Problem A. Combinatory logic

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Combinatory logic is a field in mathematical logic developed in the first half of the XX century by Moses Sheinfinkel and Haskell Curry. Main elements of combinatory logic are constants, variables and terms. To construct terms we use the following induction rules:

1. if c is a constant then c is a combinatory term;
2. if x is a variable then x is a combinatory term;
3. if A and B are combinatory terms then (AB) is a combinatory term;
4. there are no other combinatory terms.

In this definition the expression (AB) denote the operation of application that is function (combinator) A being applied to its argument B. We denote variables by lowercase letters and combinators by upper case letters. In our case only the cobinators can be constants. To decrease number of brackets in expressions we make an assumption that the missed brackets must be placed according to left associativety rule:

$$ABC = ((AB)C)$$

So we assume that "ABC"is the correct form of combinatory term.

We choose as axioms the combinators I, K, S defined by the following rules:

- $Ia = a$;
- $Kab = a$;
- $Sabc = ac(bc)$.

We won't go deep in to details of combinatory logic and will compute the combinatory terms using the following rules:

- To compute combinator I we take next term, for K we take two next terms, for S three next terms. If there are not enough terms the string will not change.
- Each time we compute leftmost combinator or left combinator in some bracket.
- We will omit left brackets if they are present.

Examples:

$$S(KII)ab = S(I)ab = (I)b(ab) = b(ab);$$

$$SIISa = IS(IS)a = SSa;$$

$$S(KK)Iab = (KK)a(Ia)b = KKa(Ia)b = K(Ia)b = Kab = a.$$

Given the combinator P written as expression in I, K, S. Compute the application of P to the sequence containing the first n lowercase Latin letters.

Input

The first line of the input file contains one integer number n ($1 \le n \le 8$). The second line contain the description of combinator P (the sting of symbols "I", "K", "S", "(" and ")"). P must be a combinatory term. The string's length is at least 1 and at most 150 000. The maximum nesting level of brackets can not exceed 100. The number of combinators in computation can not exceed 50 000.

Output

The output file contains the resulting string of application of P to the sequence of the first n Latin letters. The string length will not exceed 8 and it will contain lowercase Latin letters.

Examples

input.txt	output.txt
1 I	a
2 K	a
1 SII	aa
1 S(SII)I	aaa
3 S((S(KS)K)(S(KS)K)S)(KK)	acb
2 SS(K(SKK))	abb

Problem B. Sequences

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Let's consider the sequences of n symbols. We say the sequence is k-simple if the symbols on positions i and $i+k$ $(1 \le i \le n-k)$ are different. Determine the quantity of k-simple sequences consisting of the first m Latin letters.

Input

The first line of the input file has three integer numbers n $(1 \le n \le 40)$, m $(1 \le m \le 8)$ and k $(1 \le k \le 8)$.

Output

The first line of the output file has one integer number equal to the quantity of such sequences.

Examples

input.txt	output.txt
3 2 2	4

In that example the sequences are: aab, abb, baa, bba.

Problem C. Stone game

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Two players play the following game. There are n heaps of stones, i-th heap has a_i stones. The players make moves in turn. On the each turn the active player chooses the heap and removes an arbitrary positive number of stones. The player removing the last stone will loose the game. For the given position determine if the first player will win and if it is true describe the winning move. Two players play optimally.

Input

The first line of the input file has one integer number n ($1 \le n \le 100$) equal to the number of heaps. The second line has n integer numbers a_i ($1 \le a_i \le 100$) that is equal to number of stones in the i-th heap.

Output

The first line of the output file is "Lose" of the best strategy for both players lead to the lost game for the first player and "Win" otherwise. If output is "Win" than the second line contains two integer numbers equal to the number of the heap and number of stones to be removed on the first turn, respectively.

Examples

input.txt	output.txt
2 3 3	Lose
2 3 2	Win 1 1
3 3 3 3	Win 1 3

Problem D. Equal Vertices

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Compare all vertices of a finite vertex-labeled directed graph in which every vertex has at most one outgoing edge.

Two vertices are equal if they cannot be distinguished by the following process:

1. Compare labels. Whenever they differ, the process stops and reports vertices are not equal.
2. Compare outgoing degrees. Whenever they differ, the process stops and reports vertices are not equal.
3. When both vertices have no outgoing edges, the process stops and reports vertices are equal.
4. Follow outgoing edges and repeat the process.

Please note, when the process reports vertices are equal, it implies that vertices cannot be distinguished.

Input

The first line of the input file contains a single number n ($2 \le n \le 10^5$) — the number of vertices.

The following n lines describe vertices. Each description contains a lowercase English letter (vertex label) and number t separated by space. $t = 0$ means that vertex has no outgoing edges. Otherwise, it has an edge to t-th vertex. Vertices are numbered from 1 to n in the order they given.

Output

Write to the output file exactly n lines.

On the i-th line write the equality class number of the i-th vertex. Equal vertices should have the same class number.

Classes are numbered as follows: the first vertex belongs to the first class, the next vertex that doesn't belong to the first class belongs to the second class, and so on.

