，也就是將這 n 個整數序列任意拼接後可以達到的最大

非空區間和。

測資限制

- $1 \leq n \leq 10^5$
- $1 \leq k_i \leq 10^5$ ($1 \leq i \leq n$)
- 所有 n 個整數序列的長度之和 $\sum_{i=1}^n k_i \leq 10^6$
- $|a_{i,j}| \leq 10^9$ ($1 \leq i \leq n, 1 \leq j \leq k_i$)
- 輸入皆為整數

範例測試

Sample Input	Sample Output
4 3 1 4 -6 2 -1 -3 6 6 -11 5 -13 3 12 4 14 -7 -3 15	39
1 1 -1000000000	-1000000000

評分說明

本題共有 4 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	8	$n = 1$
2	14	$n \leq 400$
3	22	$n \leq 3000$
4	56	無額外限制

C. 頒獎音樂 (Ceremony)

問題敘述

經過了多年的努力，你總算在全國資訊學科能力競賽拿到第一名。滿心期待上台領獎的你，在頒獎開始之前路過頒獎場地，聽見裡面傳來測試頒獎音樂的聲音。「這也太難聽了吧……」你認為這個音樂一點也不適合頒獎。好在身為演算法專家、音樂天才又是資安超人的你，可以輕而易舉的竄改頒獎音樂。

頒獎音樂由 N 個音符組成，第 i 個音符的音高是 a_i 。你可以修改這個音樂任意次，但為了不要被發現，你在一次修改中，只能選擇一個滿足第一個音符和最後一個音符音高相同的區間，並將這個區間反轉。正式地說，每次修改你會選擇兩個數字 $1 \leq l \leq r \leq N$ ，滿足 $a_l = a_r$ ，並將 a_l, a_{l+1}, \dots, a_r 反轉變成 a_r, a_{r-1}, \dots, a_l 。

你認為「低高低」或「高低高」的旋律是不好聽的，因此你要盡可能減少這樣的旋律。換句話說，你希望在做完所有操作以後，滿足 $a_{i-1} < a_i > a_{i+1}$ 或 $a_{i-1} > a_i < a_{i+1}$ 的 i ($1 < i < N$) 數量盡可能少。

輸入格式

$$N$$

$$a_1 \ a_2 \ \dots \ a_N$$

- N 表示頒獎音樂中音符的數量。
- a_i 為本來的頒獎音樂中的第 i 個音符音高。

輸出格式

$$b_1 \ b_2 \ \dots \ b_N$$

- b_i 為你修改過後的音樂中，第 i 個音符的音高。
- 若有多種可能的解，輸出任意一種。

測資限制

- $3 \leq N \leq 10^6$
- $1 \leq a_i \leq N$
- 輸入皆為整數

範例測試

Sample Input	Sample Output
10 2 2 6 2 1 3 6 4 1 4	2 6 4 1 2 2 6 3 1 4
10 2 5 5 5 10 2 3 3 10 2	2 10 5 5 5 2 3 3 10 2
10 8 4 9 4 6 5 8 2 4 8	8 2 4 9 4 6 5 8 4 8
7 2 3 2 1 4 3 4	2 3 4 3 2 1 4

評分說明

本題共有 5 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	9	$N \leq 10$
2	20	$a_i \leq 2$
3	17	$N \leq 1000$
4	21	$N \leq 10^5$
5	33	無額外限制

D. 小風的遊戲 (Game)

問題敘述

小風最近受邀幫 NHSPCCamp 設計一個遊戲，遊戲會由 N 座城市和 M 條連接城市的雙向道路組成，其中任何城市都能經由若干個道路走到另一座城市。每個城市由 1 到 N 編號。一開始玩家在城市 S ，終點是城市 T ，玩家的目標是要從 S 經過若干條道路走到 T 。

每條道路上都會有一隻怪物，每隻怪物的血量都是一個不超過 10^{12} 的正整數，玩家必須要斬殺道路上的怪物才能走到道路另一端的城市，而斬殺的怪物會在他成功通過道路後在同一個道路上重生，對於一隻血量為 x 的怪物，玩家需要花恰好 x 秒的時間斬殺他，因此可以視通過一條道路的時間為斬殺怪物所需的時間。

