

第三屆大台南高一生程式設計排名賽

說明與注意事項

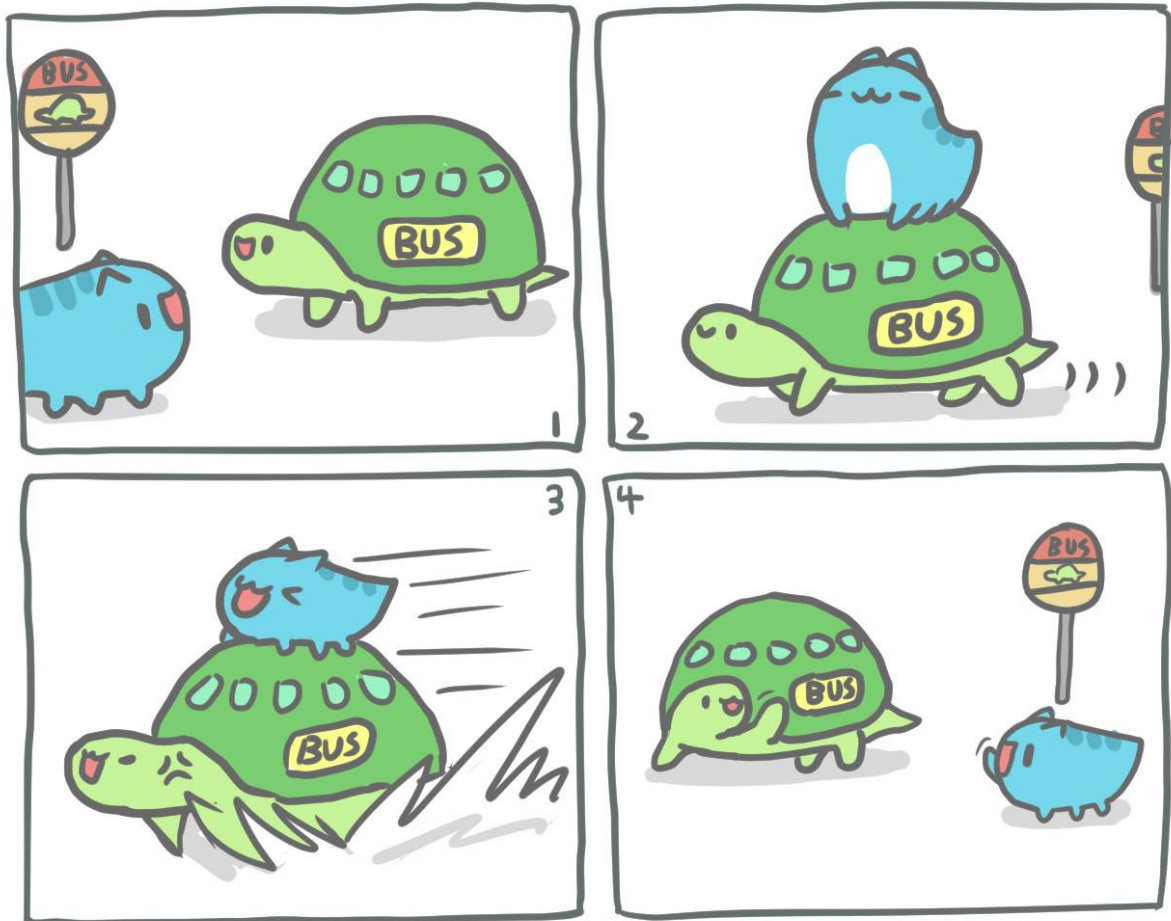
1. 本次競賽總共有 7 題，使用電腦自動評分，請上傳原始程式碼(*.cpp / *.c)
2. 競賽系統使用與全國學科能力競賽/資訊奧林匹亞相同的CMS系統，評測伺服器為 Ubuntu 14.04，編譯器版本為g++ 4.9.2 / gcc 4.9.2
3. 比賽時間: 13:00 ~ 16:00，共3小時，競賽結束後將無法再上傳程式碼
4. 排名方式以得到的總分排序，如果同分則同名次，可以在這裡觀看記分板：
<http://cms.tfcis.org/rank/>
比賽結束前一小時凍結記分板，不再更新
5. 如對題目有疑問可使用競賽系統中的訊息詢問
6. C語言scanf、printf對於64位元整數，請使用 **%lld** 做為格式化標記

題目列表

編號	簡稱	題目名稱	時間限制	記憶體限制	配分
A	turtle	烏龜公車	0.5 秒	512 MiB	97
B	forest	小雞森林	0.5 秒	512 MiB	103
C	start	開動	1 秒	512 MiB	150
D	potential	小雞的潛力	1 秒	512 MiB	200
E	tentacle	害羞的咪卡	1 秒	512 MiB	250
F	knapsack	不貳過	20 秒	2048 MiB	300
G	colorful	色彩繽紛	1 秒	512 MiB	430

