Problem A. Load Distribution

Time Limit 1000 ms
Mem Limit 262144 kB
OS Windows

And so the competition has begun!

The team consisting of Aidar, Begimai, and Viktor rushed to read the conditions of all n problems to find the easiest ones.

The team agreed to read the problems in the following manner:

- Aidar reads starting from problem number 1 and moves towards problem n.
- Viktor reads starting from problem number n and moves towards problem 1.
- Begimai starts reading from a fixed problem s between 1 and n, after which she starts moving either towards problem 1 or towards problem n (the direction is also fixed).

It is known that

- each participant reads each problem in the same amount of time.
- a participant will stop if they reach a problem that has already been read by another team member.
- if two participants are supposed to read the same problem, they will read it **together** for reliability.

Determine how many problems each team member will read.

Input

The first line contains an integer n $(3 \leq n \leq 15)$ — the number of problems in the competition.

The second line contains an integer s $(1 < s < n)$ — the problem from which Begimai starts reading.

The third line specifies Begimai's reading direction:

- L, if Begimai reads the problems towards problem 1.

- R, if Begimai reads the problems towards problem n.

Output

In a single line, output three integers a, b, and c $(1 \leq a, b, c; a+b+c=n$ or $a+b+c=n+1)$:

- a — how many problems Aidar will read;
- b — how many problems Begimai will read;
- c — how many problems Viktor will read.

Examples

Input	Output
12 9 L	5 5 3

Input	Output
8 7 R	6 1 1

Note

First test example

There are a total of 12 problems in the competition.

Begimai started reading from problem 9 and began moving towards problem 1.

- Aidar managed to read problems 1, 2, 3, 4, 5 — a total of 5 problems.
- Begimai managed to read problems 9, 8, 7, 6, 5 — a total of 5 problems.
- Viktor managed to read problems 12, 11, 10 — a total of 3 problems.

Second test example

There are a total of 8 problems in the competition.

Begimai started reading from problem 7 and began moving towards problem 8.

- Aidar managed to read problems 1, 2, 3, 4, 5, 6 — a total of 6 problems.

- Begimai managed to read problem 7 — a total of 1 problem.
- Viktor managed to read problem 8 — a total of 1 problem.

Problem B. What to solve next?

Time Limit 1000 ms
Mem Limit 262144 kB
OS Windows

After the guys submitted their first task, the question arose — which task should be solved next?

Begimai, who had experience in such competitions, suggested looking at the page with the "Results".

This page shows how many tasks each team has solved — and how many attempts the team made on each task:

- If a team solved a task, it is indicated in the table with a +, followed by the number of incorrect attempts before the first successful one.
- If a team did not solve a task, it is indicated in the table with a -, followed by the number of incorrect attempts already made.
- If a team did not make any attempts on a task, nothing is indicated in the table for that task.

Begimai explained that additional information about the difficulty of tasks can be extracted from this statistics:

- If a task is hardly solved, except by the teams at the very top of the table — then, most likely, it is more difficult for beginners compared to other tasks.
- If different teams make many incorrect attempts on the same task — then, most likely, there are some "underwater stones" and tricky cases that are not obvious at first glance.

The guys decided to **collect a small statistics for each task**:

- the number of teams that solved the task;
- the number of teams that made at least one attempt on the task;
- the total number of incorrect attempts made by teams on the task.

Input

The first line contains two integers n and p $(1 \le n, p \le 100)$ — the number of teams and the number of tasks in the results table.

The i-th of the following n lines contains information about the results of the i-th team — p pairs of values $R_{i,j}$, separated by spaces:

- If the i-th team did not make attempts on task j, then $R_{i,j}$ will consist of `-` and the number 0.
- If the i-th team made k incorrect attempts on task j, but **did not solve** it, then $R_{i,j}$ will consist of `-` and the number k $(1 \leq k \leq 100)$.
- If the i-th team made k incorrect attempts on task j, but then **solved** it, then $R_{i,j}$ will consist of `+` and the number k $(0 \leq k \leq 100)$.

