

香港女子電腦奧林匹克競賽 2025/26

題目總覽

編號	名稱	執行時間限制	記憶體限制	子任務
G261	去吃壽司	1.000 s	256 MB	9 + 9 + 15 + 16 + 18 + 25 + 8
G262	移除字母	1.000 s	256 MB	11 + 22 + 14 + 21 + 32
G263	螺絲	1.000 s	256 MB	7 + 9 + 17 + 24 + 23 + 20
G264	無盡打地鼠	1.000 s	256 MB	13 + 22 + 18 + 13 + 16 + 18
G265	迷你地鐵	1.000 s	256 MB	7 + 10 + 17 + 22 + 23 + 21
G266	全速前進	1.000 s	256 MB	8 + 14 + 20 + 19 + 21 + 18

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。
- 輸出將自動被修正如下：每行最尾的連續的空格會被刪除，及在輸出最後補上換行符(如沒有)。其他格式問題則不會修正。

C++ 使用者請注意 `cin` / `cout` 可能導致輸入輸出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 C++ 中它是 `long long` 而其 `scanf`/`printf` 代號是 `%lld`。

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。

主 辦



協 辦



贊 助



G261 - 去吃壽司

執行時間限制: 1.000 s / 記憶體限制: 256 MB

愛麗絲和鮑伯正在一家壽司店用餐。在壽司店裏，有不同種類的壽司，以及一些副餐。不同壽司的價格按其碟的顏色分類，其中白色最便宜，黑色最貴。菜單如下：

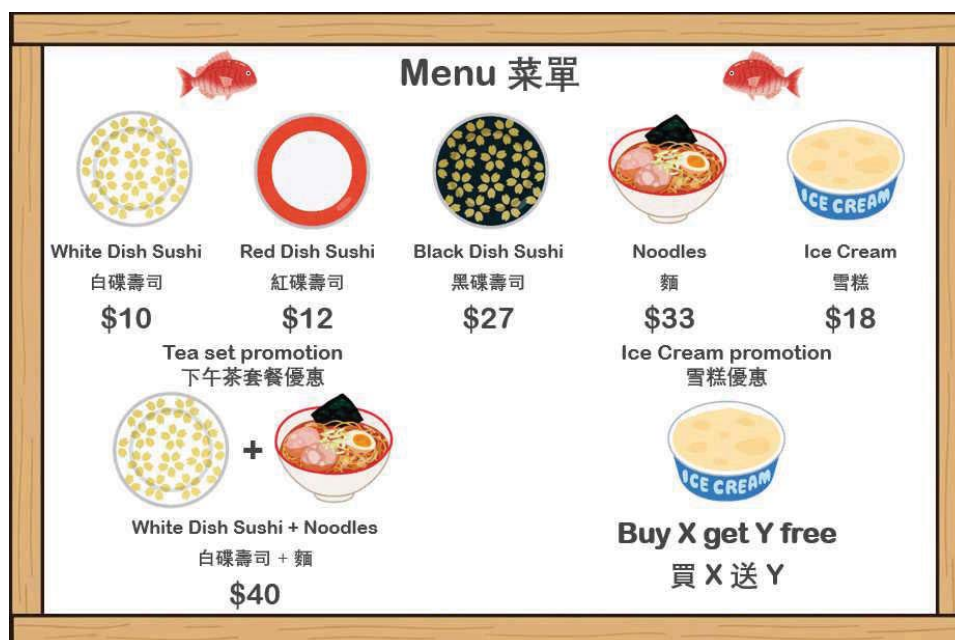


圖 1：壽司店的菜單和優惠。

正式地，壽司店提供以下餐點：

編號	名稱	價格
1	白碟壽司	\$10
2	紅碟壽司	\$12
3	黑碟壽司	\$27
4	麵	\$33
5	雪糕	\$18

此外，壽司店還提供一些特別優惠：

- 從 14:00 到 16:59（含），有一款售價為 \$40 的下午茶套餐，包含一碟白碟壽司和一碗麵。此套餐僅在這段時間內供應。
- 雪糕有買 X 送 Y 的優惠。如果 $Y = 0$ ，則此特別優惠不生效。