A. 烏龜公車

turtle



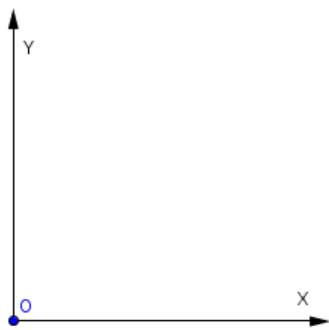
貓貓蟲-咖波 www.facebook.com/capocat

主人不陪咖波玩，還要叫咖波幫她做工作，於是咖波決定打消要主人陪他玩的念頭，自己搭烏龜公車出去玩。但是他聽說烏龜公車很不一樣，必須要有目的地的座標才能到達。

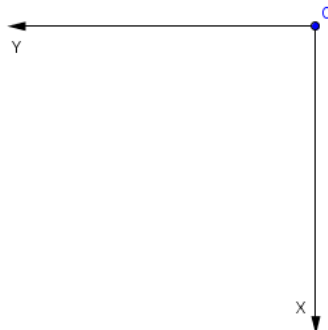
咖波把目的地的座標都記錄好之後，很開心的來到車站，卻發現烏龜公車的座標系和他使用的不一樣！聰明的咖波決定現場寫出一個程式來轉換座標。

給兩個不同的直角系，請做座標轉換。

咖波原來的座標：



烏龜公車的座標：



烏龜公車的座標原點在咖波原來的座標系上是(100,100)

輸入說明

輸入有一行，包含兩個數字X, Y，以空白隔開，代表咖波想去的其中一個目的地的座標

輸出說明

輸出一行，包含兩個數字X', Y'，以空白隔開，代表此目的地在烏龜公車的座標系上的座標

輸入限制

X, Y 為正整數

$$0 \leq X \leq 100$$

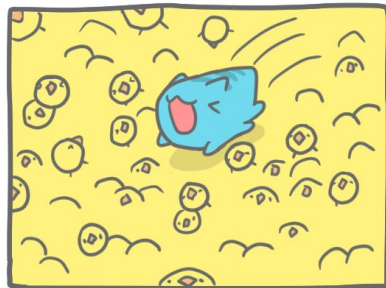
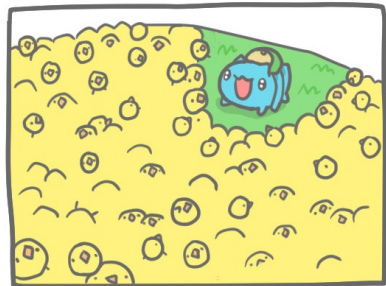
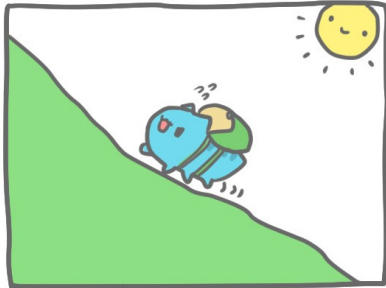
$$0 \leq Y \leq 100$$

範例

INPUT	OUTPUT
0 0	100 100

B. 小雞森林

forest



長途跋涉後，咖波抵達了充滿小雞的小雞森林。

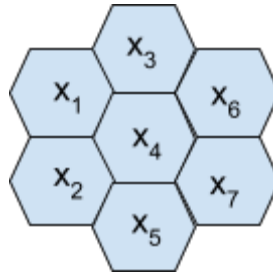
小雞森林大致分割成如圖的七個區塊，如圖，像個蜂窩。咖波想要數出區塊裡的小雞數量，但是咖波的立體視覺 (stereopsis) 不太好，只能數出自己面前有多少小雞，而不知道哪些比較近，哪些比較遠。

咖波最後把每個方向數出的小雞總和都寫下來了，咖波打算只去其中一個區塊，他希望在出發之前知道那個區塊有幾隻小雞。

你能告訴他這些區塊有多少小雞嗎？

這樣咖波也許會分你一隻可愛(不好吃)的小雞~>W<

小雞森林的形狀：



咖波把數出來的各方向總和用九個數字表示: $A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3$

A_1 = 紅色區域數字和 A_2 = 藍色區域數字和 A_3 = 藍色區域數字和	B_1 = 紅色區域數字和 B_2 = 藍色區域數字和 B_3 = 藍色區域數字和	C_1 = 紅色區域數字和 C_2 = 藍色區域數字和 C_3 = 藍色區域數字和

