Diamond Collector

Bessie the cow, always a fan of shiny objects, has taken up a hobby of mining diamonds in her spare time! She has collected N diamonds ($N \leq 1000$) of varying sizes, and she wants to arrange some of them in a display case in the barn.

Since Bessie wants the diamonds in the case to be relatively similar in size, she decides that she will not include two diamonds in the case if their sizes differ by more than K (two diamonds can be displayed together in the case if their sizes differ by exactly K). Given K, please help Bessie determine the maximum number of diamonds she can display in the case.

INPUT FORMAT (file diamond.in):

The first line of the input file contains N and K ($0 \leq K \leq 10{,}000$). The next N lines each contain an integer giving the size of one of the diamonds. All sizes will be positive and will not exceed $10{,}000$.

OUTPUT FORMAT (file diamond.out):

Output a single positive integer, telling the maximum number of diamonds that Bessie can showcase.

SAMPLE INPUT:

```
5 3
1
6
4
3
1
```

SAMPLE OUTPUT:

```
4
```

Problem credits: Nick Wu

Milk Pails

Farmer John has received an order for exactly M units of milk ($1 \le M \le 1{,}000$) that he needs to fill right away. Unfortunately, his fancy milking machine has just become broken, and all he has are three milk pails of integer sizes X, Y, and M ($1 \le X < Y < M$). All three pails are initially empty. Using these three pails, he can perform any number of the following two types of operations:

- He can fill the smallest pail (of size X) completely to the top with X units of milk and pour it into the size-M pail, as long as this will not cause the size-M pail to overflow.

- He can fill the medium-sized pail (of size Y) completely to the top with Y units of milk and pour it into the size-M pail, as long as this will not cause the size-M pail to overflow.

Although FJ realizes he may not be able to completely fill the size-M pail, please help him determine the maximum amount of milk he can possibly add to this pail.

INPUT FORMAT (file pails.in):

The first, and only line of input, contains X, Y, and M, separated by spaces.

OUTPUT FORMAT (file pails.out):

Output the maximum amount of milk FJ can possibly add to the size-MM pail.

SAMPLE INPUT:

```
17 25 77
```

SAMPLE OUTPUT:

```
76
```

In this example, FJ fills the pail of size 17 three times and the pail of size 25 once, accumulating a total of 76 units of milk.

Problem credits: Brian Dean

Circular Barn

Being a fan of contemporary architecture, Farmer John has built a new barn in the shape of a perfect circle. Inside, the barn consists of a ring of n rooms, numbered clockwise from $1 \dots n$ around the perimeter of the barn ($3 \leq n \leq 1,000$). Each room has doors to its two neighboring rooms, and also a door opening to the exterior of the barn.

Farmer John wants exactly r_i cows to end up in each room i ($1 \leq r_i \leq 100$). To herd the cows into the barn in an orderly fashion, he plans to unlock the exterior door of a single room, allowing the cows to enter through that door. Each cow then walks clockwise through the rooms until she reaches a suitable destination. Farmer John wants to unlock the exterior door that will cause his cows to collectively walk a minimum total amount of distance. Please determine the minimum total distance his cows will need to walk, if he chooses the best such door to unlock. The distance walked by a single cow is the number of interior doors through which she passes.

INPUT FORMAT (file cbarn.in):

The first line of input contains n. Each of the remaining n lines contain $r_1 \dots r_n$.

OUTPUT FORMAT (file cbarn.out):

Please write out the minimum total amount of distance the cows collectively need to travel.

SAMPLE INPUT:

```
5
4
7
8
6
4
```

SAMPLE OUTPUT:

```
48
```

In this example, the best solution is to let the cows enter through the door of the room that requires 7 cows.

Problem credits: Brian Dean

Fence Painting

Several seasons of hot summers and cold winters have taken their toll on Farmer John's fence, and he decides it is time to repaint it, along with the help of his favorite cow, Bessie. Unfortunately, while Bessie is actually remarkably proficient at painting, she is not as good at understanding Farmer John's instructions.

