

香港電腦奧林匹克競賽 2015/16

高級組

題目總覽

題號	題目	子任務	佔分
S161	軍事訓練	19 + 14 + 17 + 28 + 22	100
S162	露寶的盛宴	20 + 35 + 45 (提供部分分數)	100
S163	等差數列	13 + 15 + 25 + 21 + 26	100
S164	愛麗斯的餐點	11 + 9 + 27 + 15 + 38	100

對於所有題目

執行時間限制：1 秒

記憶體限制：256 MB

注意：

除非特別注明，否則輸入輸出將依照以下格式：

- 同一行中，數字與數字或字元之間需有一個空格。
- 同一行中，字元與字元之間並無空格。
- 每個字串需放在獨立的行。
- 每行的最尾不可為空白符。
- 不可有空行，輸入輸出最尾的換行符例外。

C++ 程序員請注意 (`cin / cout`) 可能導致 I/O 樽頸使程式執行變慢。

有些題目可能需要使用 64 位元整數。在 Pascal 中它是 `int64`。在 C/C++ 中它是 `long long int`。

C/C++ 程序員應使用 `"%lld"` 進行 64 位元整數的輸入/輸出。

所有題目均有細分多個子任務，你需要通過該子任務中的所有測試數據才能得到分數。

軍事訓練

題目

位元組國 (Byteland) 的軍隊是由機械人組成，機械軍隊的其中一個優點是所有士兵都絕對服從指令。

露寶 (Robo) 是一種新的機械人，它們正接受基本軍事訓練，以提升進行重覆性動作的能力。軍訓進行於一個 $N \times N$ 的網格，網格左上角是 $(1, 1)$ ，右下角是 (N, N) 。在第 0 時間單位，露寶處於 (r_0, c_0) ，並會收到一些指令，這些指令形成一串長度為 K 的字串。字串的每個字元可以是 U、D、L 或 R，分別代表「上」、「下」、「左」、「右」。

每單位時間露寶都會做同一件事。首先，它會看看字串的第一個字元。如果字元是 U，它便向上移一格；如果字元是 D，它便向下移一格；L 與 R 亦相類似。然後，它會檢查自己是否出界。如果它處於 $N \times N$ 網格之外，軍訓便完畢。否則，它會把字串的第一個字元搬到字串的最尾。

最近，基本軍訓加入了一個新項目以提升露寶的格鬥能力。在第 0 時間單位，一個沙包機械人被放於 (r_1, c_1) ，它會留在那兒直至軍訓完畢。每當露寶遇上沙包機械人（即是它們處於同一格），露寶便會打沙包機械人一拳。請注意，如果你把沙包機械人放於 (r_0, c_0) ，露寶不會立刻打它，而是在以後重新回到 (r_0, c_0) 時才會打它。

位元組國軍隊的將軍想知道新軍訓的效用。你的任務是計算出，如果適當地放置沙包機械人，露寶最多可打他多少拳。

輸入

第一行有兩個整數 N 和 K 。

第二行有兩個整數 r_0 和 c_0 。 $(1 \leq r_0, c_0 \leq N)$

第三行有一串長度為 K 的字串，是露寶在第 0 時間單位收到的指令。字串只包含 U、D、L 和 R。保證露寶在第 K 時間單位不會處於 (r_0, c_0) 。

輸出

輸出一個整數，代表露寶最多可打沙包機械人多少拳。

樣例

輸入	輸出	輸入	輸出	輸入	輸出
3 2	0	3 5	3	2 6	2
1 1		1 1		1 1	
LU		DDURL		RLDUUU	

解釋

在樣例一中，露寶在第 1 時間單位時已經出界，所以他不會再出拳。

在樣例二中，露寶的路徑為 $\rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (4, 1)$ 。

沙包機械人應置於 $(2, 1)$ 。

在樣例三中，露寶的路徑為 $\rightarrow (1, 2) \rightarrow (1, 1) \rightarrow (2, 1) \rightarrow (1, 1) \rightarrow (0, 1)$ 。