輸入說明

總共三行

第1行三個數字，分別為 A_1, A_2, A_3

第2行三個數字，分別為 B_1, B_2, B_3

第3行三個數字，分別為 C_1, C_2, C_3

輸出說明

輸出一行，包含兩個數字，以空白分隔，第一個數 k ，代表你算出了 X_k ，也就是你建議咖波去哪一格，第二個數字就是你所算出的 X_k ，代表那一格有幾隻小雞。

因為你只想要一隻小雞，所以幫咖波算出任何一格就行了。

根據你精闢的觀察，無法確認的格子是可能存在的，要是錯了咖波可能會把你吃掉，所以你不會冒這個險，對吧？

輸入限制

森林裡每一個格子裡的數字($X_1 \sim X_7$)皆為小於等於 10^5 的非負整數

範例

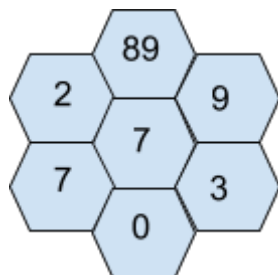
INPUT	OUTPUT
9 96 12 98 12 7 91 23 3	1 2
	2 7
	3 89
	4 7
	5 0
	6 9
	7 3

對於這筆範例測資，輸出右側範例輸出的其中任一種可能皆視為正確

注意 請只輸出一行，請勿同時輸出好幾個格子的解答

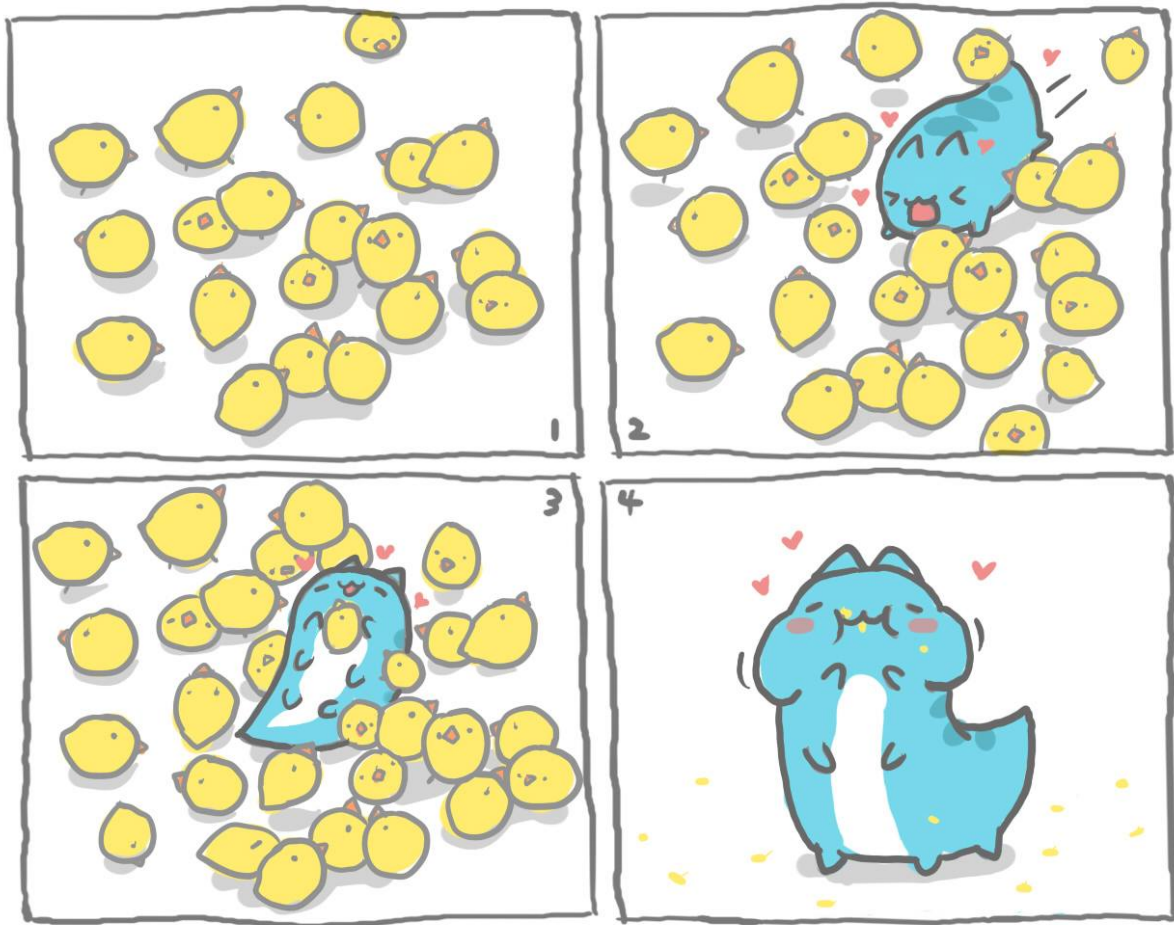
範例說明

以下這張圖是這個森林中小雞數量的示意圖



C. 開動

start



FB: CapooCat

咖波數完小雞的數量後，該是開動的時候了。

但是聰明的咖波很懂餐桌禮儀的！他有一些規則必須遵守，那就是吃小雞的順序。

首先，他必須決定從哪一個路口出發，以及從哪一個路口結束，且只能走一次。

第二，咖波不會在路口以外的地方停止，也不會回頭，但把經過的小雞全都吃光是基本的。

第三，吃小雞的總數是有限制的，吃太多主人會說他太胖，吃太少沒有飽足感。