If we regard the fence as a one-dimensional number line, Farmer John paints the interval between $x=a$ and $x=b$. For example, if $a=3$ and $b=5$, then Farmer John paints an interval of length 2. Bessie, misunderstanding Farmer John's instructions, paints the interval from $x=c$ to $x=d$, which may possibly overlap with part or all of Farmer John's interval. Please determine the total length of fence that is now covered with paint.

INPUT FORMAT (file paint.in):

The first line of the input contains the integers aa and bb, separated by a space ($a<b$).

The second line contains integers c and d, separated by a space ($c<d$).

The values of a, b, c, and d all lie in the range $0 \ldots 100$, inclusive.

OUTPUT FORMAT (file paint.out):

Please output a single line containing the total length of the fence covered with paint.

SAMPLE INPUT:

```
7 10
4 8
```

SAMPLE OUTPUT:

```
6
```

Here, 6 total units of fence are covered with paint, from $x=4$ all the way through $x=10$.

Problem credits: Brian Dean

Combination Lock

Farmer John's cows keep escaping from his farm and causing mischief. To try and prevent them from leaving, he purchases a fancy combination lock to keep his cows from opening the pasture gate.

Knowing that his cows are quite clever, Farmer John wants to make sure they cannot easily open the lock by simply trying many different combinations. The lock has three dials, each numbered 1..N (1 <= N <= 100), where 1 and N are adjacent since the dials are circular. There are two combinations that open the lock, one set by Farmer John, and also a "master" combination set by the lock maker. The lock has a small tolerance for error, however, so it will open even if the numbers on the dials are each within at most 2 positions of a valid combination. For example, if Farmer John's combination is (1,2,3) and the master combination is (4,5,6), the lock will open if its dials are set to (1,N,5) (since this is close enough to Farmer John's combination) or to (2,4,8) (since this is close enough to the master combination). Note that (1,5,6) would not open the lock, since it is not close enough to any one single combination.

Given Farmer John's combination and the master combination, please determine the number of distinct settings for the dials that will open the lock. Order matters, so the setting (1,2,3) is distinct from (3,2,1).

INPUT FORMAT (file combo.in):

Line 1: The integer N.

Line 2: Three space-separated integers, specifying Farmer John's combination.

Line 3: Three space-separated integers, specifying the master combination (possibly the same as Farmer John's combination).

OUTPUT FORMAT (file combo.out):

Line 1: The number of distinct dial settings that will open the lock.

SAMPLE INPUT (file combo.in):

50
1 2 3
5 6 7

INPUT DETAILS:

Each dial is numbered 1..50. Farmer John's combination is (1,2,3), and the master combination is (5,6,7).

SAMPLE OUTPUT: 249

Problem credits: Brian Dean

Bessie Goes Moo

Farmer John and Bessie the cow love to exchange math puzzles in their free time. The last puzzle FJ gave Bessie was quite difficult and she failed to solve it. Now she wants to get even with FJ by giving him a challenging puzzle.

Bessie gives FJ the expression (B+E+S+S+I+E)(G+O+E+S)(M+O+O), containing the seven variables B,E,S,I,G,O,M (the "O" is a variable, not a zero). For each variable, she gives FJ a list of up to 500 integer values the variable can possibly take. She asks FJ to count the number of different ways he can assign values to the variables so the entire expression evaluates to a multiple of 7.

Note that the answer to this problem can be too large to fit into a 32-bit integer, so you probably want to use 64-bit integers (e.g., "long long"s in C or C++).

INPUT FORMAT (file bgm.in):

The first line of the input contains an integer NN. The next NN lines each contain a variable and a possible value for that variable. Each variable will appear in this list at least once and at most 500 times. No possible value will be listed more than once for the same variable. All possible values will be in the range -10^5 to 10^5.

OUTPUT FORMAT (file bgm.out):

Print a single integer, giving the number of ways FJ can assign values to variables so the expression above evaluates to a multiple of 7.