沙包機械人應置於 $(1, 1)$ 。

子任務

子任務 1 (19 分)

$$N = 5$$

$$K = 3$$

$$r_0 = c_0 = 3$$

第一個指令是 U

子任務 2 (14 分)

$$1 \leq N \leq 1000$$

$$1 \leq K \leq 20$$

指令只由 L 和 R 組成

子任務 3 (17 分)

$$1 \leq N \leq 1000$$

$$1 \leq K \leq 20$$

子任務 4 (28 分)

$$1 \leq N \leq 10^9$$

$$1 \leq K \leq 50$$

子任務 5 (22 分)

$$1 \leq N \leq 10^9$$

$$1 \leq K \leq 2000$$

露寶的盛宴

題目

你知道機械人吃甚麼嗎？在位元組國裏，露寶機械人的膳食很簡單 – 他們只吃瓶裝油。平日，瓶裝油只有一款口味可選擇；在一年一度的「露寶節」(Annual Robos' Festival)，則有四款口味提供。不同口味的瓶裝油的瓶子有著不同的顏色，分別是紅、藍、橙、紫。

每年露寶節都會舉行一場盛宴。預備盛宴時，機械工人們會把瓶裝油有規律地排列在地上，形成一個圖案。

不妨把地面看為笛卡兒坐標平面 (Cartesian coordinate plane)。平面上的兩點 (x_1, y_1) 與 (x_2, y_2) 之間的曼哈頓距離 (Manhattan distance) 是 $|x_1 - x_2| + |y_1 - y_2|$ 。現考慮平面上每個整數坐標點 (x, y) 與原點 $(0, 0)$ 的曼哈頓距離除以 4 的餘數。

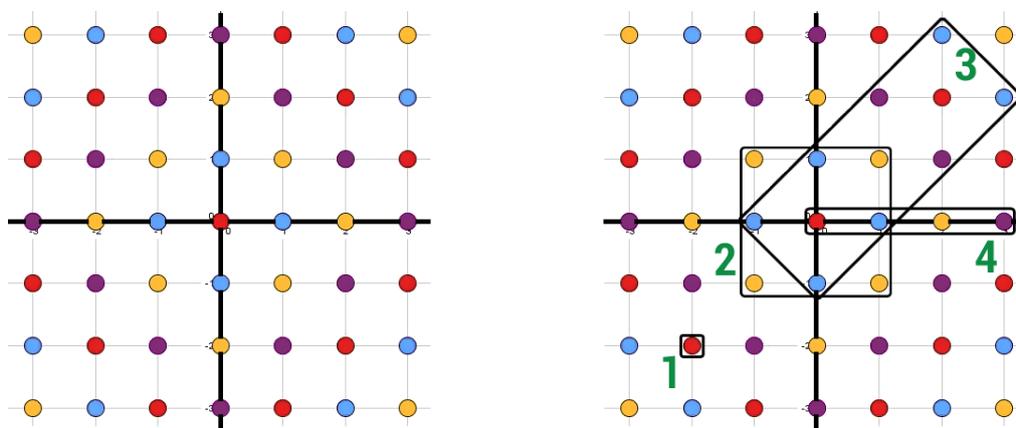
如果餘數是 0， (x, y) 將會放一瓶紅色 (R) 瓶裝油。

如果餘數是 1， (x, y) 將會放一瓶藍色 (B) 瓶裝油。

如果餘數是 2， (x, y) 將會放一瓶橙色 (O) 瓶裝油。

如果餘數是 3， (x, y) 將會放一瓶紫色 (P) 瓶裝油。

下圖展示原點 $(0, 0)$ 附近的圖案，此圖之彩色版本另外印於草稿紙。



不同口味的瓶裝油有著不同的美味值。四款口味 (紅、藍、橙、紫) 的瓶裝油的美味值分別可用四個整數 T_R, T_B, T_O, T_P 表示。

今年有 N 位露寶機械人參加盛宴。它們排好隊，按照指定的規則取走若干瓶油。輪到一個露寶時，它會執行以下其中一個動作：

1. 指定一個四邊均與坐標軸平行的長方形，並取走長方形內所有瓶裝油 (包括在長方形邊界上的瓶裝油)。
2. 指定一個四邊均與坐標軸呈 45 度角的長方形，並取走長方形內所有瓶裝油 (包括在長方形邊界上的瓶裝油)。