因為小風不希望他的遊戲太早被破關，但又不希望太過於煩躁讓大家都不想玩，因此他希望存在一種通過道路的方法使得總耗時恰好為 D 秒，而且不存在任何花不到 D 秒就能從城市 S 走到城市 T 的方法。

小風早就對每隻怪物的強弱有些許規劃，更明確的說，假設第 i 條道路連接城市 u_i, v_i 且路上的怪物血量為 h_i ，小風心中有個排列 p_1, p_2, \dots, p_M ，代表小風希望怪物的血量必須滿足 $h_{p_1} < h_{p_2} < \dots < h_{p_M}$ 。

因為他接下來很久都沒空處理這個遊戲，不過他除了怪物的血量還沒決定以外，所有的參數都已經設定好了，因此他希望你先告訴他在目前有的條件下是否能構造出滿足所有條件的 h_1, h_2, \dots, h_M ，而且如果構造得出來，他也希望你幫他構造出一組合法的怪獸血量序列。

輸入格式

```

N M S T D
p_1 p_2 \dots p_M
u_1 v_1
u_2 v_2
\vdots
u_M v_M

```

- N, M 分別表示城市和道路的數量。
- S, T 分別表示玩家的起始城市和終點城市。
- D 的意義如題敘所述。
- p_i 代表 h_{p_i} 是所有血量中第 i 小的。
- u_i, v_i 表示第 i 條道路連接的兩座城市。

輸出格式

```

ans
h_1 h_2 \dots h_M

```

- ans 為 YES 或 NO，若存在一組能滿足所有條件的 h_1, h_2, \dots, h_M ，則 ans 為 YES，否則為 NO。
- 如果 ans 為 YES 才需要輸出第二行。
- $1 \leq h_i \leq 10^{12}$ 。
- h_i 必須是正整數。
- h_1, h_2, \dots, h_M 為滿足所有條件的一組合法怪物血量序列，如果存在多組合法的序列，輸出任意一組即可。若你輸出的序列不滿足此條件，仍可以獲得 50% 的分數，詳見評分說明。

測資限制

- $2 \leq N \leq 10^5$
- $N - 1 \leq M \leq \min\{2 \times 10^5, \frac{N(N-1)}{2}\}$
- $1 \leq S, T \leq N$
- $S \neq T$
- $1 \leq D \leq 10^{11}$
- $1 \leq p_i \leq M$
- $1 \leq u_i, v_i \leq N$
- $u_i \neq v_i$
- 保證任兩條道路不會連接完全相同的城市
- 保證任何城市都能經由若干條道路走到另一座城市
- 保證 p_1, p_2, \dots, p_M 兩兩相異
- 所有輸入皆為整數

範例測試

Sample Input	Sample Output
4 5 1 4 10 5 1 4 3 2 1 3 1 2 2 3 3 4 2 4	YES 4 8 7 6 3
5 4 3 4 1 1 2 3 4 3 1 1 2 2 5 5 4	NO

評分說明

本題共有 4 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	5	$M = N - 1$
2	9	$M = N$ 且每座城市都恰有兩條道路通往另外兩座城市
3	23	$N \leq 3000, M \leq 5000$
4	63	無額外限制

對於每一組子任務，若你於所有測試資料都有正確判斷是否存在能滿足題目要求的 h_1, h_2, \dots, h_M ，且你輸出的 h_1, h_2, \dots, h_M 滿足 $1 \leq h_i \leq 10^{12}$ 且 h_i 都是正整數，但不是滿足所有條件的一組合法怪物血量序列，你可以獲得該子任務 50% 的分數。

E. 好多個三口羊 (Escape)

問題敘述

鹿大資工系館地下室是一個神奇的地方，裡面有各個年級的休息區、新館自習室、麻將間等設施。系館地下室也常駐著一個可怕的生物：三口羊。三口羊是一個喜歡 merge 別人的怪人，只要看到可憐的學弟就會開始往他的方向走，如果被他撞上的話就鐵定會被 merge！最近發生了一個更可怕的事情，三口羊開始增生了！三口羊們突然一個個開始在不同位置出現，每一隻都有 merge 人的共同目標。

具體來說，系館地下室可以被看成一個 $C \times C$ 的平面，接下來依序會有 n 隻三口羊出現，第 i 隻的座標會在格子點 (x_i, y_i) ，保證 $0 \leq x_i, y_i \leq C$ 。