Example

input.txt	output.txt
4 a 2 b 3 b 0 b 3	1 2 3 2

Problem E. Pinball

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	4 seconds
Memory limit:	256 Mebibytes

Pinball is an arcade game. The player gets the points by manipulating the metal ball in the field by flippers. The main goal of the game is to get the maximum number of points.

The field contains many different objects but here we will use only bumpers. Geometrically the bumper is the disk of the given size. If the ball touches the bumper then player gets some points, the ball moves in any direction from bumper (it could possibly keep its previous direction) till it hits another bumper. The ball can move to the bumper that is visible from the previous one i.e. it is possible to draw the line between centers of the bumpers not intersection and not touch other bumpers. The ball can not return to the bumper that it touched before going to the current one. In other words if ball hit bumper a then bumper b then it can not go back to a.

Let the center of the first bumper has the coordinates (x_1, y_1) and its radius is equal to r_1, the center and radius of the second bumper are x_2, y_2 and r_2 correspondingly. Then the travel time between the bumpers is equal to $\lceil \frac{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}-r_1-r_2}{20} \rceil$ seconds (rounded up to nearest integer). We assume that game starts at $t = 0$ and the ball hit the top bumper (the center of bumper has the maximum y coordinate; if there are several such bumpers we take the bumper with maximum y coordinate and maximum x coordinate) .

Write a program that will process two types of queries for the given field.

Input

The first line of the input file contains the integer number n ($3 \le n \le 7$) equal to the number of bumpers. The following n lines each contain 4 integers x_i, y_i, r_i и s_i ($0 \le x_i, y_i \le 50$, $1 \le r_i \le 5$, $-100 \le s_i \le 100$) defining coordinates of i-th bumper, its radius and number of points that player gets when the ball hit that bumper. Bumpers don't intersect and don't touch each other. The ball is has some direction to go.

The next line contain the number q ($1 \le q \le 10\,000$) equals to the number of queries. The next q strings contain the integer number defining type of the query and another integer t_j ($0 \le t_j \le 10^9$) for the query of the first type and two integers t_j и x_j ($0 \le t_j \le 10^9$, $1 \le x_j \le n$) for the query of the second type.

Output

For each query of the first type write the integer number equal to the maximum number of game points for the first t_i seconds.

For each query of the second type write the integer number equal to the maximum number of game points on t_j second of the game of ball previously hit the x_j bumper. If that situation is impossible then write symbol "`#`".

All answers to the queries must be placed in different lines.

Examples

input.txt	output.txt
3 0 0 1 1 50 0 1 1 0 50 1 1 20 1 1 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 2 1 1 2 2 1 2 3 1 2 4 1 2 5 1 2 6 1 2 7 1 2 8 1 2 9 1 2 10 1	1 1 2 2 2 3 3 3 3 4 # # 2 2 2 # 3 3 3 #

Problem F. Combinatory logic 2

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Combinatory logic is a field in mathematical logic developed in the first half of the XX century by Moses Sheinfinkel and Haskell Curry. Main elements of combinatory logic are constants, variables and terms. To construct terms we use the following induction rules:

1. if c is a constant then c is a combinatory term;
2. if x is a variable then x is a combinatory term;
3. if A and B are combinatory terms then (AB) is a combinatory term;
4. there are no other combinatory terms.

In this definition the expression (AB) denote the operation of application that is function (combinator) A being applied to its argument B. We denote variables by lowercase letters and combinators by upper case letters. In our case only the cobinators can be constants. To decrease number of brackets in expressions we make an assumption that the missed brackets must be placed according to left associativety rule:

$$ABC = ((AB)C)$$

So we assume that "ABC"is the correct form of combinatory term.

We choose as axioms the combinators I, K, S defined by the following rules:

- $Ia = a$;
- $Kab = a$;
- $Sabc = ac(bc)$.

We won't go deep in to details of combinatory logic and will compute the combinatory terms using the following rules:

- To compute combinator I we take next term, for K we take two next terms, for S three next terms. If there are not enough terms the string will not change.
- Each time we compute leftmost combinator or left combinator in some bracket.
- We will omit left brackets if they are present.

Examples:

$$S(KII)ab = S(I)ab = (I)b(ab) = b(ab);$$

$$SIISa = IS(IS)a = SSa;$$

$$S(KK)Iab = (KK)a(Ia)b = KKa(Ia)b = K(Ia)b = Kab = a.$$

Lets assume that we have applied P to the sequence containing the first n lowercase Latin letters and obtain the sequence X. Given the sequence X and number n find any expression of P in combinators I, K, S.

Input

The first line of the input file contains one integer number n ($1 \le n \le 8$). The second line contains a nonempty sequence up to 8 symbols. It can contain only the first n lowercase Latin letters.

Output

The output file contains the string with symbols "I", "K", "S", "(" and ")" describing combinatory term P not exceeding 400 000 symbols. The maximum nesting level of brackets must not exceed 150 and the number of combinators in computation must not exceed 150 000. Assume that answer is always exists.

