Hong Kong Olympiad in Informatics 2021/22 Junior Group

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
J221	Bus Route Category	1.000 s	256 MB	7 + 12 + 19 + 21 + 26 + 15
J222	Spicy Ramen	1.000 s	256 MB	14 + 17 + 25 + 17 + 27
J223	Ice Cream	1.000 s	256 MB	3 + 17 + 17 + 7 + 26 + 30
J224	Digit Implant Strategy	1.000 s	256 MB	11 + 6 + 12 + 11 + 24 + 36

Notice:
Unless otherwise specified, inputs and outputs shall follow the format below:

- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- Outputs will be automatically fixed as follows: Trailing spaces in each line will be removed and an end-of-line character will be added to the end of the output if not present. All other format errors will not be fixed.

C++ programmers should be aware that using C++ streams (`cin` / `cout`) may lead to I/O bottlenecks and substantially lower performance.

For some problems 64-bit integers may be required. In Pascal it is `int64`. In C/C++ it is `long long` and its token for `scanf`/`printf` is `%lld`.

All tasks are divided into subtasks. You need to pass all test cases in a subtask to get points.

J221 - BUS ROUTE CATEGORY

Time Limit: 1.000 s / Memory Limit: 256 MB

Bob loves patterns! One day, he discovered that all bus route numbers in Heung Shing follow some format and patterns which sparked his interest in finding the patterns. After researching online, he learned that all bus route numbers in Heung Shing must consist of 0 or 1 uppercase letter prefix, a number between 1 and 999 (inclusive) with no leading zeros, followed by 0 or 1 uppercase letter suffix. Below are the lists of bus route categories represented by the digits and letters:

Part A	
Letter Prefix	Category A
No Letter Prefix	`Normal`
`A`	`Airport`
`B`	`Border`
`N`	`Overnight`
Other letters	Invalid

Part B	
Hundreds Digit	Category B
No Hundreds Digit	`Normal`
`1`	`Cross River`
`2`	`Air-conditioned`
`3`	`Holiday`
Other numbers	Invalid

Part C	
Tens Digit	Category C
`0`, `1`, `2` or No Tens Digit	`Downtown`
`3`, `4`	`West District`
`7`	`North District`
`9`	`East District`
Other numbers	Invalid

Part D	
Letter Suffix	Category D
`A`, `B`, `C` or No Letter Suffix	`Normal`
`P`	`Peak Hour`
`S`	`Special`
`X`	`Express`
Other letters	Invalid

The unit digit can be any number. If any of the above Categories is Invalid, the bus route number is considered invalid. The following pairs of Categories are also considered invalid:

Incompatible Pair		
1.	`Overnight`	`Peak Hour`
2.	`Holiday`	`Peak Hour`

Bob wants to randomly come up with a bus route number and know Categories that are represented by this route number. As a friend of Bob, please help Bob summarize the Bus Route Category!

INPUT

The first and only line is the bus route number Bob has come up with. The input is guaranteed to follow the following format: must consist of 0 or 1 uppercase letter prefix, a number between 1 and 999 (inclusive) with no leading zeros, followed by 0 or 1 uppercase letter suffix.

OUTPUT

If the bus route number is invalid, output `Invalid`.

Otherwise, output a single line of Categories the bus route number represent. The Categories must be output in the order of Category C, Category B, Category D, Category A. `Normal` should be output exactly once if and only if all Category A, Category B and Category D are `Normal`, otherwise, do not output `Normal`. Each Category should be separated by a space.

SAMPLE TESTS

	Input	Output
1	`1`	`Downtown Normal`

There is no Letter Prefix, so Category A is `Normal`.
There is no Hundreds Digit, so Category B is `Normal`.
There is no Tens Digit, so the Category C is `Downtown`.
There is no Letter Suffix, so Category D is `Normal`.
Note that Categories A, Categories B and Categories D are `Normal`, so output `Normal` once.

2	`N208A`	`Downtown Air-conditioned Overnight`

The Letter Prefix is `N`, so Category A is `Overnight`.
The Hundreds Digit is `2`, so Category B is `Air-conditioned`.
The Tens Digit is `0`, so the Category C is `Downtown`.
The Letter Suffix is `A`, so Category D is `Normal`.
Note that only Category D is `Normal`, so `Normal` is omitted.

3	`A199X`	`East District Cross River Express Airport`

The Letter Prefix is `A`, so Category A is `Airport`.
The Hundreds Digit is `1`, so Category B is `Cross River`.
The Tens Digit is `9`, so the Category C is `East District`.
The Letter Suffix is `X`, so Category D is `Express`.