每種優惠都可以多次使用。例如，如果愛麗絲和鮑伯購買 $2 \times X$ 杯雪糕，他們將免費獲得 $2 \times Y$ 杯雪糕。

現在，愛麗絲和鮑伯在 $H:M$ 到達壽司店。他們想點他們最喜歡的餐點。具體來說，他們想要至少 A_1 份編號為 1 的餐點，至少 A_2 份編號為 2 的餐點，如此類推，直到至少 A_5 份編號為 5 的餐點。你的任務是假設他們在到達時立即點餐，而且只點餐一次，計算購買他們想要的所有餐點的最低費用。

輸入

輸入的前 5 行每行包含一個整數。其中第 i 行 ($1 \leq i \leq 5$) 包含整數 A_i ，代表愛麗絲和鮑伯想要的編號為 i 的餐點的最少數量。

下一行包含兩個整數 H 和 M ，代表他們到達壽司店的時間，採用 24 小時制表示，且無前導零。

下一行包含兩個整數 X 和 Y ，代表雪糕優惠的詳細資訊。

輸出

輸出一個整數，即立即購買他們想要的所有餐點的最低費用。

提示

無需驗證輸入。不要輸出提示訊息。（例如：`請輸入 X 和 Y：`，`答案是：`）。不要輸出任何單位。

你可以使用以下樣例測試你的程式。你的輸出應與樣例輸出完全相同。

任務被劃分為子任務。部分正確的解法仍將獲得部分分數。

樣例

輸入	輸出
1 1 4 0 1 1 12 0 1 0	109

愛麗絲和鮑伯在 `12:00` 到達壽司店，且 $Y = 0$ ，因此兩個特別優惠都不生效。

最低總費用為 $\$10 + 4 \times \$12 + \$0 + \$33 + \$18 = \109 。

此樣例符合子任務 3、4、6 和 7 的限制。






$$\$10 + 4 \times \$12 + \$33 + \$18 = \$109$$

圖二：樣例一的圖例。

	輸入	輸出
2	4 0 0 2 0 15 12 1 0	100

只有下午茶套餐優惠生效。
他們應該購買 2 份下午茶套餐共 \$80，並分開購買 2 份白碟壽司共 \$20。總費用為 \$100。
此樣例符合子任務 4 和 7 的限制。



圖三：樣例二的圖例。

3	0 0 0 0 0 7 12 0 1 1	72
---	---	----

只有雪糕買 1 送 1 的優惠生效。
他們可以購買 4 份雪糕，並免費獲得 4 份雪糕。總費用為 $4 \times \$18 = \72 。
此樣例符合子任務 5、6 和 7 的限制。



圖四：樣例三的圖例。

4	1 0 0 1 6 12 0 3 2	115
---	--------------------------------------	-----

輸入

輸出

只有雪糕買 3 送 2 的優惠生效。
他們可以購買 4 份雪糕，並免費獲得 2 份雪糕，此外再分開購買一份白碟壽司和一碗麵。總費用為 $\$10 + \$33 + 4 \times \$18 = \115 。
此樣例符合子任務 6 和 7 的限制。

5

7	1072
20	
4	
5	
40	
16 48	
4 2	

下午茶套餐優惠和雪糕買 4 送 2 的優惠都生效。
此樣例符合子任務 7 的限制。

子任務

對於所有數據：
對於 $1 \leq i \leq 5$ ， $0 \leq A_i \leq 100$
 $10 \leq H \leq 21$
 $0 \leq M \leq 59$
 $1 \leq X \leq 100$
 $0 \leq Y \leq 100$

	估分	約束條件
1	9	$H:M = \boxed{12:00}$ $Y = 0$ $A_1 = 0, A_2 = 2, A_3 = 0, A_4 = 1, A_5 = 1$
2	9	$H:M = \boxed{15:00}$ $X = Y = 1$ $A_1 = 2, A_2 = 1, A_3 = 1, A_4 = 1, A_5 = 3$
3	15	$H:M = \boxed{12:00}$ $Y = 0$
4	16	$Y = 0$
5	18	$H:M = \boxed{12:00}$ $X = Y = 1$
6	25	$H:M = \boxed{12:00}$
7	8	無額外約束