經過咖波精密的調查，他發現小雞們的排列形成了一張特別的地圖，地圖上有 N 個路口，一條馬路連接著兩個路口，每條馬路邊會有特定數量的小雞，整個小雞社區是互相連通的，也就是從任意路口一定找得到一條路徑到達另一路口，而且懶惰的小雞絕對不會想要蓋多餘的馬路，只要能把那 N 個路口連接起來就夠了，所以只會蓋恰好 $N-1$ 條路，注意：路的盡頭也算是一個路口喔。

咖波已經調查完小雞社區的形狀了，他把那些路口由0開始編號，他想要從0號路口開始吃，有沒有辦法找到一條路徑使得吃到的小雞數量不會太多也不會太少呢？

請你幫他寫個程式算算看吧。

輸入說明

第1行有一個數字N，代表小雞社區總共有N的路口，這些路口被咖波編號成 0~N-1，咖波會從編號0的路口出發。

第2行兩個數字 Eat_{min} 和 Eat_{max} ，代表咖波最少要吃 Eat_{min} 、最多能吃 Eat_{max} 隻小雞。

接下來有N-1行，每行3個數 a, b, x，代表編號a的路口和編號b的路口間有一條路，這條路邊總共有x隻小雞

輸出說明

請告訴咖波從0號路口開始吃能不能滿足他的需求？

如果可以，請輸出"Eat Eat Eat"

否則，請輸出"Go find other chickens"

輸入限制

100%的測試資料保證：

$$1 \leq N \leq 10^4$$

$$0 \leq a, b \leq N-1$$

$$1 \leq x \leq 10^4$$

$$1 \leq Eat_{min} < Eat_{max} \leq 10^9$$

40%的測試資料保證：

$$1 \leq N \leq 100$$

$$1 \leq x \leq 100$$

範例一

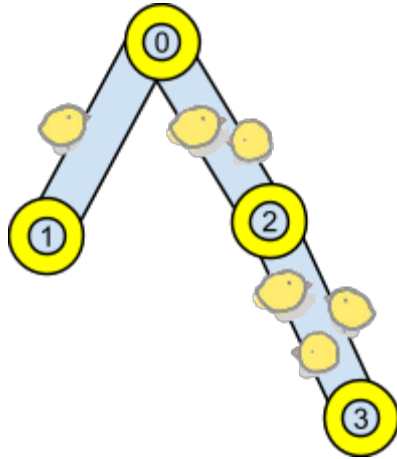
INPUT	OUTPUT
4 2 3 0 1 1 0 2 2 2 3 3	Eat Eat Eat