8e7 想要躲避三口羊的 merge 攻擊，具體來說，一開始 **8e7** 可能會在 $C \times C$ 平面裡的格子點 (x, y) ，接著所有的三口羊都會看到 **8e7** 並同時試著 merge 他。他每次可以沿著格線上下左右移動一格（或是不動）。而他移動完之後所有三口羊也可以移動一格（或是不動）。每一隻三口羊都可以獨立移動，且所有三口羊會依照最佳策略移動。只要有任意一隻三口羊的位置在某個時間與 **8e7** 重疊他就被 merge 了（注意，如果一開始的位置就有三口羊的話那就算被 merge）。如果 **8e7** 可以離開 $C \times C$ 的範圍，代表他（暫時）是安全的。

對於所有的 i ，**8e7** 想知道前 i 隻三口羊出現後有多少個（在平面內）起始位置會被 merge。請你寫一個程式救救 **8e7**（拜託了他很需要）。

輸入格式

```

n C
x1 y1
x2 y2
⋮
xn yn

```

- n 表示三口羊的數量。
- C 表示系館的邊長。
- x_i, y_i 表示第 i 隻三口羊的位置。

輸出格式

```

ans1
ans2
⋮
ansn

```

- ans_i 為第 i 隻三口羊出現之後，會被 merge 的起始位置數。

測資限制

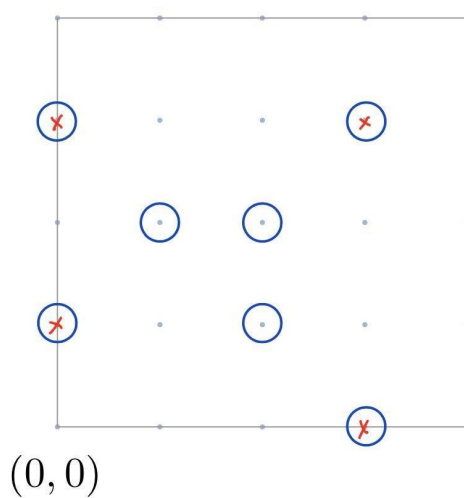
- $1 \leq n, C \leq 2 \times 10^5$
- $0 \leq x_i, y_i \leq C, \forall 1 \leq i \leq N$
- 所有輸入皆為整數

範例測試

Sample Input	Sample Output
4 4 0 3 0 1 3 0 3 3	1 2 5 7
5 7 0 0 1 0 1 5 1 6 3 2	1 2 3 4 8

範例測試說明

下圖是範例測資 1 在三口羊都出現過後的圖示。



紅色的'X'代表三口羊的位置，藍色圓圈代表一個會被 merge 的起始點。

評分說明

本題共有 5 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	7	$n, C \leq 30$
2	10	$n \leq 5 \times 10^4, C \leq 1000$
3	14	$n \leq 1000$
4	31	$n \leq 5 \times 10^4$
5	38	無額外限制



F. LCA 遊戲 (LCA)

問題敘述

Alice 跟 Bob 常常在很大的資料結構或是圖上面玩各種無聊的遊戲。今天 Alice 給了 Bob 一棵樹跟樹上一個點 A ，Alice 心裡會想著一個點 B ，Bob 的目標是找出點 B 。樹上有 N 個點，編號由 1 到 N 。

每一次詢問，Bob 可以選擇一個點 C ，Alice 會告訴 Bob：如果把這棵樹的根定為 C ， A 和 B 的 LCA (最低共同祖先) 的編號。

Alice 會盡可能的讓 Bob 花更多次的詢問，因此 Alice 可能會中途改變心裡 B 的值，只要這個新的 B 點和前面所回答的答案皆無矛盾。Bob 也同樣絕頂聰明，他會找到最佳策略，最小化自己需要的最大詢問數。

請幫助 Bob 計算他至少需要問幾次才能找出點 B 。

輸入格式

```

N A
u1 v1
u2 v2
⋮
uN-1 vN-1

```

- N 表示節點的數量。
- A 為點 A 的編號。
- u_i, v_i 表示第 i 條邊連接的節點。

輸出格式

```
ans
```

- ans 為 Bob 最多需要問幾次才能找出點 B 。

測資限制

- $1 \leq N \leq 10^6$
- $1 \leq u_i, v_i, A \leq N$
- 輸入的圖保證是一棵樹
- 所有輸入皆為整數

範例測試

Sample Input	Sample Output
5 1 1 2 2 3 3 4 4 5	1
5 3 1 2 2 3 3 4 4 5	2
7 1 1 2 1 3 2 4 2 5 3 6 3 7	3