G262 - 移除字母

執行時間限制: 1.000 s / 記憶體限制: 256 MB

鮑伯十分喜歡漂亮且三為一組的字串。對於三為一組的字串，當且僅當三個字串各自以字母順序排序後是相等時，我們稱之為漂亮的。

- $\boxed{AABC}$ 、 $\boxed{ACBA}$ 、 $\boxed{BAAC}$ 這組字串是漂亮的，因為在各自排序後，它們都等於 $\boxed{AABC}$ 。
- $\boxed{AABC}$ 、 $\boxed{ABCA}$ 、 $\boxed{BA}$ 這組字串並不漂亮，因為在各自排序後，它們分別等於 $\boxed{AABC}$ 、 $\boxed{AABC}$ 和 $\boxed{AB}$ ，並不相等。
- 注意，由三個空字串組成的一組也是漂亮的。

作為鮑伯的好朋友，你打算在他生日時送他一組漂亮的字串。

現在，你有三為一組由大寫字母組成的字串，分別為 S_1 、 S_2 和 S_3 ，這組字串可能是，亦可能不是漂亮的。為確保鮑伯喜歡你的禮物，你可以執行以下操作使 S_1 、 S_2 和 S_3 成為一組漂亮的字串：

- 選擇任意大寫字母 X 。
- 從三個字串中選擇兩個（ S_1 和 S_2 、 S_1 和 S_3 或 S_2 和 S_3 ）。
- 在選擇的兩個字串中移除所有字母 X 。注意，未被選擇的那個字串不會被改變。

例如，當 $S_1 = \boxed{ABBGCD}$ ， $S_2 = \boxed{AABBCD}$ ，且 $S_3 = \boxed{DCA}$ ，而你選擇在 S_1 和 S_2 中移除 $\boxed{A}$ ，那麼 S_1 和 S_2 分別變成 $\boxed{BBGCD}$ 和 $\boxed{BBCD}$ ，而 S_3 不變。當 $S_1 = \boxed{A}$ ， $S_2 = \boxed{B}$ ，且 $S_3 = \boxed{C}$ ，而你選擇在 S_1 和 S_3 中移除 $\boxed{C}$ ，那麼 S_3 變成一個空字串，而 S_1 和 S_2 不變。

請找出最少需要執行多少次操作才能使 S_1 、 S_2 和 S_3 變成一組漂亮的字串。

輸入

輸入總共包含 3 行，每行包含一個由大寫字母組成的字串，分別表示 S_1 、 S_2 和 S_3 。

輸出

輸出一個整數，表示最少需要執行多少次操作才能使 S_1 、 S_2 和 S_3 變成一組漂亮的字串。

樣例

	輸入	輸出
1	ABBGCD AABBCD DCA	4

可透過執行以下操作，使這組字串變得漂亮：

- 在 S_1 和 S_2 中移除 $\boxed{A}$ 。
- 在 S_1 和 S_3 中移除 $\boxed{A}$ 。
- 在 S_1 和 S_2 中移除 $\boxed{B}$ 。
- 在 S_1 和 S_2 中移除 $\boxed{G}$ 。

以字母順序排序後， S_1 、 S_2 和 S_3 都變成了 $\boxed{CD}$ 。

可以證明最少需要執行 4 次操作才能使這組字串變得漂亮。

	輸入	輸出
2	Z Z Z	0

這組字串原本就是漂亮的，不需要執行任何操作。

3	Q W E	3
---	--------------------------	--------------

可透過執行以下操作，使這組字串變得漂亮：

- 在 S_1 和 S_2 中移除 **Q**。
- 在 S_2 和 S_3 中移除 **W**。
- 在 S_1 和 S_3 中移除 **E**。

S_1 、 S_2 和 S_3 都變成了空字串。

可以證明最少需要執行 3 次操作才能使這組字串變得漂亮。

子任務

對於所有數據：

設 $|S|$ 為 S 的長度

$1 \leq |S_1|, |S_2|, |S_3| \leq 10^5$

S_1 、 S_2 和 S_3 由大寫字母組成

	佔分	約束條件
1	11	$S_2 = S_3$ $ S_1 = S_2 = S_3 = 1$
2	22	$S_2 = S_3$ $ S_1 = 2$ $ S_2 = S_3 = 1$
3	14	$S_2 = S_3$ 每個字母在一個字串中最多出現一次
4	21	每個字母在一個字串中最多出現一次
5	32	無額外約束