SAMPLE INPUT:

```
10
B 2
E 5
S 7
I 10
O 16
M 19
B 3
G 1
I 9
M 2
```

SAMPLE OUTPUT:

```
2
```

The two possible assignments are

```
(B,E,S,I,G,O,M) = (2, 5, 7, 9,  1, 16, 19) -> 51,765
                = (2, 5, 7, 9,  1, 16, 2 ) -> 34,510
```

Problem credits: Brian Dean

Reordering the Cows

Farmer John's N cows (1 <= N <= 100), conveniently numbered 1..N, are standing in a row. Their ordering is described by an array A, where A(i) is the number of the cow in position i. Farmer John wants to rearrange them into a different ordering for a group photo, described by an array B, where B(i) is the number of the cow that should end up in position i.

For example, suppose the cows start out ordered as follows:
A = 5 1 4 2 3

and suppose Farmer John would like them instead to be ordered like this:
B = 2 5 3 1 4

To re-arrange themselves from the "A" ordering to the "B" ordering, the cows perform a number of "cyclic" shifts. Each of these cyclic shifts begins with a cow moving to her proper location in the "B" ordering, displacing another cow, who then moves to her proper location, displacing another cow, and so on, until eventually a cow ends up in the position initially occupied by the first cow on the cycle. For example, in the ordering above, if we start a cycle with cow 5, then cow 5 would move to position 2, displacing cow 1, who moves to position 4, displacing cow 2, who moves to position 1, ending the cycle. The cows keep performing cyclic shifts until every cow eventually ends up in her proper location in the "B" ordering. Observe that each cow participates in exactly one cyclic shift, unless she occupies the same position in the "A" and "B" orderings.

Please compute the number of different cyclic shifts, as well as the length of the longest cyclic shift, as the cows rearrange themselves.

INPUT FORMAT (file reorder.in):

Line 1: The integer N.
Lines 2..1+N: Line i+1 contains the integer A(i).
Lines 2+N..1+2N: Line 1+N+i contains the integer B(i).

OUTPUT FORMAT (file reorder.out):

Line 1: Two space-separated integers, the first giving the number of cyclic shifts and the second giving the number cows involved in the longest such shift. If there are no cyclic shifts, output -1 for the second number

SAMPLE INPUT:

```
5
5
1
4
2
3
2
5
3
1
4
```

SAMPLE OUTPUT:

```
2 3
```

OUTPUT DETAILS:
There are two cyclic shifts, one involving cows 5, 1, and 2, and the other involving cows 3 and 4.
Always the troublemaker, Bessie the cow has stolen Farmer John's tractor and taken off down the road!

Problem credits: Brian Dean

Speeding Ticket

The road is exactly 100 miles long, and Bessie drives the entire length of the road before ultimately being pulled over by a police officer, who gives Bessie a ticket for exceeding the speed limit, for having an expired license, and for operating a motor vehicle while being a cow. While Bessie concedes that the last two tickets are probably valid, she questions whether the police officer was correct in issuing the speeding ticket, and she wants to determine for herself if she has indeed driven faster than the speed limit for part of her journey.

The road is divided into N segments, each described by a positive integer length in miles, as well as an integer speed limit in the range $1 \ldots 100$ miles per hour. As the road is 100 miles long, the lengths of all N segments add up to 100. For example, the road might start with a segment of length 45 miles, with speed limit 70, and then it might end with a segment of length 55 miles, with speed limit 60.

Bessie's journey can also be described by a series of segments, M of them. During each segment, she travels for a certain positive integer number of miles, at a certain integer speed. For example, she might begin by traveling 50 miles at a speed of 65, then another 50 miles at a speed of 55. The lengths of all M segments add to 100 total miles. Farmer John's tractor can drive 100 miles per hour at its fastest.

Given the information above, please determine the maximum amount over the speed limit that Bessie travels during any part of her journey.

INPUT FORMAT (file speeding.in):

The first line of the input contains N and M, separated by a space.
The next N lines each contain two integers describing a road segment, giving its length and speed limit.
The next M lines each contain two integers describing a segment in Bessie's journey, giving the length and also the speed at which Bessie was driving.

OUTPUT FORMAT (file speeding.out):

Please output a single line containing the maximum amount over the speed limit Bessie drove during any part of her journey. If she never exceeds the speed limit, please output 0.