Output

In the j-th of the p lines, output three integers S_j, T_j, F_j separated by spaces:

- S_j $(0 \leq S_j \leq n)$ — the number of teams that solved task j;
- T_j $(S_j \leq T_j \leq n)$ — the number of teams that made at least one attempt on task j;
- F_j $(0 \leq F_j \leq 10^9)$ — the total number of **incorrect** attempts on task j.

Examples

Input	Output

`2 3`
`+ 0 + 3 - 5`
`+ 1 - 0 - 12`
	`2 2 1`
	`1 1 3`
	`0 2 17`

Note

First test example

In the results table, there are 2 teams and 3 tasks:

- The first team corresponds to the first line:
  - The first pair is + and 0 — the first task was solved by the team without incorrect attempts.
  - The second pair is + and 3 — the second task was solved by the team after 3 incorrect attempts.
  - The third pair is - and 5 — the team has already made 5 incorrect attempts, but has not yet solved the third task.
- The second team corresponds to the second line:

  - The first pair is + and 1 — the first task was solved by the team after one incorrect attempt.
  - The second pair is - and 0 — the team did not make attempts on the second task.
  - The third pair is - and 12 — the team has already made 12 incorrect attempts, but has not yet solved the third task.

In total, it turns out that

- The first task was solved by 2 teams, 2 teams also made attempts on it, and the total number of incorrect attempts is 1.
- The second task was solved by 1 team, 1 team also made attempts on it, and the total number of incorrect attempts is 3.
- The third task was solved by 0 teams, 2 teams made attempts on it, and the total number of incorrect attempts is 17.

Problem C. Integer Overflow

Time Limit 1000 ms
Mem Limit 262144 kB
OS Windows

Victor completed the solution for the problem and, without waiting for feedback from the team, submitted it to the system.

Upon refreshing the submission page, Victor saw the unpleasant verdict `Wrong Answer`.

It was unpleasant for two reasons:

- First, it meant an extra 20 minutes of penalty time;
- Second, Victor had no idea what he had done wrong — and that frustrated him the most!

His teammates Aidar and Begimai noticed this.

They observed that Victor's solution used two integer variables A and B, after which the result of their multiplication $C = A \cdot B$ was computed.

Moreover, all three variables were declared as 32-bit.

Both Aidar and Begimai suspected that the program was experiencing **integer overflow** — a situation where the value of an integer does not fit into the specified data type.

- Aidar suggested that perhaps the values of A and/or B themselves do not fit into 32 bits.
- Begimai believed that A and B were fine, but their product C did not fit.
- In response to both suggestions, Victor remarked that he should have written the solution in Python, where integers have unlimited size.

The team could always rewrite it in Python later — right now they wanted to understand what data types should be used for A, B, and C.

Important Information

- The problem assumes only three integer types:
  - 32-bit type: $[-2^{31}; 2^{31} - 1] = [-2147483648; 2147483647]$;
  - 64-bit type: $[-2^{63}; 2^{63} - 1] =$ $[-9223372036854775808; 9223372036854775807]$;
  - 128-bit type: $[-2^{127}; 2^{127} - 1]$.

- If an X-bit integer is multiplied by a Y-bit integer, the result of their product will be computed as a Z-bit number, where $Z = \max(X, Y)$.

Input

The first line contains an integer A $(1 \le A \le 10^{18})$ — the value of variable A.

The second line contains an integer B $(1 \le B \le 10^{18})$ — the value of variable B.

Output

In the first line, output the integer T_A — the required number of bits for variable A.

In the second line, output the integer T_B — the required number of bits for variable B.

In the third line, output the integer T_C — the required number of bits for variable C.

All three numbers T_A, T_B, T_C can only take values 32, 64, or 128:

- **At least one** of the equalities $T_C = T_A$ or $T_C = T_B$ must hold.
- If multiple correct answers are possible, output the answer with the **minimum sum** $T_A + T_B + T_C$.

