Problem A. Salad Bar

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

Bytea went to a salad bar. There are n fruits one next to another on the bar counter. Specifically, these are apples and oranges. Bytea can pick any contiguous part of the line of fruits for her salad.

The fruits she chooses will be added to the salad either from left to right or from right to left. As Bytea loves oranges, she requires that throughout the salad making process, the number of oranges in it should always be no smaller than the number of apples, regardless of whether these are added from left to right or from right to left. Help Bytea out by writing a program that will find the longest contiguous part of the line of fruits that satisfies her requirements.

Input

The first line of the standard input contains a single integer n ($1 \leq n \leq 1\,000\,000$), denoting the number of fruits. The next line contains a string of n characters $a_1 a_2 \ldots a_n$ ($a_i \in \{\texttt{j}, \texttt{p}\}$). These stand for Polish names of apples and oranges: *jabłka* and *pomarańcze*). Consequently, if $a_i = \texttt{j}$, then the i-th fruit in a line is an apple, and otherwise it is an orange.

Output

The first and only line of the standard output should contain a single integer equal to the number of fruits in the longest contiguous part of the line that satisfies Bytea's requirements. Note that it could be the case that 0 is the correct result.

Example

standard input	standard output
6 jpjppj	4

Note

Once the leftmost and the rightmost apples are discarded, Bytea can order a salad out of all remaining fruits.

Problem B. Hotels

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

There are n towns in Byteotia, connected with only $n-1$ roads. Each road directly links two towns. All the roads have the same length and are two way. It is known that every town can be reached from every other town via a route consisting of one or more (direct-link) roads. In other words, the road network forms a *tree*.

Byteasar, the king of Byteotia, wants three luxury hotels erected to attract tourists from all over the world. The king desires that the hotels be in different towns and at the same distance one from each other.

Help the king out by writing a program that determines the number of possible locations of the hotel triplet in Byteotia.

Input

The first line of the standard input contains a single integer n ($1 \le n \le 100\,000$), the number of towns in Byteotia. The towns are numbered from 1 to n.

The Byteotian road network is then described in $n-1$ lines. Each line contains two integers a and b ($1 \le a < b \le n$) , separated by a single space, that indicate there is a direct road between the towns a and b.

Output

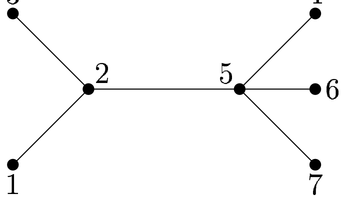
The first and only line of the standard output should contain a single integer equal to the number of possible placements of the hotels.

Example

standard input	standard output

7
1 2
5 7
2 5
2 3
5 6
4 5
	5

Note



The triplets (unordered) of towns where the hotels can be erected are: $\{1,3,5\}$, $\{2,4,6\}$, $\{2,4,7\}$, $\{2,6,7\}$, $\{4,6,7\}$.

Problem C. Bricks

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	64 mebibytes

Little Bitek and his friends spent the whole yesterday playing with colorful bricks in the kindergarten. Initially they made building models but they quickly got bored with that. Then they decided to place the bricks in a line, one after another. To avoid a dull look, they tried to avoid putting two bricks of the same color next to each other. After a long while they succeeded in placing all the bricks while observing this rule. Then the day care was over, and the children went home with their parents.

Today Bitek came to the kindergarten early. He was satisfied to see that their yesterday's creation was intact. But then he tripped over in a most unfortunate way, falling right on the line of bricks, which all mixed in a pile. The boy quickly sorted them by color and wondered how best to quickly reassemble the perfect line. Luckily, he managed to recall what were the colors of the two bricks on both ends of the line.

Help little Bitek out and tell him how to order the bricks in a line so that no two bricks of the same color are next to each other and the bricks at the ends of the line have the colors that he recalled. Note that Bitek could have made a mistake in recalling the two colors or perhaps he did not find some of the blocks after falling into them, so the reconstruction might not be possible.