然後，新的瓶裝油會補上，令地面回復原來的圖案。

你的任務是計算出每個露寶取得的瓶裝油的美味值總和。

輸入

第一行有四個整數： T_R 、 T_B 、 T_O 與 T_P 。

第二行是整數 N 。

以下 N 行的第 i 行描述露寶 i 的動作。每行有五個整數： C 、 x_1 、 y_1 、 x_2 與 y_2 。 $C = 1$ 代表機械人執行了動作 1， $C = 2$ 則代表機械人執行了動作 2。 (x_1, y_1) 與 (x_2, y_2) 是機械人所指定的長方形的對角的坐標。請注意，長方形的面積可能是零。

輸出

輸出 N 行。第 i 行是露寶 i 取得的瓶裝油的美味值總和。

樣例

輸入	輸出	部分分數輸出	部分分數輸出
1 2 4 8	1	1	(任何整數)
4	25	25	(任何整數)
1 -2 -2 -2 -2	34	(任何整數)	34
1 -1 1 1 -1	15	15	(任何整數)
2 0 -1 2 3			
1 3 0 0 0			

解釋

露寶 1: 1 紅。總和 = $1 \times 1 = 1$

露寶 2: 1 紅、4 藍、4 橙。總和 = $1 \times 1 + 4 \times 2 + 4 \times 4 = 25$

露寶 3: 2 紅、6 藍、1 橙、2 紫。總和 = $2 \times 1 + 6 \times 2 + 1 \times 4 + 2 \times 8 = 34$

露寶 4: 1 紅、1 藍、1 橙、1 紫。總和 = $1 \times 1 + 1 \times 2 + 1 \times 4 + 1 \times 8 = 15$

子任務

子任務 1 (20 分)

$0 \leq T_R = T_B = T_O = T_P \leq 100$ (四款口味的瓶裝油的美味值相等)

$1 \leq N \leq 10000$

$-10^8 \leq x_1, y_1, x_2, y_2 \leq 10^8$

子任務 2 (35 分)

$0 \leq T_R, T_B, T_O, T_P \leq 100$

$1 \leq N \leq 100$

$-100 \leq x_1, y_1, x_2, y_2 \leq 100$

子任務 3 (45 分)

$0 \leq T_R, T_B, T_O, T_P \leq 100$

$1 \leq N \leq 10000$

$-10^8 \leq x_1, y_1, x_2, y_2 \leq 10^8$

計分方式

在一個子任務內，如果你的程式對於每一測試都能正確輸出至少一種動作的答案，你會得到子任務的 80% 分數 (即 16、28 及 36 分)。請注意，即使你只想處理其中一種動作，你的程式仍然要輸出 N 行。

等差數列

題目

瓊斯博士 (Dr. Jones) 是位元組國學園 (Byteland Academy) 的教授。最近，他正在教班上的同學甚麼是等差數列：若一條數列中相鄰兩項的差為常數，則稱它為等差數列，例如數列 1, 3, 5 和數列 7, 5, 3 都是等差數列。

今天，瓊斯博士非常疲倦，為了令數學科的課堂時間比較輕鬆，他決定今天不教任何新知識，只問同學問題。瓊斯博士拿出了 R 張卡牌，卡牌上分別寫有數字 $1, 2, \dots, R$ 。然後，他請班上最傑出學生的學生，愛麗絲 (Alice) 在 R 張卡牌中任選 N 張，並把它們由左至右以一個特別的方式排列在一條直線上。

我們由左至右地考慮那 N 張卡牌上的數字，我們可以視它為一條長度為 N 的數列，不妨以 S 表示此數列，即是 S_i 為由左面起第 i 張卡上的數字。瓊斯博士要求愛麗絲在排列她選出來的 N 張卡牌時，要令生成出來的數列不包含任何長度為 3 的等差子數列，也就是說，在愛麗絲排列出來的數列 S 中，不可以存在 (i, j, k) ($1 \leq i < j < k \leq N$) 令 $S_j - S_i = S_k - S_j$ 。