Examples

Input	Output
20000 100000	32 32 32

Input	Output
1000000 300000	64 32 64

Input	Output
3000000 3000000000000	32 64 64

Input	Output
1000000000000000000 1000000000000000000	128 64 128

Note

A Minute of Useful Information

- The 32-bit integer type in C++ and Java is denoted as `int`.
- The 64-bit integer type in C++ is denoted as `long long`.
- The 64-bit integer type in Java is denoted as `long`.
- 128-bit integer types in C++ and Java depend on the compiler, but their use comes with additional difficulties.
- The `int` type in Python has an unlimited number of bits.

First test example

- $A = 2 \cdot 10^4$;
- $B = 10^5$;
- $C = A \cdot B = 2 \cdot 10^9$.

All three numbers fit into a 32-bit data type.

Second test example

- $A = 10^6$;
- $B = 3 \cdot 10^5$;
- $C = A \cdot B = 3 \cdot 10^{11}$.

A and B fit into a 32-bit data type, but C fits only into a 64-bit one.

It is necessary to make either variable A or B 64-bit.

Third test example

- $A = 3 \cdot 10^6$;
- $B = 3 \cdot 10^{12}$;
- $C = A \cdot B = 9 \cdot 10^{18}$.

A fits into a 32-bit data type, but B and C fit only into a 64-bit one.

Since B is already 64-bit, A can remain 32-bit.

Fourth test example

- $A = 10^{18}$;
- $B = 10^{18}$;
- $C = A \cdot B = 10^{36}$.

A and B fit into a 64-bit data type, but C fits only into a 128-bit one.

It is necessary to make either variable A or B 128-bit.

Problem D. Final Rankings

Time Limit 1000 ms
Mem Limit 262144 kB
OS Windows

The championship director Artem is constantly improving the testing system, making it more stable, efficient, and user-friendly.

This time, Artem has upgraded the team sorting system in the results table. To test his modification, Artem decided to run it on the results of one of the previous championships.

However, a problem arose — Artem only has the log of submissions, but he does not have the final ranking of the teams.

Help Artem — based on the available information about the teams' submissions, calculate the final results and their rankings.

Rules for Sorting Teams and Determining Rankings

- First, teams are sorted in descending order of the number of **correctly** solved problems.
- In case of a tie in the number of solved problems, teams are sorted in ascending order of **total** penalty time:
  - Suppose a team correctly solved a problem after M minutes from the start of the competition, having made exactly F penalty attempts on that problem.
  - In this case, the team will receive $(M + 20 \cdot F)$ penalty time for solving this problem.
  - Subsequent attempts on an already solved problem do not change the penalty time.
  - If a problem is not solved, no penalty is incurred regardless of the number of attempts.
- In case of a tie in both the number of solved problems and total penalty time, teams share the **same** rank.
  - For convenience, such teams are sorted in the table in ascending order of their identifiers.

Input

The first line contains two integers n and p $(1 \le n \le 10^3; 1 \le p \le 20)$ — the number of teams and the number of problems in the competition.

The second line contains an integer q $(1 \le q \le 10^5)$ — the number of submission records in the log.

In the j-th of the following q lines, information about the j-th submission in the system is provided — four integers M_j, ID_j, T_j, and V_j are given separated by spaces:

- M_j $(1 \le M_j \le 300)$ — the minute at which the submission was made;
- ID_j $(1 \le ID_j \le n)$ — the identifier of the team that made the submission;
- T_j $(1 \le T_j \le p)$ — the problem number for which the submission was made;
- V_j $(0 \le V_j \le 1)$ — the result of the submission testing (1 indicates success, 0 indicates failure).

It is guaranteed that the submissions in the log are ordered chronologically, i.e., $M_j \le M_{j+1}$ for all $1 \le j < q$.

This also means that even if $M_j = M_{j+1}$, it should be considered that the j-th submission occurred **before** the $(j+1)$-th.