G263 - 螺絲

執行時間限制: 1.000 s / 記憶體限制: 256 MB

一場突如其來的停電事故在 HKGOI 期間發生！專業電工鮑伯被緊急召來搶修。

經過仔細檢查，鮑伯發現配電箱中有 N 個螺絲鬆開了，這些螺絲編號為 1 到 N 。為了修復故障，鮑伯必須按順序從螺絲 1 到螺絲 N 擰緊它們。第 i 個螺絲的直徑為 D_i 。

作為一名專業電工，鮑伯擁有 10^9 把螺絲刀，對應每一個從 1 到 10^9 的整數尺寸。

只有當螺絲刀的尺寸與螺絲的直徑兼容時，螺絲才能被擰緊。鮑伯知道容差閾值是 K ，這意味著當且僅當 $x - K \leq y \leq x + K$ 時，可以使用尺寸為 y 的螺絲刀擰緊直徑為 x 的螺絲。

鮑伯一次只能使用一把螺絲刀。擰緊螺絲不花費時間，然而，他可以隨時將當前使用的螺絲刀換成工具箱中任何其他螺絲刀。每一次更換都需要 1 分鐘來找到新的工具。選擇第一把螺絲刀不花費時間。

由於 HKGOI 的組織者希望比賽能盡快重新開始，請幫助鮑伯找到他擰緊所有螺絲所需的最短時間（以分鐘計）。

輸入

輸入的第一行包含一個整數 N ，表示需要擰緊的螺絲數量。

輸入的第二行包含一個整數 K ，表示容差閾值。

接下來的 N 行各包含一個整數，其中第 i 行包含整數 D_i ，表示第 i 個螺絲的直徑。

輸出

輸出一個整數，代表鮑伯擰緊所有螺絲所需的最短時間（以分鐘計）。

樣例

輸入		輸出	
1	7	3	
	2		
	2		
	3		
	1		
	5		
	6		
	100		
	3		

有 $N = 7$ 個鬆開的螺絲，其直徑依序為 $[2, 3, 1, 5, 6, 100, 3]$ 。容差閾值 K 等於 2。鮑伯可以透過以下方法，更換 3 次螺絲刀，在 3 分鐘內擰緊所有螺絲：

- 使用尺寸為 1 的螺絲刀，擰緊螺絲 1 到 3。
- 將尺寸為 1 的螺絲刀更換為尺寸為 6 的螺絲刀，並用它擰緊螺絲 4 到 5。
- 將尺寸為 6 的螺絲刀更換為尺寸為 100 的螺絲刀，並用它擰緊螺絲 6。
- 將尺寸為 100 的螺絲刀更換為尺寸為 3 的螺絲刀，並用它擰緊螺絲 7。

儘管存在其他方法來擰緊這些螺絲，可以證明 3 分鐘是所需的最短時間。
這個樣例符合子任務 6 的約束條件。

	輸入	輸出
2	2 6 1 6	0

鮑伯可以使用尺寸為 6 的螺絲刀來擰緊所有螺絲，因此他不需要進行任何更換。
這個樣例符合子任務 1、2、4 和 6 的約束條件。

3	6 2 1 3 4 10 15 17	2
---	---	--------------

這個樣例符合子任務 5 和 6 的約束條件。

4	4 0 2 1 1 2	2
---	--	--------------

這個樣例符合子任務 3 和 6 的約束條件。

子任務

對於所有數據：
 $1 \leq N \leq 10^5$
 $0 \leq K \leq 10^9$
對於 $1 \leq i \leq N$ ， $1 \leq D_i \leq 10^9$

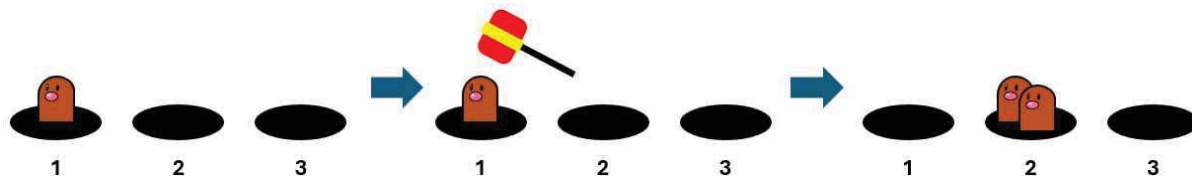
	估分	約束條件
1	7	對於 $1 \leq i \leq N$ ， $D_i \leq K$
2	9	$N = 2$
3	17	$K = 0$
4	24	答案必定是 0 或 1
5	23	對於 $2 \leq i \leq N$ ， $D_{i-1} \leq D_i$
6	20	無額外約束