SAMPLE INPUT:

```
3 3
40 75
50 35
10 45
40 76
20 30
40 40
```

SAMPLE OUTPUT:

```
5
```

In this example, the road contains three segments (40 miles at 75 miles per hour, followed by 50 miles at 35 miles per hour, then 10 miles at 45 miles per hour). Bessie drives for three segments (40 miles at 76 miles per hour, 20 miles at 30 miles per hour, and 40 miles at 40 miles per hour). During her first segment, she is slightly over the speed limit, but her last segment is the worst infraction, during part of which she is 5 miles per hour over the speed limit. The correct answer is therefore 5.

Problem credits: Austin Bannister and Brian Dean

Awkward Digits

Bessie the cow is just learning how to convert numbers between different bases, but she keeps making errors since she cannot easily hold a pen between her two front hooves.

Whenever Bessie converts a number to a new base and writes down the result, she always writes one of the digits wrong. For example, if she converts the number 14 into binary (i.e., base 2), the correct result should be "1110", but she might instead write down "0110" or "1111". Bessie never accidentally adds or deletes digits, so she might write down a number with a leading digit of "0" if this is the digit she gets wrong.

Given Bessie's output when converting a number N into base 2 and base 3, please determine the correct original value of N (in base 10). You can assume N is at most 1 billion, and that there is a unique solution for N.

Please feel welcome to consult any on-line reference you wish regarding base-2 and base-3 numbers, if these concepts are new to you.

INPUT FORMAT (file digits.in):

Line 1: The base-2 representation of N, with one digit written incorrectly.

Line 2: The base-3 representation of N, with one digit written incorrectly.

OUTPUT FORMAT (file digits.out):

Line 1: The correct value of N.

SAMPLE INPUT:

```
1010
212
```

INPUT DETAILS:

When Bessie incorrectly converts N into base 2, she writes down "1010". When she incorrectly converts N into base 3, she writes down "212".

SAMPLE OUTPUT:

```
14
```

OUTPUT DETAILS:

The correct value of N is 14 ("1110" in base 2, "112" in base 3).

Problem credits: Brian Dean

Wormholes

Farmer John's hobby of conducting high-energy physics experiments on weekends has backfired, causing N wormholes (2 <= N <= 12, N even) to materialize on his farm, each located at a distinct point on the 2D map of his farm.

According to his calculations, Farmer John knows that his wormholes will form N/2 connected pairs. For example, if wormholes A and B are connected as a pair, then any object entering wormhole A will exit wormhole B moving in the same direction, and any object entering wormhole B will similarly exit from wormhole A moving in the same direction. This can have rather unpleasant consequences. For example, suppose there are two paired wormholes A at (0,0) and B at (1,0), and that Bessie the cow starts from position (1/2,0) moving in the +x direction. Bessie will enter wormhole B, exit from A, then enter B again, and so on, getting trapped in an infinite cycle!

Farmer John knows the exact location of each wormhole on his farm. He knows that Bessie the cow always walks in the +x direction, although he does not remember where Bessie is currently located. Please help Farmer John count the number of distinct pairings of the wormholes such that Bessie could possibly get trapped in an infinite cycle if she starts from an unlucky position.

INPUT FORMAT (file wormhole.in):

Line 1: The number of wormholes, N.

Lines 2..1+N: Each line contains two space-separated integers describing the (x,y) coordinates of a single wormhole. Each coordinate is in the range 0..1,000,000,000.

OUTPUT FORMAT (file wormhole.out):

Line 1: The number of distinct pairings of wormholes such that Bessie could conceivably get stuck in a cycle walking from some starting point in the +x direction.

SAMPLE INPUT:

```
4
0 0
1 0
1 1
0 1
```

INPUT DETAILS:

There are 4 wormholes, forming the corners of a square.

SAMPLE OUTPUT:

```
2
```

OUTPUT DETAILS:

If we number the wormholes 1..4, then by pairing 1 with 2 and 3 with 4, Bessie can get stuck if she starts anywhere between (0,0) and (1,0) or between (0,1) and (1,1). Similarly, with the same starting points, Bessie can get stuck in a cycle if the pairings are 1-3 and 2-4. Only the pairings 1-4 and 2-3 allow Bessie to walk in the +x direction from any point in the 2D plane with no danger of cycling.

Problem credits: Brian Dean