Input

There are three integers in the first line of the standard input, k, p, and q ($1 \le k \le 1\,000\,000$, $1 \le p, q \le k$), separated by single spaces, denoting the number of brick colors, and the colors of the first and the last brick in the desired arrangement, respectively. In the second line, there are k integers, $i_1, i_2, \ldots, i_k$ ($1 \le i_j \le 1\,000\,000$), separated by single spaces. The number i_j signifies that Bitek has exactly i_j bricks of color j. You may assume that the total number of bricks does not exceed one million, i.e., that $n = i_1 + i_2 + \ldots + i_k \le 1\,000\,000$.

Output

Your program should print n integers to the standard output, separated by single spaces. The numbers should represent the colors of successive bricks in an arrangement that satisfies aforementioned constraints. If no such arrangement exists, your program should print only a single integer: 0.

If there are several correct answers, your program can pick one arbitrarily.

Example

standard input	standard output

`3 3 1`
`2 3 3`
	`3 2 3 1 2 3 2 1`

`3 3 1`
`2 4 2`
	`0`

Note

In the first example, another correct arrangement is

`3 1 2 3 2 3 2 1.`

In the second example Bitek must have made a mistake — it is impossible to arrange the bricks so that they satisfy the constraints.

Problem D. Couriers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	128 mebibytes

Byteasar works for the BAJ company, which sells computer games. The BAJ company cooperates with many courier companies that deliver the games sold by the BAJ company to its customers. Byteasar is inspecting the cooperation of the BAJ company with the couriers. He has a log of successive packages with the courier company that made the delivery specified for each package. He wants to make sure that no courier company had an unfair advantage over the others.

If a given courier company delivered more than half of all packages sent in some period of time, we say that it *dominated* in that period. Byteasar wants to find out which courier companies dominated in certain periods of time, if any.

Help Byteasar out! Write a program that determines a dominating courier company or that there was none.

Input

The first line of the standard input contains two integers, n and m ($1 \leq n, m \leq 500\,000$), separated by a single space, that are the number of packages shipped by the BAJ company and the number of time periods for which the dominating courier is to be determined, respectively. The courier companies are numbered from 1 to (at most) n.

The second line of input contains n integers, $p_1, p_2, \ldots, p_n$ ($1 \leq p_i \leq n$), separated by single spaces; p_i is the number of the courier company that delivered the i-th package (in shipment chronology).

The m lines that follow specify the time period queries, one per line. Each query is specified by two integers, a and b ($1 \leq a \leq b \leq n$), separated by a single space. These mean that the courier company dominating in the period between the shipments of the a-th and the b-th package, including those, is to be determined.

Output

The answers to successive queries should be printed to the standard output, one per line. (Thus a total of m lines should be printed.) Each line should hold a single integer: the number of the courier company that dominated in the corresponding time period, or 0 if there was no such company.

Example

standard input	standard output

7 5
1 1 3 2 3 4 3
1 3
1 4
3 7
1 7
6 6
	1
	0
	3
	0
	4

Problem E. Solar lamps

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	128 mebibytes

Byteasar has a large and pretty garden. As he would like to be able to appreciate its beauty even after dusk, he installed lamps across the garden.

The lamps are directional, i.e., they illuminate only a certain angle, common to them all. Moreover, Byteasar has aligned them so that they all face the same direction. Last but not least, these are solar lamps, i.e., they come with solar panels but no batteries, strangely enough! You might think the panels are thus useless, and each lamp will require electricity at night, but not quite: A lamp will produce light if a sufficient number of lamps illuminate it.

By now, Byteasar has even come up with an order he is going supply the lamps with electricity, thus turning them on. For simplicity, we number the lamps from 1 to n in this order, i.e., the lamp no. i is supplied with electricity at time i. The only thing left for Byteasar (and you, of course!) is to figure out when exactly each lamp will start emitting light. Help Byteasar by writing a program that will determine the answer to this question.

Input

The first line of the standard input contains a single integer n ($1 \le n \le 200\,000$): the number of lamps Byteasar installed. In the second line of input, there are four integers X_1, Y_1, X_2, Y_2 ($-10^9 \le X_i, Y_i \le 10^9$, $(X_i, Y_i) \neq (0, 0)$), separated by single spaces, that describe the area illuminated by every lamp. Namely, if there is a lamp located at the point (x, y), then it illuminates the area (together with its edge) within the smaller of the two angles formed by two rays that both originate at (x, y) such that the i-th (for $i = 1, 2$) ray passes through $(x + X_i, y + Y_i)$. This angle is always smaller than 180 degrees.