G264 - 無盡打地鼠

執行時間限制: 1.000 s / 記憶體限制: 256 MB

愛麗絲正在玩一個新的街機遊戲，名為「無盡打地鼠」。

在這個遊戲中，地鼠可以出現在 $N + 1$ 個不同的洞，編號為 $1, 2, \dots, N + 1$ 。遊戲開始時，只有一隻地鼠在洞 1 中。每當愛麗絲敲打一隻在洞 i （其中 $1 \leq i \leq N$ ）中的地鼠時，它就會消失，並有兩隻新的地鼠出現在洞 $i + 1$ 中。你不能敲打在洞 $N + 1$ 中的地鼠。



愛麗絲敲打一隻在洞 1 中的地鼠。之後，它消失了，兩隻地鼠出現在洞 2 中。

當愛麗絲能夠令洞 N 中有剛好 M 隻地鼠，她便勝出此遊戲。請幫助她判斷她是否能勝出，如果可能，則求出最少所需的敲打次數。

輸入

輸入的第一行包含一個整數 N ，計算地鼠數量的洞。

輸入的第二行包含一個整數 M ，愛麗絲勝出所需的地鼠數量。

輸出

如果無法令洞 N 中有剛好 M 隻地鼠，則輸出 -1 。否則，輸出最少所需的敲打次數。

樣例

	輸入	輸出
1	5 6	7

愛麗絲可以按以下順序敲打地鼠，使得在洞 5 中剛好有 6 隻地鼠：

- 使用一次敲打，敲打在洞 1 中的一隻地鼠。隨後兩隻地鼠出現在洞 2 中。
- 使用一次敲打，敲打在洞 2 中的一隻地鼠。隨後兩隻地鼠出現在洞 3 中。
- 使用兩次敲打，敲打在洞 3 中的兩隻地鼠。隨後四隻地鼠出現在洞 4 中。
- 使用三次敲打，敲打在洞 4 中的三隻地鼠。隨後六隻地鼠出現在洞 5 中，愛麗絲勝出遊戲。

我們可以證明這是愛麗絲贏得遊戲所需的最少敲打次數。最後兩步需要用到多次的敲打，因為每次我們只能敲打一隻地鼠。

	輸入	輸出
2	3 3	4

愛麗絲可以按以下順序敲打地鼠，使得在洞 3 中剛好有 3 隻地鼠：

- 使用一次敲打，敲打在洞 1 中的一隻地鼠。隨後兩隻地鼠出現在洞 2 中。
- 使用一次敲打，敲打在洞 2 中的一隻地鼠。隨後兩隻地鼠出現在洞 3 中。
- 使用一次敲打，敲打在洞 3 中的一隻地鼠。隨後兩隻地鼠出現在洞 4 中，而洞 3 剩餘一隻地鼠。
- 使用一次敲打，敲打在洞 2 中的一隻地鼠。隨後兩隻地鼠出現在洞 3 中。洞 3 總共有三隻地鼠，因此愛麗絲勝出遊戲。

我們可以證明這是愛麗絲贏得遊戲所需的最少敲打次數。

3	8 5	11
4	3 10	-1
5	1 1	0

子任務

對於所有數據：

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

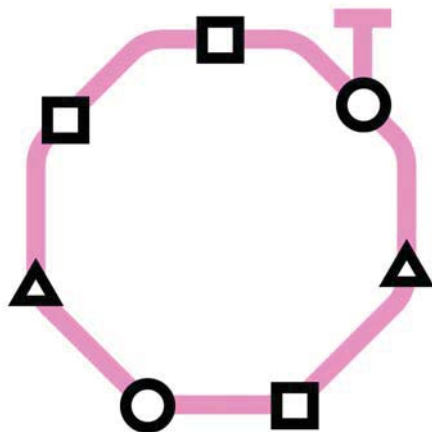
	佔分	約束條件
1	13	$M \leq 2$
2	22	$N, M \leq 1000$
3	18	$M = 2^k$ ，其中 k 為整數 愛麗絲必定能勝出
4	13	$M = 2^k$ ，其中 k 為整數
5	16	$N \leq 10^5$
6	18	無額外約束