Output

In the i-th of n lines, output four integers R_i, ID_i, S_i, X_i separated by spaces:

- R_i $(1 \le R_i \le n)$ — the final rank of the team on the i-th line of the table;
- ID_i $(1 \le ID_i \le n)$ — the identifier of the team;
- S_i $(0 \le S_i \le p)$ — the number of problems solved by the team;
- X_i $(0 \le X_i \le 10^9)$ — the total penalty time of the team.

Examples

Input	Output
5 3 23 15 1 1 0 30 2 2 0 30 1 1 1 43 3 1 0 43 2 1 1 65 3 1 0 80 3 2 0 87 3 1 1 99 2 2 1 99 5 2 0 100 1 2 0 100 4 1 0 130 1 2 0 130 5 2 1 130 2 3 0 135 5 2 0 140 5 3 1 199 2 3 0 200 4 3 0 200 1 2 1 280 4 2 0 290 2 3 0 299 2 3 1	1 2 3 521 2 1 2 290 2 5 2 290 4 3 1 127 5 4 0 0

Note

First test example

A total of 5 teams participated in the competition, and 3 problems were available for solving.

- Team 1:
  - solved problem 1 at time 30, having made one penalty attempt before that — received 50 minutes of penalty.
  - solved problem 3 at time 200, having made two penalty attempts before that — received 240 minutes of penalty.
- Team 2:
  - solved problem 1 at time 43 with no penalty attempts — received 43 minutes of penalty.
  - solved problem 2 at time 99 with one penalty attempt — received 119 minutes of penalty.
  - solved problem 3 at time 299, having made three penalty attempts before that — received 359 minutes of penalty.
- Team 3:

  - solved problem 1 at time 87 with two penalty attempts — received 127 minutes of penalty.
  - made an incorrect attempt on problem 2, but did not solve it — did not receive penalty time.
- Team 4:
  - made one incorrect attempt on each problem, but did not solve anything — did not receive penalty time.
- Team 5:
  - solved problem 2 at time 130, having made one penalty attempt before that — received 150 minutes of penalty.
  - solved problem 3 at time 140 with no penalty attempts — received 140 minutes of penalty.
  - made another attempt on problem 2 after solving it — this did not affect its final rank in the table.

In total:

- Team 2 is in first place with 3 solved problems and 521 minutes of penalty.
- Team 1 is in second place with 2 solved problems and 290 minutes of penalty.
- Team 5 is in second place with 2 solved problems and 290 minutes of penalty.
- Team 3 is in fourth place with 1 solved problem and 127 minutes of penalty.
- Team 4 is in fifth place with 0 solved problems and 0 minutes of penalty.

Problem E. TL, ML or OK?

Time Limit 1000 ms
Mem Limit 262144 kB
OS Windows

Important:

- This problem contains very simplified information about what happens in real life.
- The values specified in the statement are approximate.
- In a real situation, the calculations may be much more complex and depend on more details.

Artistic problem statement

Aidar finally finished writing the solution to the problem — he was confident in its correctness but doubted its efficiency.

Aidar's solution worked as follows: on each of the n iterations of the loop, it performed q calculations and added k integers to the data structure.

Before submitting to the system, Aidar decided to show the final code to his teammates and ask for their opinions:

- Begimai said that the solution performs too many calculations — and therefore will not fit within the time limit.
- Viktor noted that the solution adds too many integers to the data structure — and therefore will not fit within the memory limit.
- Aidar himself still hoped that the team's estimates were too pessimistic and that the solution would pass all tests successfully.

Based on the available information about Aidar's solution, determine what verdict it will receive.

Fixed values used in the problem

It is known that in the problem Aidar is solving, the limits are set to 1 second and 256 megabytes.

- The number of operations that the language can perform in 1 second:
  - `C++`: $5 \cdot 10^8$;
  - `Java`: 10^8;
  - `Python`: $2 \cdot 10^7$.