評分說明

本題共有 2 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	29	存在一個節點 v ，使得以節點 v 為根時，輸入是一棵二元樹
2	71	無額外限制

G. 幸運數字 (Number)

問題敘述

小 B 是一位熱愛旅遊的天才，最近他決定不顧女友反對前往各個國家旅遊！

小 B 打算前往 Q 個很特別的國家，其中第 i 個國家的貨幣只有 $1, b_i, b_i^2, b_i^3, \dots$ 等 b_i 的非負整數冪次的幣值。在這些國家買東西時，為了加速付款的速度，小 B 總是會使用盡量少的貨幣數量進行付費。

在每個國家，小 B 都打算購買 N 樣紀念品。特別的是，無論在哪個國家，第 i 樣紀念品的價格都是那個國家的 a_i 元。

因為某些原因，小 B 特別喜歡 M 這個整數。小 B 定義他對一個國家的「喜好程度」為滿足以下條件的紀念品 i 的數量：當小 B 在該國家要購買第 i 個紀念品時，在使用最少可能的貨幣數量的情況下，使用的貨幣數量恰好為 M 。

顧著到處出遊的小 B 想請你幫他算算他對每一個國家的喜好程度，好讓他決定應該先前往哪些國家，你能寫一支程式完成他的要求嗎？

輸入格式

```

N M Q
a1 a2 ... aN
b1
b2
...
bQ

```

- N, M, Q, a_i, b_i 的意義如題目所述。

輸出格式

```

ans1
ans2
...
ansQ

```

- ans_i 為小 B 對第 i 個國家的喜好程度。

測資限制

- $1 \leq N \leq 2 \times 10^6$

- $1 \leq M, Q, a_i \leq 2 \times 10^6$
- $2 \leq b_i \leq 10^9$
- 所有輸入皆為整數

範例測試

Sample Input	Sample Output
10 3 10	1
1 2 3 4 5 6 7 8 9 10	2
2	3
3	2
4	2
5	2
6	2
7	1
8	1
9	1
10	
11	

評分說明

本題共有 4 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	9	$N, Q \leq 1000$
2	29	$a_i \leq 40000$
3	24	$a_i \leq 3 \times 10^5$
4	38	無額外限制

H. 圓規 (Compass)

問題敘述

奇異果很擅長和圓形有關的問題，例如圓形交點、圓形聯集面積、扇形聯集面積等等的問題都難不倒他。

今天，奇異果在路邊發現了一個大小為 7001×7001 的棋盤，正中央的格子座標為 $(0, 0)$ 。一開始，所有格子都是白色的，只有一個格子 $(r, 0)$ 是黑色的。因為這個棋盤實在是太大了，只能站在棋盤外面的奇異果並不知道 r 確切是多少，只知道 $0 < r \leq 3000$ 。

奇異果想要畫一個以 $(0, 0)$ 為圓心、半徑為 r 的圓。你可能會很疑惑要怎麼在棋盤上畫圓，因此奇異果告訴了你他心目中的圓形的定義：

- 對於 $0 \leq x \leq \sqrt{r^2 - y^2}$ ，格子 $(x, \lfloor \sqrt{r^2 - y^2} \rfloor)$ 、 $(-x, \lfloor \sqrt{r^2 - y^2} \rfloor)$ 、 $(x, -\lfloor \sqrt{r^2 - y^2} \rfloor)$ 、 $(-x, -\lfloor \sqrt{r^2 - y^2} \rfloor)$ 是黑色的。
- 對於 $0 \leq y \leq \sqrt{r^2 - x^2}$ ，格子 $(\lfloor \sqrt{r^2 - x^2} \rfloor, y)$ 、 $(\lfloor \sqrt{r^2 - x^2} \rfloor, -y)$ 、 $(-\lfloor \sqrt{r^2 - x^2} \rfloor, y)$ 、 $(-\lfloor \sqrt{r^2 - x^2} \rfloor, -y)$ 是黑色的。
- 其他格子是白色的。