The n input lines that follow specify the locations of the lamps: the i-th such line contains two integers x_i, y_i, ($-10^9 \le x_i, y_i \le 10^9$) separated by a single space, that indicate that the lamp no. i is located at the point (x_i, y_i). You may assume that no two lamps share their location.

The last line of the input contains n integers $k_1, k_2, \ldots, k_n$ ($1 \le k_i \le n$), separated by single spaces, that signify that if the lamp no. i is in the area illuminated by at least k_i other lamps, then it will start emitting light as well.

Output

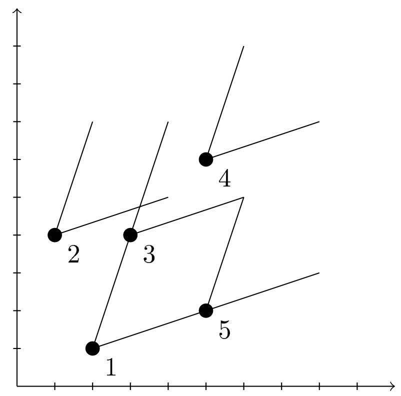
Your program should print out to the standard output a single line with n integers $t_1, \ldots, t_n$, separated by single spaces. The number t_i should be the time when the lamp no. i starts producing light.

Example

standard input	standard output

5
3 1 1 3
2 1
1 4
3 4
5 6
5 2
1 2 1 3 2
	1 2 1 2 5

Note



At time 1 Byteasar powers on the lamp 1, which also causes the lamp 3 to produce light. Once the lamp 2 is powered on, the lamp 4 begins to emit light (being illuminated by lamps 1, 2, and 3).

Problem F. Solar panels

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	128 mebibytes

Having decided to invest in renewable energy, Byteasar started a solar panels factory. It appears that he has hit the gold as within a few days n clients walked through his door. Each client has ordered a single rectangular panel with specified width and height ranges.

The panels consist of square photovoltaic cells. The cells are available in all integer sizes, i.e., with the side length integer, but all cells in one panel have to be of the same size. The production process exhibits economies of scale in that the larger the cells that form it, the more efficient the panel. Thus, for each of the ordered panels, Byteasar would like to know the maximum side length of the cells it can be made of.

Input

The first line of the standard input contains a single integer n ($1 \leq n \leq 1000$): the number of panels that were ordered. The following n lines describe each of those panels: the i-th line contains four integers $s_{min}, s_{max}, w_{min}, w_{max}$ ($1 \leq s_{min} \leq s_{max} \leq 10^9$, $1 \leq w_{min} \leq w_{max} \leq 10^9$), separated by single spaces; these specify the minimum width, the maximum width, the minimum height, and the maximum height of the i-th panel respectively.

Output

Your program should print exactly n lines to the standard output. The i-th line is to give the maximum side length of the cells that the i-th panel can be made of.

Example

standard input	standard output

4
3 9 8 8
1 10 11 15
4 7 22 23
2 5 19 24
	8
	7
	2
	5

Note

Byteasar will produce four solar panels of the following sizes: 8×8 (a single cell), 7×14 (two cells), 4×22 or 6×22 (22 or 33 cells respectively), and 5×20 (four cells).

Problem G. Criminals

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	64 megabytes

Byteburg is a beautiful town by a river. There are n houses along the river, numbered downstream with successive integers from 1 to n. Byteburg used to be a nice quiet town in which everyone was happy. Alas, this changed recently, as two dangerous criminals – Bitie and Bytie set up shop in it. They did so many robberies already that the citizens are afraid to leave their houses.

Bitie and Bytie do no mere burglaries but rather whole raids: each time they leave their houses and walk towards each other, never turning back. Bitie walks downstream (towards larger numbers) while Bytie walks upstream (towards smaller numbers). Along the way, before they meet, each chooses several houses to break into and steal precious items (and vital data). After the robberies they meet in a house and divide their loot. Byteburgers are sick of this already – they would all rather have their tranquility restored. So they asked the detective Bythony for help.