範例二

INPUT	OUTPUT
4 3 4 0 1 1 0 2 2 2 3 3	Go find other chickens

範例說明

範例1和範例2的地圖相同，如下：



圖中黃色為路口，藍色部份是馬路

從0號路口走到1號路口，總共會經過1隻小雞

從0號路口走到2號路口，總共會經過2隻小雞

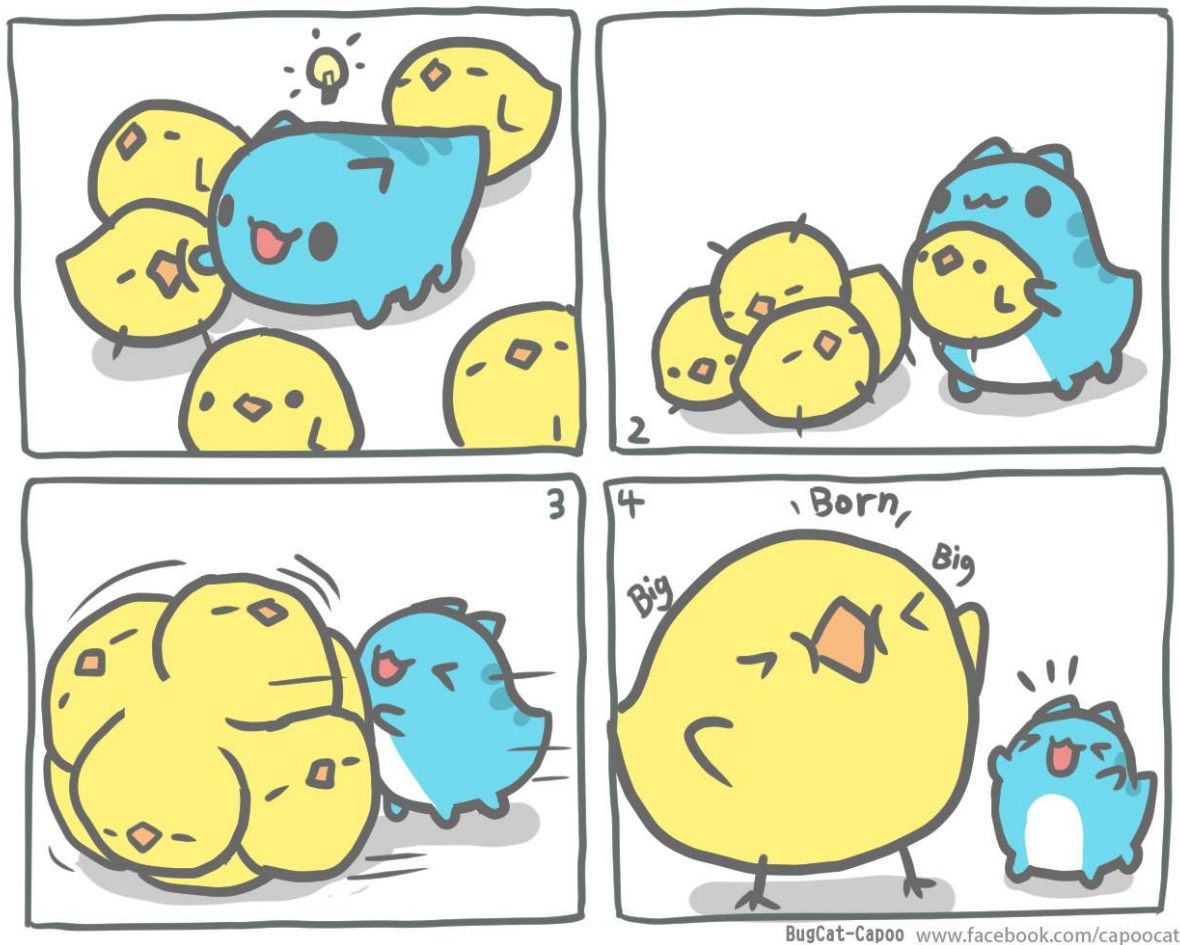
從0號路口走到3號路口，總共會經過5隻小雞 (2+3)

對於範例1，走到路口二可以吃到2隻小雞，符合題目要求的2~3隻小雞

對於範例2，不管走到哪一個點都沒辦法吃到3~4隻小雞

D. 小雞的潛力

potential



咖波覺得計算小雞的飽足感有點累。好險聰明的咖波很快就發現更好的方法。

咖波發現，小雞是可以合成和分割的。於是他這麼做：

咖波先把 N 隻大小不同的小雞排成一直線，飽足感由左至右為 $H_1, H_2, \dots, H_n$ ，咖波可以任選連續的小雞 $H_i, H_{i+1}, H_{i+2}, \dots, H_r$ ，把這些小雞的大小同時增加或減少一個數值 V 。要是小雞變得太小，飽足感也可能會小於零(消化的耗能還比不上得到的能量)。

最後咖波想知道最大的小雞和最小的的小雞，相差多少飽足感。但是咖波只想快點繼續進食。你能幫咖波解決這個問題嗎？

實作說明

你必須上傳一個指定函數，其函數原型如下：

```
int Answer(int N,int P,int *H,int *L,int *R,int *V);
```

參數意義代表如下：

N：小雞的數量

P：修改的次數

H[0],H[1]...H[N-1]：小雞一開始的大小，注意：H[0]是題目敘述中的H₁喔，別搞錯了

L[0],L[1]...L[P-1]：每次修改的左界

R[0],R[1]...R[P-1]：每次修改的右界

V[0],V[1]...V[P-1]：每次修改增加的大小

回傳值意義代表如下：

一個整數，為完成操作後，最大和最小的的小雞大小差多少。

我們可以參考這一個範例：

```
N=5  
P=3  
H[]={0,0,1,0,1}  
L[]={1,2,1}  
R[]={3,5,1}  
V[]={1,-1,2}
```

代表有5隻小雞，大小分別為{0,0,1,0,1}，接下來有3次修改。

第一次修改的區間為H₁~H₃，增加大小1，修改完後小雞大小為{1,1,2,0,1}

第二次修改的區間為H₂~H₅，增加大小-1，修改完後小雞大小為{1,0,1,-1,0}

第三次修改的區間為H₁~H₁，增加大小2，修改完後小雞大小為{3,0,1,-1,0}

於是答案為最大減最小，即3-(-1)=4。

限制

函式被使用的次數不超過一次

$$1 \leq N \leq 10^6$$

$$0 \leq P \leq 10^6$$

$$-100 \leq H[i] \leq 100$$

$$1 \leq L[i] \leq R[i] \leq N$$

$$-100 \leq V[i] \leq 100$$

子任務

子任務	分數	額外輸入限制
1	18	$N \leq 5, P = 1$
2	22	$N \leq 5$
3	30	$P = 0$
4	30	$P \leq 10^4$
5	40	修改的區間全部不相交
6	60	無