剛好他在旁邊發現一台神奇的機器，能夠幫助他達成目標。這台機器可以做的操作如下：

- set_col(x, y, len, val) :**
 $val \in \{0, 1\}$ ，對於 $0 \leq k < len$ ，如果 $val = 1$ ，把 $(x, y + k)$ 變成黑色，反之變成白色。
- set_row(x, y, len, val) :**
 $val \in \{0, 1\}$ ，對於 $0 \leq k < len$ ，如果 $val = 1$ ，把 $(x + k, y)$ 變成黑色，反之變成白色。
- not_col(x, y, xout, yout, len) :**
對於 $0 \leq k < len$ ，如果 $(x, y + k)$ 本來是白色的，把 $(x_{out}, y_{out} + k)$ 變成黑色，反之變成白色。
- not_row(x, y, xout, yout, len) :**
對於 $0 \leq k < len$ ，如果 $(x + k, y)$ 本來是白色的，把 $(x_{out} + k, y_{out})$ 變成黑色，反之變成白色。
- and_col(x1, y1, x2, y2, xout, yout, len) :**
對於 $0 \leq k < len$ ，如果 $(x_1, y_1 + k)$ 和 $(x_2, y_2 + k)$ 本來**都是黑色的**，把 $(x_{out}, y_{out} + k)$ 變成黑色，反之變成白色。
- and_row(x1, y1, x2, y2, xout, yout, len) :**
對於 $0 \leq k < len$ ，如果 $(x_1 + k, y_1)$ 和 $(x_2 + k, y_2)$ 本來**都是黑色的**，把 $(x_{out} + k, y_{out})$ 變成黑色，反之變成白色。
- or_col(x1, y1, x2, y2, xout, yout, len) :**
對於 $0 \leq k < len$ ，如果 $(x_1, y_1 + k)$ 和 $(x_2, y_2 + k)$ 本來**有至少一個是黑色的**，把 $(x_{out}, y_{out} + k)$ 變成黑色，反之變成白色。
- or_row(x1, y1, x2, y2, xout, yout, len) :**
對於 $0 \leq k < len$ ，如果 $(x_1 + k, y_1)$ 和 $(x_2 + k, y_2)$ 本來**有至少一個是黑色的**，把 $(x_{out} + k, y_{out})$ 變成黑色，反之變成白色。
- xor_col(x1, y1, x2, y2, xout, yout, len) :**
對於 $0 \leq k < len$ ，如果 $(x_1, y_1 + k)$ 和 $(x_2, y_2 + k)$ 本來**有恰好一個是黑色的**，把 $(x_{out}, y_{out} + k)$

變成黑色，反之變成白色。

- `xor_row(x1, y1, x2, y2, xout, yout, len)` :
對於 $0 \leq k < len$ ，如果 $(x_1 + k, y_1)$ 和 $(x_2 + k, y_2)$ 本來有**恰好一個是黑色的**，把 $(x_{out} + k, y_{out})$ 變成黑色，反之變成白色。

每一個操作都會先讀取需要知道的格子顏色，再把要修改的格子一起修改。因此，如果有讀取的格子也是要修改的格子，那麼會被讀取到的是執行這整個操作之前的狀態。

注意如果奇異果所做的操作，會導致這台機器必須要讀取或修改棋盤外的格子，這台機器就會爆炸。要是他做了超過 6×10^6 個操作，或是如果他做的所有操作中， len 的總和超過 10^9 ，這台機器也會爆炸。雖然這台機器聽起來充滿了危險，但奇異果真的很想要畫圓形。不過，奇異果累了，所以請你告訴他應該要如何操作，才能滿足 r 不管是多少，都能使得棋盤的最終結果是奇異果心中完美的圓形。

實作細節

你需要在首行加入 `#include "Compass.h"`，並完成以下函式：

```
void draw_circle();
```

你的程式可以呼叫以下函式：

```
void set_col(int x, int y, int len, int val);
void set_row(int x, int y, int len, int val);
void not_col(int x, int y, int xout, int yout, int len);
void not_row(int x, int y, int xout, int yout, int len);
void and_col(int x1, int y1, int x2, int y2, int xout, int yout, int len);
void and_row(int x1, int y1, int x2, int y2, int xout, int yout, int len);
void or_col(int x1, int y1, int x2, int y2, int xout, int yout, int len);
void or_row(int x1, int y1, int x2, int y2, int xout, int yout, int len);
void xor_col(int x1, int y1, int x2, int y2, int xout, int yout, int len);
void xor_row(int x1, int y1, int x2, int y2, int xout, int yout, int len);

void debug(int x, int y);
```