The detective established that the bandits live in houses of the same color but he does not know which one. Just a moment ago, an anonymous tip claimed that the robbers are on a raid. Fearing for their own safety, the source did not say which houses will be broken into. They did however specify their colors. As it turns out, the bandits are quite superstitious – each of them will rob a house of each color at most once.

Bythony can wait no longer. He intends to ambush the criminals at their meeting place. Aid Bythony in his undertaking by writing a program to find all possible meeting places of the robbers.

Input

There are two integers in the first line of the standard input, n and k ($3 \le n \le 1\,000\,000$, $1 \le k \le 1\,000\,000$, $k \le n$), separated by a single space, that specify the number of houses and the number of house colors in Byteburg respectively. The colors are number with successive integers from 1 to k. In the second line of input, there is a sequence of n integers, $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le k$), separated by single spaces. These are the colors of successive houses in Byteburg.

In the third line of input, there are two integers m and l ($1 \le m, l \le n$, $m + l \le n - 1$), separated by a single space, specifying the numbers of houses (to be) broken into by Bitie and Bytie respectively. In the fourth line of input, there are m pairwise different integers $x_1, x_2, \ldots, x_m$ ($1 \le x_i \le k$), separated by single spaces. These are the colors of houses robbed by Bitie in the order of being broken into (i.e., excluding Bitie's house). In the fifth, which is the last, line of input, there are l pairwise different integers $y_1, y_2, \ldots, y_l$ ($1 \le y_i \le k$), separated by single spaces. These are the colors of houses robbed by Bytie in the order of being broken into (again, these do not include Bytie's house). Moreover, $x_m = y_l$ is the color of the house in which the robbers will divide the plunder. (Clearly, they have to break into that one as well!)

Output

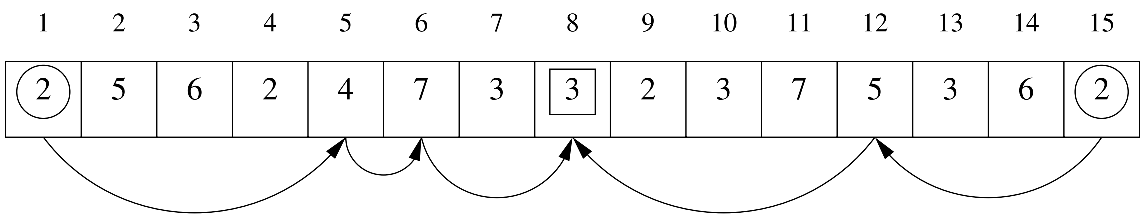
Your program it to print exactly two lines to the standard output. The first of those should give the number of houses in which the criminals can meet while respecting aforementioned constraints. The second line should contain the increasing sequence of the numbers of those houses, separated by single spaces. If the robbers cannot meet at all, the first line should contain the number 0 while the second one should be empty.

Example

standard input	standard output

15 7
2 5 6 2 4 7 3 3 2 3 7 5 3 6 2
3 2
4 7 3
5 3
	3
	7 8 10

Note



In above example, the bandits may live in houses of color 2 (Bitie in the house no. 1 or 4, Bytie in the house no. 15) or 6 (Bitie in the house no. 3, Bytie in the house no. 14). Whether he lived in the house no. 1 or 4, Bitie could rob the following houses: 5 (of color 4), 6 (of color 7), and then either 7, 8 or 10 (of color 3). Bitie could rob the following houses: 12 (of color 5), meeting Bitie afterwards in either of 7, 8 or 10 (of color 3). The figure above depicts a scenario in which Bitie lives in the house no. 1 and the robbers meet in the house no. 8.

Problem H. Rally

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	128 mebibytes

An annual bicycle rally will soon begin in Byteburg. The bikers of Byteburg are natural long distance cyclists. Local representatives of motorcyclists, long feuding the cyclists, have decided to sabotage the event.