範例程式

為方便您撰寫此題，下提供範例程式檔案，在指定位置撰寫您的程式碼即可。下提供測試資料的輸入方法。

下載: 題目附件中 potential.cpp

範例程式輸入

第一行有兩個數字：N,P，接下來下一行有N個數字，分別代表H[i]的數值，在接下來有P行，每行有三個數字，分別為L[i],R[i],V[i]。

範例程式輸出

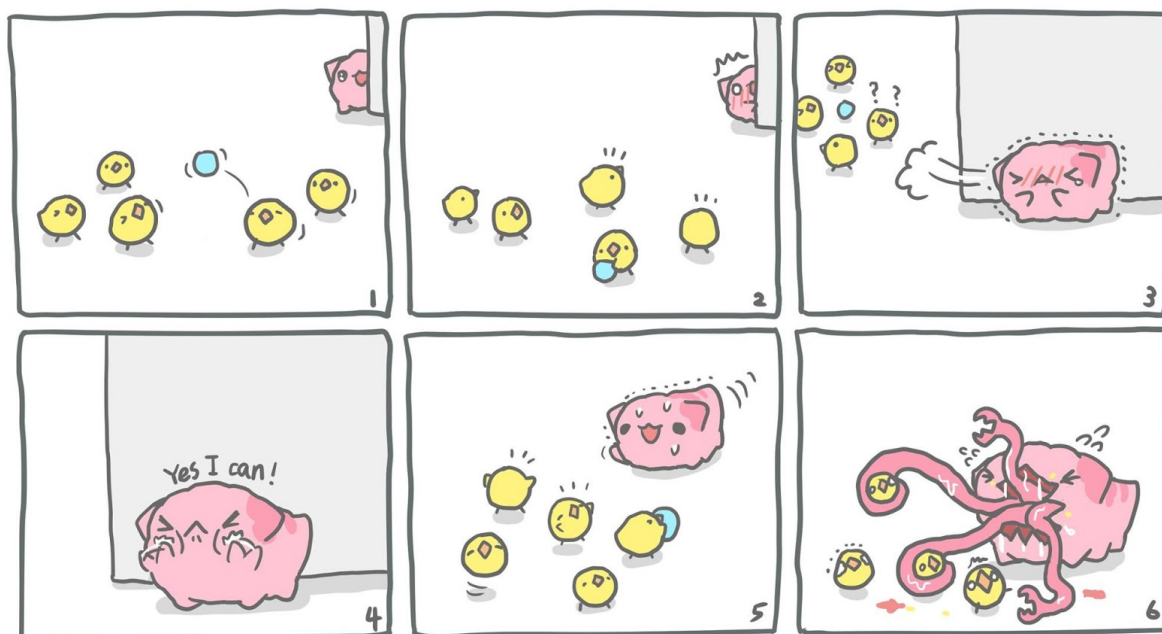
直接輸出Answer的回傳值。

範例輸入、輸出

INPUT	OUTPUT
5 3 0 0 1 0 1 1 3 1 2 5 -1 1 1 2	4

E. 害羞的咪卡

tentacle



BugCat-Capoo www.facebook.com/capooocat

咖波吃了好多小雞之後，發現咪卡也來了。

咪卡是一隻害羞的貓貓蟲，他的吃法和咖波不太一樣。她是用觸手來吃小雞的。

咪卡每次能伸出一隻、兩隻、四隻、八隻……觸手，不同數量的觸手必須消耗不同能量。

假設咪卡一次伸出 k 隻觸手，必須恰好吃 k 隻小雞。

輸入說明

輸入總共有三行。

第一行有一個整數 X ，代表總共有 X 隻小雞。

第二行一個數字 N ，代表咪卡最多能一次伸出 2^{N-1} 隻觸手。

第三行有 N 個正整數 a_i ($i=0\sim N-1$)， a_i 帶表咪卡 2^i 隻觸手所需要花費的能量。

注意：輸入數字較大，某些數字可能必須使用長整數(long long int)才能儲存。

輸出說明

請你計算如果咪卡想要吃掉所有的小雞至少需要耗費多少能量，並輸出所需消耗能量除以1000000007的餘數。

輸入限制

$$1 \leq X \leq 10^{18}$$

$$1 \leq N \leq 60$$

$$1 \leq a_i \leq 10^9$$

子任務

子任務	分數	額外輸入限制
1	109	$1 \leq X \leq 20$ $1 \leq N \leq 5$ $1 \leq a_i \leq 20$
2	141	無