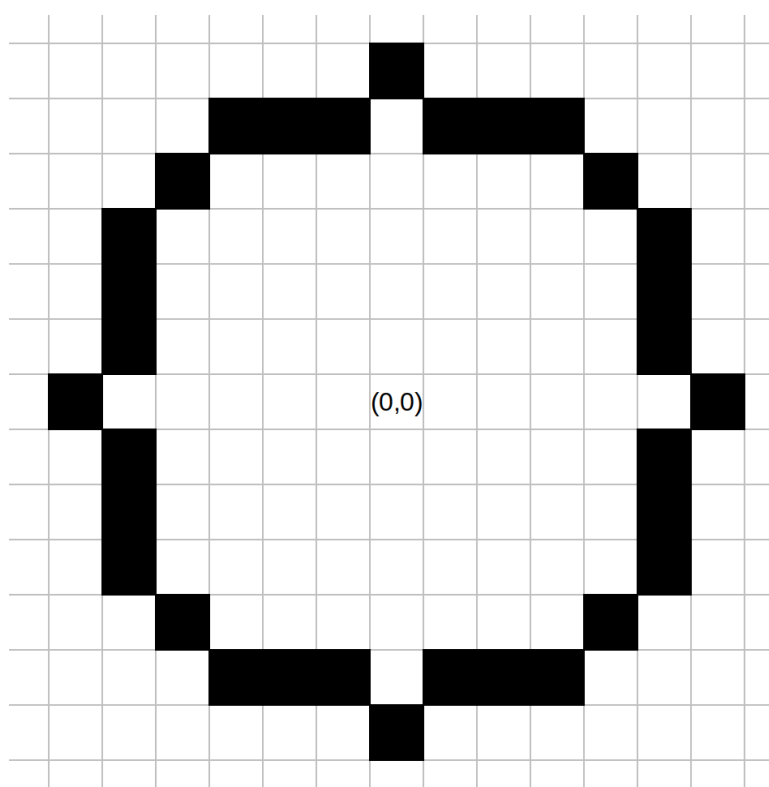
- `debug(x, y)` 會使範例評分程式輸出目前格子 (x, y) 的顏色，1 表示黑色、0 表示白色。在實際的評測過程中，此函式不會有任何效果，亦不計入操作次數。
- 其他函式的詳細功能請見問題敘述。
- `val` 必須是 0 或 1。
- `x, y, x1, y1, x2, y2, xout, yout` 必須是 $[-3500, 3500]$ 內的整數。
- `len` 必須是正整數。
- 在 `not_col, set_col, and_col, or_col, xor_col` 中，`y, y1, y2, yout` 加上 `len-1` 必須在 $[-3500, 3500]$ 內。
- 在 `not_row, set_row, and_row, or_row, xor_row` 中，`x, x1, x2, xout` 加上 `len-1` 必須在 $[-3500, 3500]$ 內。
- 你總共可以呼叫除了 `debug()` 以外的函式最多 6×10^6 次。
- 呼叫的所有函式的 `len` 總和不能超過 10^9 。

互動範例

當 $r = 6$ 時，一個會畫出正確的圓形的例子如下：

```
set_col(5, 1, 3, 1);
set_col(5, -3, 3, 1);
or_row(6, 0, 5, 0, 4, 4, 1);
set_row(-3, 5, 7, 1);
and_col(0, 4, 0, 4, 0, 5, 2);
not_row(-4, 4, -4, 4, 1);
xor_col(-5, -3, 5, -3, -5, -3, 7);
xor_col(-5, 0, 6, 0, -6, 0, 1);
set_row(-3, -5, 7, 1);
not_col(0, -6, 0, -6, 2);
set_col(-4, -4, 1, 1);
set_col(4, -4, 1, 1);
```

操作後的結果如下圖：



評分說明

本題共有 3 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	5	$r \leq 1000$
2	10	$r \leq 2500$
3	85	無額外限制

