HPC³ 2024
Problem E, English
Shredded Secrets
Maximum Points: 40

You have recently been fired from your job in the International Pirate Treasure Security Council (IPTSC) for eating all the break room snacks and embezzling money. This means that they will wipe your memory, shred all of your documents, and throw them in the trash. This is terrible news because you happen to know the secret locations of all treasure under the Council's watch. However, you have an undiscovered accomplice in another department who can retrieve the shredded paper from the trash. You plan on communicating the locations of the treasure to your accomplice who will then go and find them, making you both fabulously wealthy.

The treasure can be represented as an array of points P of length n on a grid of size $N \times N$ $(1 \leq N \leq 10^5)$. You will write an array of non-negative integers K of arbitrary length l that will be modified in some way by the shredder. Your accomplice will then receive the modified integers and must determine the original P.

Notes

- This problem requires you to write two programs: one to convert the array of points to an array of integers, and one to convert the modified array of integers back to the array of points. HPC³ offers you two ways to do this: Submitting two distinct files or submitting one file with two distinct functions.
  If you choose the second method, you must isolate all variables.
- This problem has special scoring. Submissions will be graded based on test case accuracy, runtime, memory usage, and additionally the value of l divided by n. The most points are awarded for $\frac{l}{n} \leq 3z$, and 0 for $\frac{l}{n} > 10z$.
- This problem has randomness involved. Because of this, it may manifest nondeterministic results. However, the correct solution will always be the most optimal and so will always score the maximum number of points.

Subproblem 1

The shredder is a standard IPTSC-issue shredder. All that is does is cut arrays into their individual elements and shuffle them. Formally, for an array K, the shredder will give you $\widehat{K}$, the elements of K rearranged randomly.

The value of z in the grading scale is 1 for this subproblem.

The maximum integer size you can write to K is 10^6. Given P, determine a K, then given $\widehat{K}$, determine the original value of P.

Input A format

The first line of each input contains 1 integer n.
The second line of each input contains n integer pairs: The content of array P.

```
n
P[0][0]  P[0][1]  P[1][0]  P[1][1]  …  P[n-1][0]  P[n-1][1]
```

Output A format

The first line of each input contains 1 integer l.
The second line of each input contains l integers: The content of array K.

```
l
K[0]  K[1]  K[2]  …  K[l-1]
```

Input B format

The first line of each input contains 1 integer l.
The second line of each input contains l integers: The content of array $\widehat{K}$.

```
l
K̂[0]  K̂[1]  K̂[2]  …  K̂[l-1]
```

Output B format

The first line of each input contains 1 integer n.
The second line of each input contains n integer pairs: The content of array P.

```
n
P[0][0]  P[0][1]  P[1][0]  P[1][1]  …  P[n-1][0]  P[n-1][1]
```

Example Test Cases

Input 1A

```
2
1 2 4 2
```

Output 1A

```
6
1 2 4 2 1 1
```

Input 1B

```
6
1 4 2 1 2 1
```

Output 1B

```
2
4 2 1 2
```

[1, 2, 4, 2, 1, 1] becomes [1, 4, 2, 1, 2, 1] in the randomisation process. Note that the program may respond with points in a different order than the input.

Subproblem 2

The shredder is not a shredder, but a Data Obfuscation Device! It works like this: It has a non-negative integer d $(1 \leq d \leq 100)$ and a binary array A of length a $(1 \leq a \leq l)$. For every element in a given K, K_i $(0 \leq i < l)$, if the $i \bmod a$-th element of A is 1, then $\widehat{K_i}$ will be K_i with a random integer value between $(-d, d)$ added. Otherwise, $\widehat{K_i}$ will be K_i.

The value of z in the grading scale is $\frac{2}{3}$ for this subproblem.

The maximum integer size you can write to K is 10^5. Given P, A, and d, determine a K, then given $\widehat{K}$, determine the original value of P.