範例一

INPUT	OUTPUT
52 4 1 2 3 6	39

範例二

INPUT	OUTPUT
63 4 7 1 2 2	24

範例一說明

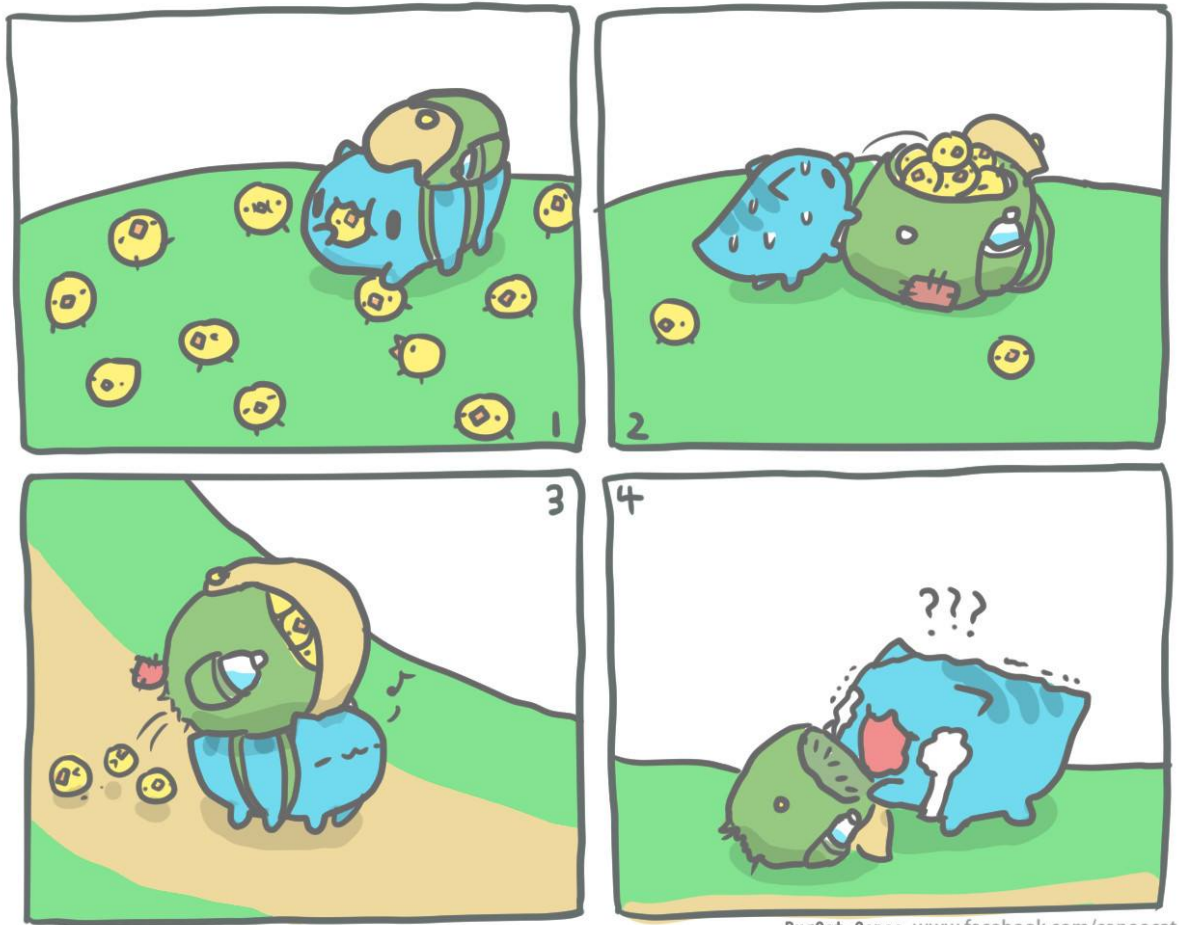
一次可以最多伸出8隻觸手

1. 一次伸出1隻，消耗能量1
2. 一次伸出2隻，消耗能量2
3. 一次伸出4隻，消耗能量3
4. 一次伸出8隻，消耗能量6

其中一種能使消耗能量最小的方法是：伸出觸手13次，一次伸出4隻觸手

F. 不貳過

knapsack



吃了好多小雞之後，咖波想把小雞帶回家繼續享用。但是咖波想起了以前的慘痛教訓。

聰明的咖波不會犯同一種錯誤。他決定寫一個程式好好計算包包的大小。

咖波總共帶了 N 個包包，每個包包有不同的容量，每隻小雞也有不同的大小和飽足度。

比較大隻的小雞不代表飽足度就愈高喔！

咖波想要他帶回家的所有小雞的飽足度總和能夠最大，你能寫個程式幫他算一算嗎？

輸入說明

第一行有一個正整數 N ，代表咖波帶了 N 個包包。

第二行有 N 個正整數， $B_0, B_1, B_2, \dots, B_{N-1}$ ，代表這 N 個包包的大小，大小為 B_i 的包包裡面裝的小雞大小總和不得超過 B_i ，否則就會發生和上次一樣的慘劇喔！