- The number of integers that can fit in 256 megabytes: $2 \cdot 10^7$.
- Adding to the data structure takes 5 operations.
- A calculation takes 2 operations.

Input

The first line contains an integer n $(1 \leq n \leq 10^6)$ — the number of iterations of the loop in Aidar's solution.

The second line contains an integer q $(0 \leq q \leq 10^6)$ — the number of calculations performed on each iteration of the loop.

The third line contains an integer k $(0 \leq k \leq 10^6)$ — the number of integers added to the data structure on each iteration of the loop.

The fourth line contains a string L, denoting the programming language used for the solution:

- L is equal to `cpp` if Aidar wrote the solution in `C++`.
- L is equal to `java` if Aidar wrote the solution in `Java`.
- L is equal to `py` if Aidar wrote the solution in `Python`.

Output

In a single line, output the string V, denoting the expected verdict:

- V should be equal to `OK` if the solution does not exceed any of the problem limits.
- V should be equal to `TL` if the solution exceeds the time limit but does not exceed the memory limit.
- V should be equal to `ML` if the solution exceeds the memory limit but does not exceed the time limit.
- V should be equal to `TL and ML` if the solution exceeds both limits.

Examples

Input	Output
20000 2000 200 java	OK

Input	Output
100000 0 300 cpp	ML

Input	Output
1000000 1000000 0 py	TL

Input	Output
1000000 2000 200 java	TL and ML

Note

First test example

Aidar wrote the solution to the problem in `Java`.

In total, the solution will perform $2 \cdot 10^4$ iterations, on each of which:

- it will perform $2 \cdot 10^3$ calculations, which will total $8 \cdot 10^7$ operations;
- it will add 200 integers to the data structure, which will total $4 \cdot 10^6$ integers and $2 \cdot 10^7$ additional operations.

Thus, the solution:

- performs $8 \cdot 10^7 + 2 \cdot 10^7 = 10^8 \leq 10^8$ — the time limit is not exceeded.
- stores $4 \cdot 10^6 \leq 2 \cdot 10^7$ integers — the memory limit is not exceeded;

Accordingly, this solution will receive the verdict `OK`.

Second test example

Aidar wrote the solution to the problem in `C++`.

In total, the solution will perform 10^5 iterations, on each of which:

- it will perform 0 additional calculations;
- it will add 300 integers to the data structure, which will total $3 \cdot 10^7$ integers and $15 \cdot 10^7$ operations.

Thus, the solution

- performs $1.5 \cdot 10^8 \leq 5 \cdot 10^8$ operations — the time limit is not exceeded.
- stores $3 \cdot 10^7 > 2 \cdot 10^7$ integers — the memory limit is exceeded;

Accordingly, this solution will receive the verdict `ML`.

Third test example

Aidar wrote the solution to the problem in `Python`.

In total, the solution will perform 10^6 iterations, on each of which:

- it will perform 10^6 calculations, which will total $2 \cdot 10^{12}$ operations;
- it will add 0 integers to the data structure, which will not add any additional operations.

Thus, the solution

- performs $2 \cdot 10^{12} > 2 \cdot 10^7$ operations — the time limit is exceeded.
- stores nothing — the memory limit is not exceeded;

Accordingly, this solution will receive the verdict `TL`.

Fourth test example

This test is completely analogous to the first test, except for the increased number of loop iterations.

In total, the solution will perform $2 \cdot 10^6$ iterations, on each of which:

- it will perform $2 \cdot 10^3$ calculations, which will total $8 \cdot 10^9$ operations;
- it will add 200 integers to the data structure, which will total $4 \cdot 10^8$ integers and $2 \cdot 10^9$ additional operations.

Thus, the solution:

- performs $8 \cdot 10^9 + 2 \cdot 10^9 = 10^{10} > 10^8$ — the time limit is exceeded.
- stores $4 \cdot 10^8 > 2 \cdot 10^7$ integers — the memory limit is exceeded;

Accordingly, this solution may receive any of the verdicts — it is necessary to output `TL` and `ML`.

