

1 Pick Precious Pairs

In a small fun game, there are T types of precious stones found in a box. For every type T_i , there are N_i stones in the box. The challenge is without looking into the box, need to pick stones from the box and form pairs of stones of similar type.

Given the number of types of stones, number of pairs need to form and the total stones of each type, find the minimum number of stones that must be picked to form at least P number of pairs of similar types of stones. Return -1 if it is not possible to form pairs.

Input/Output

Input	Output	Comments																																																						
<div>2 3</div> <div>10 9</div>	7	<div><ul style="list-style-type: none">First line 2 3 corresponds to <i>number of types of stones</i> and <i>number of pairs to form</i>.Second line 10 9 corresponds to the total number of stones of each type. <i>Type1: 10, Type2: 9</i>To form 3 pairs of stones, we need a minimum of 7 stones to be sure of forming 3 pairs. The table below gives the details.</div> <table><tr><th>Pickup Scenarios</th><th>Type-1 Stones</th><th>Type-2 Stones</th><th>Type-1 Pairs</th><th>Type-2 Pairs</th><th>Total Pairs</th></tr><tr><td>A</td><td>7</td><td>0</td><td>3</td><td>0</td><td>3</td></tr><tr><td>B</td><td>6</td><td>1</td><td>3</td><td>0</td><td>3</td></tr><tr><td>C</td><td>5</td><td>2</td><td>2</td><td>1</td><td>3</td></tr><tr><td>D</td><td>4</td><td>3</td><td>2</td><td>1</td><td>3</td></tr><tr><td>E</td><td>3</td><td>4</td><td>1</td><td>2</td><td>3</td></tr><tr><td>F</td><td>2</td><td>5</td><td>1</td><td>2</td><td>3</td></tr><tr><td>G</td><td>1</td><td>6</td><td>0</td><td>3</td><td>3</td></tr><tr><td>H</td><td>0</td><td>7</td><td>0</td><td>3</td><td>3</td></tr></table>	Pickup Scenarios	Type-1 Stones	Type-2 Stones	Type-1 Pairs	Type-2 Pairs	Total Pairs	A	7	0	3	0	3	B	6	1	3	0	3	C	5	2	2	1	3	D	4	3	2	1	3	E	3	4	1	2	3	F	2	5	1	2	3	G	1	6	0	3	3	H	0	7	0	3	3
Pickup Scenarios	Type-1 Stones	Type-2 Stones	Type-1 Pairs	Type-2 Pairs	Total Pairs																																																			
A	7	0	3	0	3																																																			
B	6	1	3	0	3																																																			
C	5	2	2	1	3																																																			
D	4	3	2	1	3																																																			
E	3	4	1	2	3																																																			
F	2	5	1	2	3																																																			
G	1	6	0	3	3																																																			
H	0	7	0	3	3																																																			
<div>3 5</div> <div>3 3 4</div>	-1	<div>Because, there is no way 5 pairs of stones can be formed. Only 4 pairs are possible with the given number of stones of each type. (1 pair of type-1 + 1 pair of type-2 + 2 pairs of type-3)</div>																																																						

2 Selling Ganesh Idols

A noted artist from Dhoolpet, Sundar Kalakar, makes Ganesh idols of different sizes and models. Initially, all idols are in white color later he paint, decorate and sells those idols to make money. Sundar defines decoration theme, a combination of different color of paints and variety of decorative items. The theme has several levels (extensions). The highest level is MAXLEVEL. He must spend some money to paint idol with a certain theme level. Theme *Level 1 (all white)* has no cost (0). Sundar spends and collects money in **Coin (s)** only. He is very greedy so wants to spend less and earn more.

To enhance or change the current level (**curThemeLevel**) of decorative theme of idol(s) to another level, say **curThemeLevel+1**, he must spend R_{cd} coins and he can sell the idol with that level of theme for S_{cd} coins. During this process, he doesn't want to borrow any money from others so he should always have 0 or more coins with him.