There are n intersections in Byteburg, connected with one way streets. Strangely enough, there are no cycles in the street network – if one can ride from intersection u to intersection v, then it is definitely impossible to get from v to u.

The rally's route will lead through Byteburg's streets. The motorcyclists plan to ride their blazing machines in the early morning of the rally day to one intersection and completely block it. The cyclists' association will then of course determine an alternative route but it could happen that this new route will be relatively short, and the cyclists will thus be unable to exhibit their remarkable endurance. Clearly, this is the motorcyclists' plan – they intend to block such an intersection that the longest route that does not pass through it is as short as possible.

Input

In the first line of the standard input, there are two integers, n and m ($2 \leq n \leq 500\,000$, $1 \leq m \leq 1\,000\,000$), separated by a single space, that specify the number of intersections and streets in Byteburg. The intersections are numbered from 1 to n. The m lines that follow describe the street network: in the i-th of these lines, there are two integers, a_i, b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), separated by a single space, that signify that there is a one way street from the intersection no. a_i to the one no. b_i.

Output

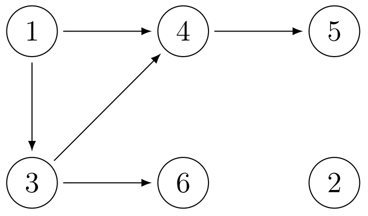
The first and only line of the standard output should contain two integers separated by a single space. The first of these should be the number of the intersection that the motorcyclists should block, and the second – the maximum number of streets that the cyclists can then ride along in their rally. If there are many solutions, your program can choose one of them arbitrarily.

Example

standard input	standard output

6 5
1 3
1 4
3 6
3 4
4 5
	1 2

Note



$$\begin{tikzcd} 1 \arrow[r] \arrow[d] & 4 \arrow[r] & 5 \\ 3 \arrow[ur] \arrow[r] & 6 & 2 \end{tikzcd}$$

Problem I. Supercomputer

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

Byteasar has designed a supercomputer of novel architecture. It may comprise of many (identical) processing units. Each processing unit can execute a single instruction per time unit.

The programs for this computer are not sequential but rather have a tree structure. Each instruction may have zero, one, or multiple *subsequent instructions*, for which it is the *parent instruction*.

The instructions of the program can be executed in parallel on all available processing units. Moreover, they can be executed in many orders: the only restriction is that an instruction cannot be executed unless its parent instruction has been executed before. For example, as many subsequent instructions of an instruction that has been executed already can be executed in parallel as there are processing units.

Byteasar has a certain program to run. Since he likes utilizing his resources optimally, he is wondering how the number of processing units would affect the running time. He asks you to determine, for a given program and number of processing units, the minimum execution time of the program on a supercomputer with this many processing units.

Input

In the first line of standard input, there are two integers, n and q ($1 \le n, q \le 1\,000\,000$), separated by a single space, that specify the number of instructions in Byteasar's program and the number of running time queries (for different numbers of processing units).

In the second line of input, there is a sequence of q integers, $k_1, k_2, \ldots, k_q$ ($1 \le k_i \le 1\,000\,000$), separated by single spaces: k_i is the number of processing units in Byteasar's i-th query.

In the third and last input line, there is a sequence of $n-1$ integers, $a_2, a_3, \ldots, a_n$ ($1 \le a_i < i$), separated by single spaces: a_i specifies the number of the parent instruction of the instruction number i. The instructions are numbered with successive integers from 1 to n, where the instruction no. 1 is the first instruction of the program.

Output

Your program should print one line consisting of q integers, separated by single spaces, to the standard output: the i-th of these numbers should specify the minimum execution time of the program on a supercomputer with k_i processing units.