4	`333P`	`Invalid`

There is no Letter Prefix, so Category A is `Normal`.
The Hundreds Digit is `3`, so Category B is `Holiday`.
The Tens Digit is `3`, so the Category C is `West District`.
The Letter Suffix is `P`, so Category D is `Peak Hour`.
Since `Holiday` and `Peak Hour` is an Incompatible Pair, so the bus route number is invalid.

5	`Z987Z`	`Invalid`

This bus route number is invalid because of the following reasons:

- The Letter Prefix `Z` is not in Part A.
- The Hundreds Digit `9` is not in Part B.
- The Tens Digits `8` is not in Part C.
- The Letter Suffix `Z` is not in Part D.

SUBTASKS

	Points	Constraints
1	7	The bus route number consists of a number between 1 and 99 (inclusive) only
2	12	The bus route number consists of a number only
3	19	The bus route number does not consist of a Letter Prefix
4	21	Category A, Category B and Category D must not be `Normal`
5	26	The bus route number is guaranteed to be valid
6	15	No additional constraints

J222 - SPICY RAMEN

Time Limit: 1.000 s / Memory Limit: 256 MB

Spicy instant ramen manufacturer RaRaRamen has produced ramen of different spiciness. For example, *Micro-spicy Ramen* with spiciness of 200 and *Ultra-spicy Ramen* with spiciness of 10000000. Here, spiciness is represented by a non-negative integer. The spicier the ramen, the larger the value of spiciness.

Now, RaRaRamen is developing a new product. The spiciness of this new product should lie in the range from 0 to R (inclusive). For the new product, a market research is conducted to gather customers' preferences.

In the research, $N + M$ customers have responded. For the first N customers, the i-th customer prefers ramen with spiciness $\geq A_i$. For the remaining M customers, the i-th customer prefers ramen with spiciness $\leq B_i$.

The new product should fulfill the preferences of at least K customers. Could you find how many different spicinesses can fulfill this requirement?

INPUT

The first line contains 4 integers, N, M, K, R.
The second line contains N integers, $A_1, A_2, \ldots, A_N$.
The third line contains M integers, $B_1, B_2, \ldots, B_M$.

OUTPUT

Output a single integer denoting the number of spicinesses that are able to fulfill at least K customers' preferences.

SAMPLE TESTS

	Input	Output
1	2 2 3 70 30 50 40 65	27

Spiciness in the ranges of 30-40 (inclusive) and 50-65 (inclusive) could satisfy at least 3 customers' preferences. The ranges include 27 different values.

	Input	Output
2	4 2 6 50 25 20 0 10 50 40	16

Spiciness in the range of 25-40 (inclusive) could satisfy 6 customers' preferences. The range include 16 different values.

	Input	Output
3	3 3 4 100 41 71 89 0 23 53	0

SUBTASKS

For all cases:
$1 \leq N, M \leq 2 \times 10^5$
$1 < K \leq N + M$
$1 < R \leq 10^9$
$0 \leq A_i, B_i \leq R$

	Points	Constraints
1	14	$N = 2, M = 2, 3 \leq K \leq 4$ $1 \leq R \leq 3000$
2	17	$1 \leq N, M \leq 1000$ $1 \leq R \leq 3000$
3	25	$1 \leq R \leq 10^6$
4	17	$M = 2, K = N + M - 1$
5	27	No additional constraints

J223 - ICE CREAM

Time Limit: 1.000 s / Memory Limit: 256 MB

David has just finished his assignment. As he walked home, two ice cream shops came into sight. David then decided to buy a scoop of ice cream from each shop as a reward. He has some coins in hand for ice cream, and would like to minimise the number of shops he gets change from upon buying the two scoops.

Specifically, David has S_1 1-dollar coins, S_2 2-dollar coins, and S_5 5-dollar coins for the scoops. He plans to get one scoop from one of the shops and finish it before getting another one from the other shop. Ice cream is priced at $\$X_A$ per scoop at shop A and $\$X_B$ at shop B. If a shop charges $\$x$ per scoop, David needs to pay at least $\$x$ to get a scoop. If he pays $\$y$, and $y > x$, he will get a change of $\$(y - x)$.

The way that the shops give changes follows a fixed algorithm. If a shop needs to give $\$z$ worth of change, and:

- if $z \geq 5$, give a 5-dollar coin, and proceed to give $\$(z - 5)$ change;
- if $2 \leq z < 5$, give a 2-dollar coin, and proceed to give $\$(z - 2)$ change;
- if $1 \leq z < 2$, give a 1-dollar coin, and proceed to give $\$(z - 1)$ change;
- if $z = 0$, terminate

For example, if $z = 13$, 4 coins will be used for change: $z = 5 + 5 + 2 + 1$. From the algorithm above, it should be obvious that for any $z \geq 0$, the combination of coins a shop gives for change is always deterministic.