例如，設 $R = 7$ 及 $N = 5$ ，數列 1, 3, 7, 6, 2 及數列 4, 6, 1, 2, 5 都是正確答案的例子，但數列 1, 3, 7, 5, 6 是不被接受的答案，這是因為它包含子數列 1, 3, 5 — 一個等差數列。

請幫愛麗絲選出及排列這些卡牌。

輸入

第一且唯一一行有兩個整數 R 和 N 。
測試數據的輸入保證最少有一組答案。

輸出

於一行內輸出 N 個以空格分隔的整數，代表一組符合瓊斯博士要求的卡牌排列。若有超過一組卡牌排列符合卡牌排列的要求，你可以輸出任何一組。最後一個數字後面不可有空格。

樣例

輸入	輸出	輸入	輸出
2000000 5	1 3 7 6 2	4 4	3 4 1 2

子任務

子任務 1 (13 分)

$1 \leq N \leq 10$
 $R = 2000000$

子任務 2 (15 分)

$1 \leq N \leq 10$
 $R = N$

子任務 3 (25 分)

$1 \leq N \leq 250$
 $R = 4000$

子任務 4 (21 分)

$1 \leq N \leq 10000$
 $R = 2000000$

子任務 5 (26 分)

$1 \leq N \leq 10000$
 $R = N$

愛麗斯的餐點

題目

上完瓊斯博士的課後，愛麗斯感到又累又餓，便決定去餐廳吃些東西。菜單上有 N 道菜式，每道菜式由上至下用 1 到 N 編號。愛麗斯看完菜單後便知道每道菜有多美味，並給予每道菜一個美味值，菜式 i 的美味值是 D_i 。

愛麗斯很趕時間，他只有 K 分鐘吃東西，而每分鐘她只能夠吃完一道菜。另外，即使是同一道菜，愛麗斯越遲吃便會越開心，這是因為她餓到不太享受最初的數道菜。如果她在第 t 分鐘時吃掉菜式 i ，她便會從這道菜中得到 $t \times D_i$ 點快樂值。總快樂值是愛麗斯從每道菜中得到的快樂值的總和。

愛麗斯懶得逐一點菜，所以她想選出菜單上剛好 K 道連續的菜式，並順序吃掉它們。同時，她想盡可能得到快樂值。例如說，菜單上有 5 道菜，它們的美味值 D_1, \dots, D_5 是 2, 4, 1, 5 和 3。愛麗斯有 3 分鐘吃東西，她應該選擇第 2 至 4 道菜，並得到 $1 \times 4 + 2 \times 1 + 3 \times 5 = 21$ 點總快樂值。

作為愛麗斯的好朋友，請你協助她找出他能得到最大的總快樂值。

輸入

第一行有兩個整數 N 和 K 。

第二行有 N 個整數，第 i 個整數是菜式 i 的美味值 D_i 。

輸出

輸出一個整數，他能得到最大的總快樂值。

樣例

輸入	輸出	輸入	輸出
5 3	21	6 2	5
2 4 1 5 3		3 1 1 1 1 1	

子任務

子任務 1 (11 分)

$$1 \leq N \leq 2000$$

$K = 1$ 即是愛麗斯只可吃一道菜

$$0 \leq D_i \leq 1000$$

子任務 2 (9 分)

$$1 \leq N \leq 2000$$

$$K = N$$

$$0 \leq D_i \leq 1000$$

子任務 3 (27 分)

$$1 \leq N \leq 2000$$

$$1 \leq K \leq N$$

$$0 \leq D_i \leq 1000$$

子任務 4 (15 分)

$$1 \leq N \leq 100000$$

$$1 \leq K \leq N$$

$$0 \leq D_1 \leq D_2 \leq \dots \leq D_{N-1} \leq D_N \leq 1000$$

子任務 5 (38 分)

$$1 \leq N \leq 100000$$

$$1 \leq K \leq N$$

$$0 \leq D_i \leq 1000$$