第三行有一個正整數 M ，代表有 M 隻小雞。

接下來 M 行，每行兩個數字， W_i, V_i ，分別代表第 i 隻小雞的重量和飽足度。

輸出說明

輸出一個正整數，代表帶回去的小雞飽足度總和最多是多少？

輸入限制

$$1 \leq N \leq 3$$

$$1 \leq B_i \leq 70$$

$$1 \leq M \leq 100$$

$$1 \leq W_i \leq 100$$

$$1 \leq V_i \leq 100$$

子任務

子任務	分數	額外輸入限制
1	102	$N = 1$
2	99	$N = 2$
3	99	$N = 3$

範例一

INPUT	OUTPUT
2 10 7 3 6 10 5 4 8 6	16

範例二

INPUT	OUTPUT
1 12 3 6 10 5 4 8 6	14

G. 色彩繽紛

colorful



貓貓蟲-咖波 www.facebook.com/capooocat

這次咖波的包包沒有破掉，真是太好了。咖波回家後開心的和帶回來的小雞玩耍，突然他發現了一隻藍色的小雞。

咖波覺得藍色的小雞真是太可愛了，聰明的咖波想到他能用顏料把其他小雞也塗成藍色的。但是不同顏色的小雞，排列出的效果也是很重要的。咖波決定寫出一個程式，讓他能預覽塗色和排列的結果。

相鄰的藍色小雞會形成一個藍色小雞團，他希望在他塗完顏色後能夠知道有多少藍色小雞團。

有 $N+1$ 隻小雞，咖波把他們排成了一直線，並將他們依序編號為 $0, 1, 2, \dots, N$ ，一開始，除了編號0的小雞本來就是藍色之外，其餘的 N 隻小雞顏色皆為黃色，咖波需要以下3種操作：

1. 將編號 $L \sim R$ 的所有小雞都塗成藍色
2. 將編號 $L \sim R$ 的所有小雞都塗成黃色
3. 計算編號 $L \sim R$ 之間(含)有幾個藍色小雞團

參考以下範例：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

假設目前小雞的狀況如圖，則如果詢問 $2 \sim 12$ 應回答有3個藍色小雞團

輸入說明

第一行有一個數字 N 代表有 $N+1$ 小雞，編號由 $0 \sim N$ 。

第二行有一個數字 Q 代表有 Q 次操作。

接下來 Q 行，每行三個數字， X, L, R ， X 代表第幾種操作， L, R 意義如題目敘述。

輸出說明

對於每個操作3，請輸出一行，包含一個數字，代表有幾個藍色小雞團

輸入限制

$$0 \leq N \leq 10^9$$

$$1 \leq Q \leq 10^5$$

$$0 \leq L, R \leq N$$

子任務

子任務	分數	額外限制
1	30	$0 \leq N \leq 100, 1 \leq Q \leq 100$
2	40	所有第1,2種操作的區間($L \sim R$)不交錯也不相鄰，對於所有操作二 $L > 0$ ，且所有第3種操作中 $L=0, R=N$
3	70	不包含第二種操作，第三種操作中 $L=0, R=N$ 且只會出現一次
4	70	不包含第二種操作，所有第三種操作中 $L=0, R=N$
5	50	第三種操作中 $L=0, R=N$ 且只會出現一次
6	50	所有第三種操作中 $L=0, R=N$
7	70	$N \leq 10^5$
8	50	無

範例

INPUT	OUTPUT
10	0
10	1
2 0 10	1
3 0 10	2
1 2 4	2
3 0 10	1
1 5 6	
3 0 10	
2 4 5	
3 0 10	
3 3 6	
3 0 5	

範例說明

一開始給定的區塊如下：

0	1	2	3	4	5	6	7	8	9	10
										

將 0 ~ 10 塗成黃色小雞：

0	1	2	3	4	5	6	7	8	9	10
										

詢問 0 ~ 10 有幾個藍色小雞團，答案是0個

將 2 ~ 4 隻小雞塗成藍色

0	1	2	3	4	5	6	7	8	9	10
										

詢問 0 ~ 10 有幾個藍色小雞團，答案是1個

將 5 ~ 6 隻小雞塗成藍色

0	1	2	3	4	5	6	7	8	9	10
										

詢問 0 ~ 10 有幾個藍色小雞團，答案是2個

將 4 ~ 5 隻小雞塗成黃色

0	1	2	3	4	5	6	7	8	9	10
										

詢問 0 ~ 10 有幾個藍色小雞團，答案是2個

詢問 3 ~ 6 有幾個藍色小雞團，答案是2個

詢問 0 ~ 5 有幾個藍色小雞團，答案是1個