Since David has just gone over his tough assignment, he is not in the capacity to come up with an approach that serves him best. You are empathetic of his situation and would like to lend a hand to him. You would like to tell him whether he can buy both scoops with the coins he has. If he can, then let him know:

1. The minimum number of shops he needs to get change from;
2. From which shop he should first buy a scoop to make that happen;
3. The combination of coins to be used in shops A and B in this plan

If there are more than one optimal solutions, you can output any.

INPUT

The first line of input consists of 3 numbers, S_1, S_2, and S_5, which are the number of 1-dollar, 2-dollar, and 5-dollar coins that David has.

The second line of input consists of 2 numbers, X_A and X_B, the price per scoop of ice cream in shop A and shop B respectively.

OUTPUT

If David cannot buy both scoops with the coins he has, output `Impossible` on the only line.

Otherwise, if it is possible for him to get both scoops, output `Possible` on the first line. Then, output the minimum number of shops he needs to get change from on the second line.

On the next 3 lines, output a valid construction to attain this minimum:

- On the first line, output `A` if David should go to shop A first, or `B` if shop B first.
- On the second line, output 3 numbers, indicating the number of 1-dollar, 2-dollar and 5-dollar coins to be used in shop A.
- On the third line, output 3 numbers, indicating the number of 1-dollar, 2-dollar and 5-dollar coins to be used in shop B.

SAMPLE TESTS

	Input	Output

1	`0 5 2`
	`10 6`
		`Possible`
		`0`
		`B`
		`0 0 2`
		`0 3 0`

David has 5 2-dollar coins and 2 5-dollar coins. A scoop is priced at 10 dollars in shop A and 6 dollars in shop B.
One way to pay for both scoops without involving any change is to get a scoop from shop B first with 3 2-dollar coins, and then from shop A with 2 5-dollar coins.

2	`0 5 2`
	`10 7`
		`Possible`
		`1`
		`B`
		`0 3 1`
		`0 1 1`

3	`0 5 2` `10 11`	`Impossible`

SUBTASKS

For all cases: $0 \le S_1, S_2, S_5, X_A, X_B \le 10^8$

	Points	Constraints
1	3	$S_2 = S_5 = 0$
2	17	$S_2 = 0$
3	17	$S_5 = 0$
4	7	$S_1 = S_2 = S_5 = 10^8$
5	26	$0 \le S_1, S_2, S_5, X_A, X_B \le 500$
6	30	No additional constraints

J224 - DIGIT IMPLANT STRATEGY

Time Limit: 1.000 s / Memory Limit: 256 MB

In Chemistry class, students are often required to do experiments that involve complicated procedures. During the experiments, students need to collect some experimental data by making careful measurements.

Now, they are required to measure the reactivity of a mysterious substance, *Hkoirite*. The reactivity is represented by a positive integer with no leading zeros. This number may be very large depending on the purity of the substance.

Prof. Sierra, being a nice teacher, demonstrated the experiment in front of the whole class. In his demonstration, the reactivity measured is the integer S. After the demonstration, Tango, a student, measured the reactivity as integer T in his own experiment. He thinks that Prof. Sierra probably has a more accurate measurement than him. So, in order to get a higher score, he decides to alter T by inserting a digit x, producing the integer T'.

The digit x can be inserted before or after T, or in between any two adjacent digits of T. For example, if T is 146 and x is 3, then T' can be 3146, 1346, 1436 or 1463.

He would like to know, which number T' he can produce such that the absolute difference of S and T', $|S - T'|$ is minimized. Could you figure it out for him? Please note that even if T is close to S, he would still insert the digit x no matter what.

INPUT

The first line contains a postive integer S.
The second line contains a postive integer T.
The third line contains a digit x.

OUTPUT

Output an integer T' such that the absolute difference $|S - T'|$ is minimized. If there are multiple T' that satisfy the requirement, output any one of them.

SAMPLE TESTS

	Input	Output
1	87663 521 8	8521

Among 8521, 5821, 5281 and 5218, 8521 has the minimum absolute difference with 87663.

	Input	Output
2	37373737 73737373737 3	373737373737

Even if not inserting the digit would make the number closer to S, the digit still has to be inserted.

	Input	Output
3	99000 9999 8	98999

SUBTASKS

For all cases:
$1 \leq$ Length of S, Length of $T \leq 10^6$
$1 \leq x \leq 9$

	Points	Constraints
1	11	$1 \leq$ Length of S, Length of $T \leq 8$
2	6	(Length of S) $<$ (Length of T) $+ 1$ x and the digits of S and T is either 3 or 7
3	12	(Length of S) $>$ (Length of T) $+ 1$ x and the digits of S and T can only be 3, 5, or 7
4	11	(Length of S) $\neq$ (Length of T) $+ 1$
5	24	The first digits of S and T are different.
6	36	No additional constraints