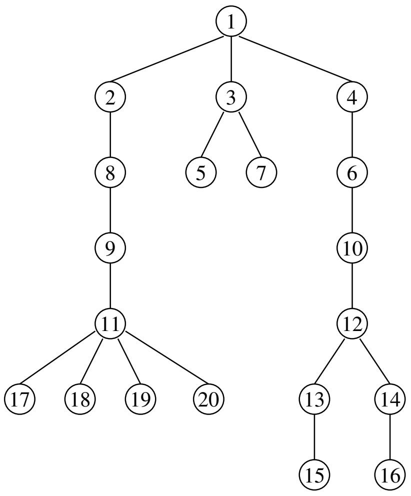
Example

standard input

20 1
3
1 1 1 3 4 3 2 8 6 9 10 12 12 13 14 11 11 11 11

standard output
8

Note



The program can be executed as follows:

Time	*Instructions*		
1	1		
2	2	3	4
3	5	6	7
4	8	10	
5	9	12	
6	11	13	14
7	15	16	17
8	18	19	20

Problem J. Tourism

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 mebibytes

King Byteasar believes that Byteotia, a land full of beautiful sights, should attract lots of tourists, who should spend lots of money, which should eventually find their way to the royal treasury. But reality does not live up to his dream. So the king instructed his councilor to look into this issue. The councilor found out that foreigners keep away from Byteotia due to its sparse road network.

Let us remark that there are n towns in Byteotia, connected by m two way roads, each road linking two different towns. The roads may lead through picturesque overflies and somewhat less picturesque tunnels. There is no guarantee that every town can be reached from every other town.

The councilor observed that the current road network does not allow for a long journey. Namely, wherever one starts the journey, it is impossible to visit more than 10 towns without passing through some town twice.

Due to limited funds in the treasury, no new roads will be constructed at the time. Instead, Byteasar has decided to build a network of tourist information points (TIPs), staffed by officers who are to advertise the short trips that are available. For each town, there should be a TIP either in this town or one of the towns directly linked by a road. Moreover, the cost of building the TIP is known for each town. Help the king by finding the cheapest way of building TIPs that will satisfy aforementioned condition.

Input

The first line of the standard input contains two integers n, m ($2 \leq n \leq 20\,000$, $0 \leq m \leq 25\,000$), separated by a single space, that specify the number of towns and roads in Byteotia respectively. The towns are numbered from 1 to n. The second line of input contains n integers $c_1, c_2, \ldots, c_n$ ($0 \leq c_i \leq 10\,000$), separated by single spaces; the number c_i specifies the cost of building a TIP in the town no. i.

Then, a description of the Byteotian road network follows. The i-th of the following m lines contains two integers a_i, b_i ($1 \leq a_i < b_i \leq n$), separated by a single space, that indicate that the towns no. a_i and b_i are linked by a road. There is at most one (direct) road between any pair of towns.

Output

Your program should print out one integer to the standard output: the total cost of building all the TIPs.

Example

standard input	standard output

6 6
3 8 5 6 2 2
1 2
2 3
1 3
3 4
4 5
4 6
	7

Note

To attain the minimum, the TIPs should be built in towns no. 1, 5, and 6. (the cost will be $3+2+2=7$).

Problem K. Freight

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

The train stations in Upper Bytown and Lower Bytown are connected with a single track rail link. It takes the train s minutes to travel between them in either direction. However, the trains departing a station have to be spaced at least one minute apart. Moreover, at all times, all the trains on the rail link have to go in the same direction.

According to the timetable at our disposal, n cargo trains destined for Lower Bytown are to pass through the Upper Bytown. They are to be loaded with goods in Lower Bytown and then return to Upper Bytown. For simplicity, we assume that loading the goods on the train takes virtually no time.

We are to determine the minimum possible time of the last train returning to Upper Bytown.

Input

The first line of the standard input contains two integers n, s ($1 \leq n \leq 1\,000\,000$, $1 \leq s \leq 10^9$), separated by a single space, that specify the number of trains and the one-way travel time respectively. The second line contains n integers $t_1, t_2, \ldots, t_n$ ($0 \leq t_1 \leq t_2 \leq \ldots \leq t_n \leq 10^9$), separated by a single space, that specify the arrival times of successive trains at the Upper Bytown station.

Output

Your program should print out a single line with a single integer to the standard output: the minimum possible time of the last train returning to Upper Bytown.

Example

standard input	standard output

`3 4`
`1 8 11`
	`20`

Note

To attain the minimum time, the trains can depart from Upper Bytown at times 1, 9, and 11, and from Lower Bytown at times 5, 15, and 16.