Examples

input.txt	output.txt
1 a	I
2 a	K
1 aa	SII
1 aaa	S(SII)I
3 acb	S((S(KS)K)(S(KS)K)S)(KK)
2 abb	SS(K(SKK))

Problem G. Minimum volume tetrahedron

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

In three dimensional space given the angle $OABC$ with the vertex $O(0,0,0)$ and points $A(a_1,a_2,a_3)$, $B(b_1,b_2,b_3)$, $C(c_1,c_2,c_3)$ on the edges (a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3 are integer positive numbers). Coordinates are given in Cartesian coordinate system. Point $P(p_1,p_2,p_3)$ (p_1, p_2, p_3 — integer possitive numbers) is placed strictly inside angle $OABC$ (i.e. the vector OP has positive elements in basis OA, OB, OC). Draw the plane through point P to cut the tetrahedron with the minimum volume from the given angle.

Input

Four lines of the input file contains the coordinates of points A, B, C, P separated by spaces. Coordinates are integer numbers not exceeding 10^3. Vectors OA, OB и OC are linearly independent.

Output

The output file contain the number equal to minimum volume of tetrahedron cut by plane from the given angle with precision 10^{-5}.

Examples

input.txt	output.txt
1 2 3 2 3 1 2 5 3 2 4 3	2.53125

Problem H. Inscribed triangle 3

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Lets take an arbitrary nondegenerate triangle. Draw the closed broken line with three segments of minimal length so that each side of triangle contain the vertex of that line and each vertex of the line is placed on some side of triangle. Is possible to have the lengths of segments of the broken line equal to a, b, c?

Input

The first line of the input file contains three integer numbers a, b и c ($-1\,000 \le a, b, c \le 1\,000$).

Output

The output file contain the line "Yes" if the lengths of segments of the broken line can be equal to a, b, c and "No" otherwise.

Examples

input.txt	output.txt
2 3 4	Yes
-1 -1 -1	No

Problem I. $A^2 + \cdots + B^2$

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

$$N = \sum_{k=A}^{B} k^2$$

Input

The input file contains a single line with two integer numbers A and B ($-10^{10^6} \le A \le B \le 10^{10^6}$) separated by space.

Output

Write to the output file single line containing digital root[1] of the number N.

Example

input.txt	output.txt
1 2	5

[1] The digital root of a number is the number obtained by adding all the digits, then adding the digits of that number, and then continuing until a single-digit number is reached.

Problem J. Birthdays

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Misha wants to get a job in the Kolya department. He doesn't know the number of clerks in the firm and assumes that there could be any number of clerks with equal probability of cases.

To verify Misha mathematical abilities Kolya asked him to determine the number of clerks knowing only the number of clerks with the same birthdays.

All clerks can be separated in groups with same birthday in same group and different birthdays in different groups. There are m_2 groups of two clerks, m_3 groups of 3 clerks, ..., m_s groups of s clerks and other groups each contain one clerk.

Assuming that there are d days in any year and probability to have birthday is equal for any day in the year help Misha find the most probable number of clerks in the firm.

Input

The first line of the input file contains two integer numbers s and d ($365 \le d \le 40\,000$, $2 \le s \le d$). The second line contains $s-1$ integer numbers m_2, m_3, ..., m_s ($0 \le \sum_{i=2}^{s} m_i \le d$).

Output

The output file must contain the positive integer number n with the most probable number of clerks. If there are several possible answers write the maximum number.

Examples

input.txt	output.txt
2 365 1	28
4 365 3 5 2	113

Problem K. Radio towers

Input file:	input.txt
Output file:	output.txt
Time limit:	8 seconds
Memory limit:	256 Mebibytes

The military base has several buildings in the field and there were no connections between them. To correct this situation army decides to build some radio towers in the field and radio tower on top of each building. Then the signals will pass from one tower to the other. Yet the budget spending must be minimal.

Formally approaching we have the table $n \times m$. Each cell is either empty or contains the building. In any empty cell we can put a tower and for all buildings we must put towers. Each tower has a power from 1 to 9. That number determines the distance that would be covered by that tower (passing signals to other towers). We assume that the distances are Euclidean. The cost of putting each tower in the cell is equal to a. If a tower has the power p we add p^2 to the cost.

Determine the places and parameters of radio towers to pass signals from any building to any other building (maybe through some proxies) to have minimal summary cost.

Input

The first line of the input file contains two integer numbers n и m ($1 \le n, m \le 20$, $2 \le n \times m \le 25$).

Next n lines contain m symbols each. They describe the field the following way: "`.`" is an empty cell, "`*`" is a cell with building. The field must contain at least 2 buildings.

The last line contains one integer number a ($1 \le a \le 500$).

Output

The output file must contain n lines of m symbols the field with towers, where "`.`" is an empty cell, "`x`" is a cell with a tower having power x where x is an integer number from 1 to 9.

Examples

input.txt	output.txt
3 3 ..* ... *.. 4	..2 .2. 2..
5 4 *... ...* .... *... ...* 9	1... 3..3 .... 2... ...3