G265 - 迷你地鐵

執行時間限制: 1.000 s / 記憶體限制: 256 MB

鮑伯是遊戲迷你地鐵的忠實粉絲。今天，他想設計一條效率最高的地鐵路線。

在玩了數小時的遊戲之後，鮑伯設定了一條有 N 個車站的環形地鐵路線。車站按順時針順序編號從 1 到 N ，地鐵沿著： $1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow N \rightarrow 1 \rightarrow \dots$ 的路線停靠這 N 個車站。地鐵將永遠沿著這條環形路線運行。地鐵從一個車站行駛到另一個車站需要 1 個單位時間。



乘客可以分為 K 種不同類型，編號從 1 到 K 。相應地，車站也可以分為 K 種不同類型。乘客可以在 N 個車站中的任意一個車站上車搭乘地鐵。類型 i 的乘客會在地鐵到達第一個為類型 i 的車站時下車。如果類型 i 的乘客在一個類型為 i 的車站開始，他們將完全不搭乘地鐵。

目前，鮑伯必須為這 N 個車站分配類型。他必須分配 A_1 個車站為類型 1， A_2 個車站為類型 2， $\dots$ ，依此類推。保證 A_i 的總和恰好為 N 。

你的目標是幫助鮑伯最佳地分配每個車站的類型，使得考慮所有可能的起始車站和所有乘客類型後，任何乘客在最壞情況需要停留在地鐵上的最長時間為最少。

輸入

輸入的第一行包含兩個整數 N 和 K 。

第二行包含 K 個整數 $A_1, A_2, \dots, A_K$ ，分別是每種車站類型所需的數量。

輸出

在第一行，輸出一個整數 T ，在最佳的車站類型分配下，任何乘客需要停留在地鐵上的最長時間。

然後輸出包含 N 個整數的一行。第 i 個整數 B_i 應該是分配給車站 i 的類型。

如果存在多個最佳的車站類型分配，您可以輸出其中任何一個。

樣例

	輸入	輸出						
1	<table><tr><td>7</td><td>3</td></tr><tr><td>2</td><td>2 3</td></tr></table>	7	3	2	2 3	<table><tr><td>3</td></tr><tr><td>1 2 3 3 1 2 3</td></tr></table>	3	1 2 3 3 1 2 3
7	3							
2	2 3							
3								
1 2 3 3 1 2 3								

- 一個類型 1 的乘客從車站 2 上車，將在車站 5 下車。
所採取的路線是 $2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ，耗時 3 單位時間。
- 一個類型 3 的乘客從車站 2 上車，將在車站 3 下車。
所採取的路線是 $2 \rightarrow 3$ ，耗時 1 單位時間。
- 一個類型 2 的乘客從車站 2 上車，將完全不搭乘地鐵，耗時 0 單位時間。

可以證明，在這個分配的最壞情況下，沒有任何乘客會在地鐵上停留超過 3 單位時間。並且，不存在一種車站類型分配，能讓所有乘客在地鐵上花費的時間都少於 3 單位時間。

2	7 3	3
	2 2 3	2 3 1 3 2 1 3

這是以上樣例的另一種最佳車站類型分配。

3	6 2	5
	5 1	2 1 1 1 1 1

4	3 3	2
	1 1 1	3 1 2

子任務

對於所有數據：

$$2 \leq K \leq N \leq 100000$$

對於所有 $1 \leq i \leq K$ ， $1 \leq A_i \leq N$

$$A_1 + A_2 + \cdots + A_K = N$$

	估分	約束條件
1	7	$K = 2$ $A_1 = A_2$
2	10	$A_1 = A_2 = \cdots = A_K$
3	17	$K = 2$ $A_1 \leq A_2 \leq 2 \times A_1$
4	22	$K = 2$
5	23	$A_1 \leq A_2 \leq \cdots \leq A_K \leq 2 \times A_1$
6	21	無額外約束