在第 3 個子任務中，假設奇異果做的操作次數為 q ，並且答案正確、機器也沒有爆炸，那麼你的得分為：

- 如果 $q \leq 7.5 \times 10^5$ ，你會得到 85 分。
- 如果 $7.5 \times 10^5 < q \leq 3.5 \times 10^6$ ，你會得到的分數是 $85 - \frac{55}{2.75 \times 10^6}(q - 7.5 \times 10^5)$ 。
- 如果 $3.5 \times 10^6 < q \leq 6 \times 10^6$ ，你會得到的分數是 $30 - \frac{30}{2.5 \times 10^6}(q - 3.5 \times 10^6)$ 。

範例評分程式

範例評分程式以下列格式讀取輸入：

- 第一列： r

r 表示圓形的半徑。

如果你的程式被評為 Accepted，範例評分程式會輸出 Accepted: q ，其中 q 表示除 `debug()` 外呼叫的函式總次數。如果你的程式被評為 Wrong Answer，範例評分程式會輸出 Wrong Answer: MSG，其中 MSG 意義如下：

- invalid operation：呼叫函式的參數不符合要求，例如修改或讀取的格子超出棋盤範圍。
- too many operations：呼叫函式次數過多，或者 len 的總和超出限制。
- wrong result：最後棋盤的長相不是奇異果想要的圓形。

在附件檔案中，有一個名為「Compass.zip」的壓縮檔，下載後解壓縮可以找到三個資料夾 `cpp`、`c` 和 `examples`，資料夾的意義分別為：

- `cpp`：內部包含一個檔案 `Compass.cpp`，你可以參考或修改這份程式碼，並將修改過的程式碼上傳至評測系統。你可以用檔案 `compile_cpp.sh` 或 `compile_cpp.bat` 在自己的電腦上編譯。
- `c`：內部包含一個檔案 `Compass.c`，你可以參考或修改這份程式碼，並將修改過的程式碼上傳至評測系統。你可以用檔案 `compile_c.sh` 或 `compile_c.bat` 在自己的電腦上編譯。
- `examples`：內部包含互動範例的輸入。

請不要嘗試撰寫題目指定需要函式以外的任何東西，例如自行輸入、輸出等。`grader.cpp` 與 `grader.c` 僅供參考用，並與 Judge 上的有所落差。

I. 子集合和 (SOS)

問題敘述

Sum Over Subsets 是一個經典的動態規劃技巧，在中國又被稱為高維前綴和。對於一個多重集合，如果將所有子集合 S 定義一個值 $c(S)$ ，這個技巧能算出所有 S 的子集 T 的價值總和。

然而，這道題目跟這個技巧沒有任何關係。

我們摯愛的 **Ji Kuai** 有一個函數 $f(S) = |S| \times \prod_{x \in S} x$ ，特別地，對於空集合 \emptyset ， $f(\emptyset) = 0$ 。

有一天，**Joy** 給了 **Ji** 一個多重集合 A 作為禮物。由於 **Ji** 是一名數學家，他想要算出 A 當中所有子集合的和，但是他很快的發現這樣太簡單了。作為一名天才競技程式選手，他覺得單純算出和太無聊了，於是他決定求出 A 的所有子集合 S 的 $f(S)$ 的和來讓 **Joy** 印象深刻。由於答案可能很大，他只在乎答案模 998244353 輸出。

Ji 在奈秒間就解決了這道題，而你每次看到 **Ji** 都好羨慕，為了能跟 **Ji** 一樣帥，請解決這題吧！

正式來說，請輸出 $ans = \sum_{S \subseteq A} f(S) \bmod 998244353$ ， f 的定義如同前述。

輸入格式

N
 $A_1 \ A_2 \ \dots \ A_N$

- N 是集合的大小。
- A_1, A_2, \dots, A_N 是集合的元素。

輸出格式

ans

- ans 意義如同問題敘述。

測資限制

- $1 \leq N \leq 10^6$
- $\forall i, 1 \leq A_i \leq 10^9$
- 所有輸入皆為整數

範例測試

Sample Input	Sample Output
2 2 3	17
4 1 2 3 4	326

評分說明

本題共有 5 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	5	$n \leq 20$
2	15	$n \leq 1000$
3	30	$n \leq 10^5$
4	15	$A_i \leq 20$
5	35	無額外限制