Problem F. Parallel Checking

Time Limit 2000 ms
Mem Limit 262144 kB
OS Windows

The competition has ended — and now Artem only needs to check the submissions of the teams for plagiarism.

Artem has a total of n submissions, and checking the i-th submission will take A_i quanta of time.

The problem is that the checking cannot take too long — it must be completed in no more than T quanta of time.

To speed up the process, Artem can parallelize it by allocating d threads. In this case, the i-th submission will be checked in $\lceil \frac{A_i}{d} \rceil$ quanta of time (the result of integer division, rounded up).

Help Artem find the **minimum** number of threads d such that $\sum \lceil \frac{A_i}{d} \rceil \leq T$.

Input

The first line contains two integers n and T $(1 \leq n \leq 2 \cdot 10^5; n \leq T \leq 10^{12})$ — the number of available submissions.

The second line contains n integers A_i $(1 \leq A_i \leq 10^{12})$ — the number of quanta of time required to check the i-th submission.

Output

In a single line, output an integer d $(1 \leq d \leq 10^{12})$ — the minimum number of threads such that $\sum \lceil \frac{A_i}{d} \rceil \leq T$.

Examples

Input	Output
5 22 15 130 120 78 43	20

Note

First test example

For $d = 20$, the submissions are checked in:

- $\lceil \frac{15}{20} \rceil = 1$;
- $\lceil \frac{130}{20} \rceil = 7$;
- $\lceil \frac{120}{20} \rceil = 6$;
- $\lceil \frac{78}{20} \rceil = 4$;
- $\lceil \frac{43}{20} \rceil = 3$.

In total, for $d = 20$, $1 + 7 + 6 + 4 + 3 = 21$ quanta of time are used.

For $d = 19$, the submissions are checked in:

- $\lceil \frac{15}{19} \rceil = 1$;
- $\lceil \frac{130}{19} \rceil = 7$;
- $\lceil \frac{120}{19} \rceil = 7$;
- $\lceil \frac{78}{19} \rceil = 5$;
- $\lceil \frac{43}{19} \rceil = 3$.

In total, for $d = 19$, $1 + 7 + 7 + 5 + 3 = 23$ quanta of time are used.

Similarly, it can be shown that for all $1 \leq d < 19$, the number of quanta of time is also at least 23.

Thus, it follows that for $T = 22$, the value $d = 20$ is the minimum possible.

Problem G. Compromise

Time Limit 2000 ms
Mem Limit 262144 kB
OS Windows

A few days before the competition, the problem authors Slava and Tolya got into a heated argument.

Slava claimed that the string S of length n is the best title for the problem.

Tolya insisted that the best title for the problem is the string T of length n.

To avoid ruining their friendship over some strings, the guys decided to find a string that would satisfy both of them.

They formulated the following criteria for the string—**compromise** P:

- P must also be of length n, just like the strings S and T.
- P must be a **palindrome** (i.e., read the same forwards and backwards)—this way, both Slava and Tolya can read it simultaneously from both sides.
- The total **distance** $d(S, P) + d(T, P)$ must be minimized (see note).

Help the guys—find the string P that satisfies the conditions described above.

Input

The first line contains an integer n $(1 \le n \le 5 \cdot 10^4)$—the length of the strings S and T.

The second line contains the string S $(|S| = n)$—the best title for the problem according to Slava.

The third line contains the string T $(|T| = n)$—the best title for the problem according to Tolya.

It is guaranteed that the strings S and T consist only of lowercase Latin alphabet characters.

Output

In a single line, output the string P—the compromise string that satisfies the following conditions:

- the length of P is n;
- P consists only of lowercase Latin alphabet characters.
- P is a palindrome.
- $d(S, P) + d(T, P)$ takes the minimum possible value.

If there are multiple strings P that satisfy the conditions—output any.