G266 - 全速前進

執行時間限制: 1.000 s / 記憶體限制: 256 MB

愛麗絲住在 HKGOI 的世界。這個世界可以用一條一維數線來表示。愛麗絲的家位於數線上的點 0 ，而傳聞在數線上的點 $D > 0$ 處藏著一塊黃金。

愛麗絲有一輛汽車，讓她可以從點 0 開車到點 D 。她的汽車每小時的最大速度為 M ，而在連續兩個小時之間最大的加速度或減速度為每小時 A 。

不幸的是，愛麗絲的汽車相當破舊，有諸多行駛上的限制。它不能倒車，而且只能以非負整數的速度行駛。此外，由於汽車的製造過程出錯，當愛麗絲在點 0 啟動引擎時，汽車的預設速度會被自動設定為每小時 S_0 （這可能為 0 ，也可能不為 0 ）。

愛麗絲迫不及待地想要拿到那塊黃金，所以她想知道她是否有可能開車到達點 D 並在那裏把車完全停下。如果這是可能的，她還想知道這樣的行程所需要的最短時間。

形式化地，令 S_i 表示愛麗絲的汽車在第 $i \geq 1$ 個小時的速度，那麼一個從點 0 到點 D 、持續 T 小時的行程應當滿足以下所有約束：

- **速度限制**：速度必須總是一個不大於 M 的非負整數，即對於所有 $i \geq 1$ ，必須滿足 $0 \leq S_i \leq M$ ；
- **有限加速**：連續兩個小時之間的速度變化必須不大於 A ，即對於所有 $i \geq 1$ ，必須滿足 $-A \leq S_{i-1} - S_i \leq A$ （特別地，當 $i = 1$ 時，從初始速度 S_0 的變化也必須不大於 A ）；
- **完全停下**： T 小時的行程後，汽車必須完全停止，這意味著速度必須能夠在 T 小時之後立即降為 0 ，亦即 $S_T \leq A$ ；
- **正確總距離**：車輛必須恰好到達點 D ，即 $S_1 + S_2 + \dots + S_T = D$ 。

還有更多傳聞說，如果你能幫愛麗絲計算出從點 0 到點 D 所需要的最短時間，她或許會將那塊黃金鑄造成一枚 HKGOI 獎牌送給你作為禮物。嗯.....

輸入

輸入的第一行包含一個正整數 D ，表示愛麗絲駕駛的距離。

輸入的第二行包含一個非負整數 S_0 ，表示汽車的預設速度。

輸入的第三行包含一個正整數 M ，表示汽車的最大速度。

輸入的第四行包含一個正整數 A ，表示汽車最大的加速度和減速度。

輸出

如果愛麗絲不可能在滿足所有限制的情況下從點 0 開車到點 D ，那麼輸出 -1 。

否則，如果存在滿足所有限制的行程，則輸出一個正整數，表示到達點 D 並把汽車完全停下所需要的最短時間（以小時計）。

樣例

	輸入	輸出
1	7 2 10 2	3

以下是其中一個可行的行程，使得愛麗絲用 3 小時到達點 7 並完全停下：在第一個小時以速度 3 行駛，在第二個小時以速度 2 行駛，在第三個小時以速度 2 行駛，之後把車完全停下。
可以證明愛麗絲需要最少 3 小時才能到達點 7 並停下。

2	20 4 7 3	4
---	---------------------------------	--------------

愛麗絲可以用以下方法，在 4 小時後達到點 20：在第一個小時以速度 5、第二個小時以速度 7、第三個小時以速度 5、第四個小時以速度 3 行駛。

3	14 6 10 1	-1
---	----------------------------------	---------------

愛麗絲不可能在到達點 14 時同時在那裏把車完全停下。

子任務

對於所有數據：

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

$0 \leq S_0 \leq 10^9$

$1 \leq A \leq 10^9$

$\max(S_0, A) \leq M \leq 10^9$

	佔分	約束條件
1	8	$M = A$
2	14	$S_0 = M$ $A = 1$ $D, M \leq 10^6$
3	20	$S_0 = 0$ $A = 1$ $D, M \leq 10^6$
4	19	$S_0 = 0$
5	21	$A = 1$
6	18	無額外約束