Given the number of idols, maximum level of decoration, initial number of coins, followed by the cost of levels of decoration themes (starting from level-2), selling prices of idols for respective level of themes, find the maximum number of coins he could possibly possess after selling those N idols.

Sample Input/Output

Input	Output	Comments
6 3 1 4 8 2 10 16	29	<ul style="list-style-type: none">The first line 6 3 1 corresponds to the number of idols, the maximum level of theme, and the initial number of coins.The second line 4 8 corresponds to the costs of 2nd and 3rd levels of decoration theme.The third line corresponds to 2 10 16 corresponds to the selling prices (in coins) for every level of decoration. <i>Selling Price of idol with theme level-1 is 2 coins, level-2 is 10 coins and level-3 is 16 coins.</i>The table below illustrates the reason of the answer.

Total Idols: 6 Initial Coins: 1						
Strategy #	Step #		Theme Level	Total Coins Spent	Total Coins Earned	Total Coins Remaining
1	1	Sell 2 idols	1	0	$2 * 2 = 4$	$1 + 4 = 5$
	2	Change 1 idol and Sell	2	$1 * 4 = 4$	$1 * 10 = 10$	$(5 - 4) + 10 = 11$
	3	Change 2 idols and sell	2	$2 * 4 = 8$	$2 * 10 = 20$	$(11 - 8) + 20 = 23$
	4	Change 1 idol and sell	3	$1 * (4+8) = 12$	$1 * 16 = 16$	$(23 - 12) + 16 = 27$
2	1	Sell 2 idols	1	0	$2 * 2 = 4$	$1+4 = 5$
	2	Change 1 idols and Sell	2	$1 * 4 = 4$	$1 * 10 = 10$	$(5 - 4) + 10 = 11$
	3	Change 1 idols and sell	2	$1 * 4 = 4$	$1 * 10 = 10$	$(11 - 4) + 10 = 17$
	4	Change 1 idol and sell	3	$1 * (4+8) = 12$	$1 * 16 = 16$	$(17 - 12) + 16 = 21$

n

3 Attractive Coupons

In order to increase seasonal sales, an online stores has announced attractive cashback coupons, numbered 1 to N, to its regular customers. Every coupon has *wait_hour* and *expiry_hour*.

Unbelievable cashbacks for a coupon are given only after its wait-hours have elapsed and has not expired yet. Multiple coupons can be redeemed in any order but only one coupon can be redeemed at any given time. There is Customer Cumulative Wait Hours (CCWH) that is the cumulative sum of wait-time of all coupons redeemed by a customer. To redeem a coupon, CCWH must less or equal to expiry-hour of that particular coupon.

Given the number of coupons, wait-hour and expiry hour of each coupon, write a program to find the maximum number of coupons that can be redeemed by a customer.

Input/Output

Input	Output	Comments
5 10 20 25 50 60 75 20 80 30 90	4	<ul style="list-style-type: none"> First line 5 corresponds to number of coupons Next five (5) lines has <i>wait_hours</i> and <i>expiry_hour</i> of every coupon <p>As the coupon can be redeemed in any order, here are the most optimal ways to redeemed as many coupons as possible:</p> <ol style="list-style-type: none"> First, take the 1st coupon, it costs 10 hours so the customer will redeem it on the 10th hour, and ready to take the next coupon on the 11th hour. Second, take the 2nd coupon, it costs 25 hours so the customer will redeem it on the 25th hour, and ready to take the next coupon on the 26th hour. Third, take the 4th coupon, it costs 20 hours so the customer will redeem it on the 45th hour, and ready to take the next coupon on the 46th hour. Fourth, take the 5th coupon, it costs 30 hours so the customer will redeem it on the 75th hour, and ready to take the next coupon on the 76th hour. The 3rd coupon cannot be taken now, since the customer will finish it on the 135th hour, which exceeds the expiry hour 75. <p>So, the customer can take 4 coupons at most</p>
3 5 4 6 4 8 7	0	<p>Customer can't take any coupon, because the wait_hour of all coupons is greater than the expiry_hour</p>