Examples

Input	Output

`4`
`axcy`
`pbqd`
	`kmmk`

Input	Output

`7`
`qwertyu`
`jhgfdsa`
	`oueieuo`

Note

Palindrome—a string that reads the same forwards and backwards.

Examples of palindromes: `tenet`, `abacaba`, `pqqp`, `d`.

Distance $d(A, B)$ between strings A and B of the same length n is calculated as $\sum |A_i - B_i|$, where $A_i - B_i$ is the difference in positions of characters A_i and B_i in the Latin alphabet.

First test example

The string S is `axcy`.

The string T is `pbqd`.

Examples of possible strings P: `kmmk`, `oddo`, `eeee`, and many others.

The total distance $d(S, P) + d(T, P)$ for P = `kmmk` will be:

- $d(S, P) = |a - k| + |x - m| + |c - m| + |y - k|$ = $|1 - 11| + |24 - 13| + |3 - 13| + |25 - 11| = 10 + 11 + 10 + 14 = 45$.

- $d(T, P) = |p - k| + |b - m| + |q - m| + |d - k|$ = $|16 - 11| + |2 - 13| + |17 - 13| + |4 - 11| = 5 + 11 + 4 + 7 = 27$.
- Accordingly, the total distance will be $45 + 27 = 72$.

It can be shown that 72 is the minimum possible value for suitable strings P.

Problem H. A + B = C

Time Limit 3000 ms
Mem Limit 262144 kB
OS Windows

In search of inspiration for problems, Slava went to scour the vastness of the internet — and found an interesting riddle.

The riddle took the form $A + B = C$, where the digits of the numbers A, B, and C were replaced by letters from the Latin (or other) alphabet.

Moreover, three additional rules applied:

- The numbers in the original identity do not have leading zeros and are not equal to zero.
- Each letter corresponds to exactly one replaceable digit.
- Each digit corresponds to exactly one replacing letter.

The solution to such riddles is any substitution "letter — digit" that transforms the expression into an identity.

For the given riddle of the described form, you need to:

- calculate the number of solutions to this riddle.
- if there is at least one solution, output any of them.

Input

The first line contains the string SA $(1 \le |SA| \le 3)$ — the string corresponding to the number A in the riddle.

The second line contains the string SB $(1 \le |SB| \le 3)$ — the string corresponding to the number B in the riddle.

The third line contains the string SC $(1 \le |SC| \le 4)$ — the string corresponding to the number C in the riddle.

It is guaranteed that the strings SA, SB, and SC consist only of uppercase Latin alphabet letters.

Output

In the first line, output an integer R $(0 \leq R \leq 10^9)$ — the number of solutions to this riddle.

If $R > 0$, then in the next three lines output any of the solutions to the riddle:

- the number A, represented by the string SA;
- the number B, represented by the string SB;
- the number C, represented by the string SC.

Examples

Input	Output
ABC DEF AAAZ	72 123 987 1110

Input	Output
AAA BBB CDC	0

Input	Output
PQ ST PV	0

Input	Output
X Y X	0

Note

First test example

The riddle `ABC + DEF = AAAZ` can be solved, for example, as $123 + 987 = 1110$:

- A corresponds to 1;
- B corresponds to 2;
- C corresponds to 3;

- D corresponds to 9;
- E corresponds to 8;
- F corresponds to 7;
- Z corresponds to 0.

Examples of other possible solutions:

- $147 + 963 = 1110$;
- $167 + 948 = 1115$.

Second test example

The riddle `AAA + BBB = CDC` has no solutions, as in this case

- either the same digit corresponds to two letters C and D;
- or at least two different digits correspond to the same letter (A or B).

Third test example

The riddle `PQ + ST = PV` has no solutions, as in this case the number represented by `ST` must have a leading zero, which is not allowed.

Fourth test example

The riddle `X + Y = X` has no solutions, as in this case the number represented by `Y` must equal zero, which is not allowed.