Input A format

The first line of each input contains 3 integers n, d, and a.
The second line of each input contains n integer pairs: The content of array P.
The third line of each input contains a binary values: The content of array A.

```
n  d  a
P[0][0]  P[0][1]  P[1][0]  P[1][1]  …  P[n-1][0]  P[n-1][1]
A[0]  A[1]  A[2]  …  A[a-1]
```

Output A format

The first line of each input contains 1 integer l.
The second line of each input contains l integers: The content of array K.

```
l
K[0]  K[1]  K[2]  …  K[l-1]
```

Input B format

The first line of each input contains 1 integer l.
The second line of each input contains l integers: The content of array $\widehat{K}$.

```
l
K̂[0]  K̂[1]  K̂[2]  …  K̂[l-1]
```

Output B format

The first line of each input contains 1 integer n.
The second line of each input contains n integer pairs: The content of array P.

```
n
P[0][0]  P[0][1]  P[1][0]  P[1][1]  …  P[n-1][0]  P[n-1][1]
```

Example Test Cases

Input 1A

```
4  5  2
1  3  1  4  3  3  4  3
0  1
```

Output 1A

```
8
7  14  18  19  19  23  24  24
```

Input 1B

```
8
7  18  18  24  19  23  24  29
```

Output 1B

```
4
1  3  1  4  3  3  4  3
```

d is 5 and a is [0, 1], so every 2nd input can be modified by {-5, 5}. In this way, [7, 14, 18, 19, 19, 23, 24, 24] becomes [7, 18, 18, 24, 19, 23, 24, 29].
The changes are [0, 4, 0, 5, 0, 0, 0, 5].

Subproblem 3

IPTSC have tight security! The shredder is a combination of the machines from the previous 2 subproblems. It will first randomise the array, then apply the Data Obfuscation Device process to it.

The value of z in the grading scale is $\frac{5}{3}$ for this subproblem.

The maximum integer size you can write to K is 10^8. Given P, A, and d, determine a K, then given $\widehat{K}$, determine the original value of P.

Input A format

The first line of each input contains 2 integers n, d, and a.
The second line of each input contains n integer pairs: The content of array P.
The third line of each input contains a binary values: The content of array A.

```
n  d  a
P[0][0]  P[0][1]  P[1][0]  P[1][1]  …  P[n-1][0]  P[n-1][1]
A[0]  A[1]  A[2]  …  A[a-1]
```

Output A format

The first line of each input contains 1 integer l.
The second line of each input contains l integers: The content of array K.

```
l
K[0]  K[1]  K[2]  …  K[l-1]
```

Input B format

The first line of each input contains 1 integer l.
The second line of each input contains l integers: The content of array $\widehat{K}$.

```
l
K̂[0]  K̂[1]  K̂[2]  …  K̂[l-1]
```

Output B format

The first line of each input contains 1 integer n.
The second line of each input contains n integer pairs: The content of array P.

```
n
P[0][0]  P[0][1]  P[1][0]  P[1][1]  …  P[n-1][0]  P[n-1][1]
```

Example Test Cases

Input 1A

```
5  20  3
1  1  2  2  3  3  4  4  5  5
0  1  1
```

Output 1A

```
25
4  7  9  2  6 22  25  27  29  38  41  44  36  43  53  56  59  51  55
67  70  73  68  72  82  85  89  83  87  97  99  96  98  94
```

Input 1B

```
25
41  93  6  98  68  25  23  4  50  93  29  50  29  54  92  36  96  14
73  93  51  6  56  68  71  43  51  87  6  94  83  25
```

Output 1B

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

[4, 7, 9, 2, 6 22, 25, 27, 29, 38, 41, 44, 36, 43, 53, 56, 59, 51, 55, 67, 70, 73, 68, 72, 82, 85, 89, 83, 87, 97, 99, 96, 98, 94] becomes

[41, 93, 6, 98, 68, 25, 23, 4, 50, 93, 29, 50, 29, 54, 92, 36, 96, 14, 73, 93, 51, 6, 56, 68, 71, 43, 51, 87, 6, 94, 83, 25].

The changes before the array is randomised are:

[0, 4, -2, 0, -6, 9, 0, 7, -10, 0, 5, 3, 0, -1, 1, 0, 3, -7, 0, 8, -3, 0, 2, -5, 0].