A. Biology

time limit per test: 2.0 s
memory limit per test: 512 MB
input: standard input
output: standard output

Vera has $A \times B$ cards. Each card has a *rank*, an integer between 0 and A - 1, and a *suit*, an integer between 0 and B - 1. All cards are distinct. A set of five different cards is known as a *hand*. Each hand is in exactly one of nine categories numbered from 1 to 9. If a hand satisfies the conditions for membership in multiple categories, it is considered to be in the lowest-numbered such category. The rules for each category are:

1. Straight flush: is a Straight and a Flush.
2. Four of a kind: four of the cards have the same rank.
3. Full house: three of the cards have the same rank and the remaining two have the same rank.
4. Flush: all five cards have the same suit.
5. Straight: the ranks of the cards in increasing order are $x, x+1, x+2, x+3, x+4$ for some integer x.
6. Three of a kind: three of the cards have the same rank.
7. Two pair: two cards have the same rank and two other cards have the same rank.
8. One pair: two cards have the same rank.
9. High card: if a hand does not satisfy any other category.

Currently, Vera has two cards with ranks a_1, a_2 and suits b_1, b_2. Of the remaining cards, Vera will choose three more cards and form a hand with her two current cards. Compute the number of different hands formed in this way that belong in each category.

Input

Line 1 contains integers A and B $(5 \le A \le 25, 1 \le B \le 4)$.

Line 2 contains integers a_1, b_1, a_2, b_2 $(0 \le a_1, a_2 \le A$ - $1, 0 \le b_1, b_2 \le B$ - $1, (a_1, b_1) \ne (a_2, b_2))$.

Output

Print one line with nine integers, the number of different of hands that belong in each category in increasing order of categories (from Straight flush to High card).

Examples

input
5 2 1 0 3 1
output
0 0 0 0 8 0 12 36 0

input
13 4 0 0 1 0
output
1 2 18 164 63 308 792 7920 10332

Note

Let (a, b) denote a card with rank a and suit b.

For the first example, Vera currently has cards $(1, 0)$ and $(3, 1)$. If she chooses additional cards $(3, 0), (4, 0), (4, 1)$, her hand will be a Two pair as there are two cards with rank 3 and two other cards with rank 4. Note that this hand also satisfies being a One pair, but Two pair is the lower-numbered category.

B. Programming a robot

time limit per test: 2.0 s
memory limit per test: 512 MB
input: standard input
output: standard output

The last edition of the world robot conference took place in China last year, the most recent innovations on robots were on display. Those ranged from machines able to make people's portraits to robots that can play soccer.

We want to program a robot, but a less sophisticated one. It takes only two commands:

- Walk(k): takes k steps forward
- TurnRight(): turns 90 degrees to its right.

At first, we just want to program this robot's movements. For that, we need you to print a sequence of commands to complete the following action: given the Cartesian coordinates of start and destination, and the initial direction the robot initially points to (North, South, East, or West), print a sequence of commands that takes the robot from its starting position to its destination.

Since we are always concerned about efficiency, we wish the sequence to be as small as possible.

Input

The input has one line with a pair of integers that indicate the starting coordinates, x_o and y_o, followed by a character that indicates a direction (`N` for North, `S` for South, `E` for East, or `O` for West). The next line contains a pair of integers indicating the destination coordinates, x_d and y_d.

- $0 \leq x_o, y_o \leq 5 \cdot 10^5$
- $0 \leq x_d, y_d \leq 5 \cdot 10^5$

Output

In the first line, print the smallest size of a sequence of commands that take the robot from start to destination. In the next lines, print one of these sequences, one command per line. For the command Walk((k)), print "A k" (without the double quotes). For the command TurnRight(), print "D".

If there is more than one sequence of instructions that work, any one of them will be accepted.

Examples

input
15 2 E 7 9
output
5 D D A 8 D A 7

input
0 0 N 0 12
output
1 A 12

C. Computer Science

time limit per test: 2.0 s
memory limit per test: 512 MB
input: standard input
output: standard output

Vera has N integers $a_1, ..., a_N$. A *margin* is a non-negative integer L such that it is possible to choose N integers $x_1, ..., x_N$ such that for all i, $1 \le i \le N$, the interval $[x_i, x_i + L]$ contains at least K of Vera's integers and also contains a_i.

Compute the minimum possible margin.

Input

Line 1 contains integers N and K $(1 \le K \le N \le 2 \times 10^5)$.

Line 2 contains N integers, $a_1, ..., a_N$ $(-10^9 \le a_i \le 10^9)$.

Output

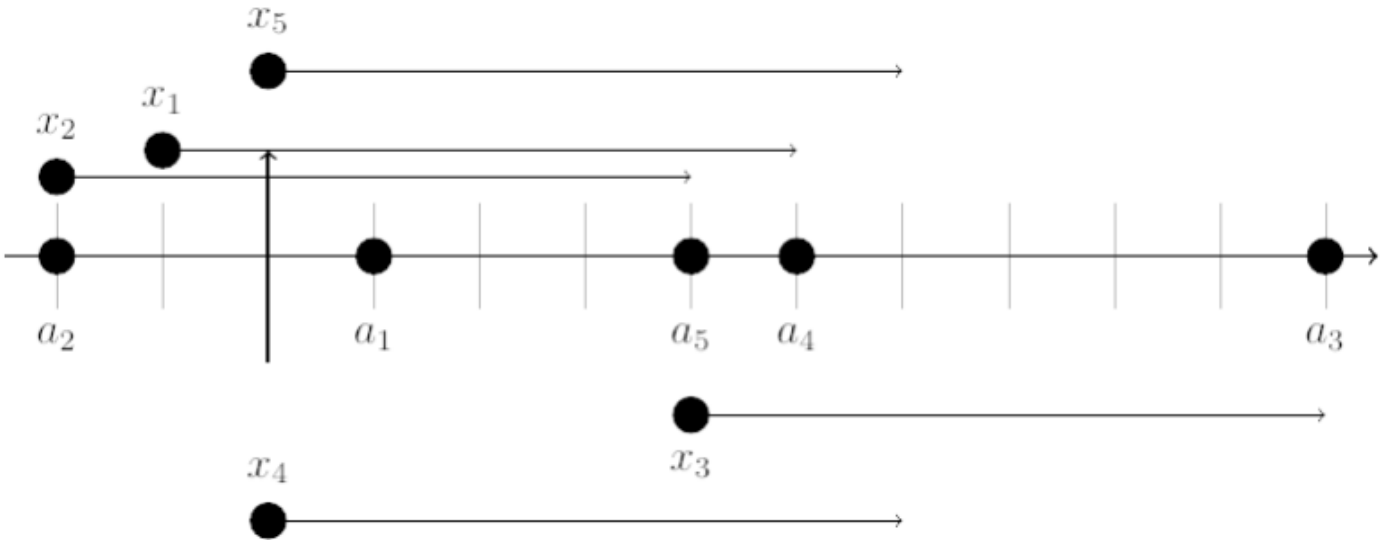
Print one line with one integer, the minimum possible margin.

Example

input
5 3 1 -2 10 5 4
output
6

Note

For the first example, one possible solution is to choose $x_1 = -1, x_2 = -2, x_3 = 4, x_4 = 0, x_5 = 0$, which is illustrated below.



D. Charles in Charge

Every day, Charles drives from his home to work and back. He uses the highways of the country that run from one city to another. Charles has decided that he wants to help the environment by buying an electrical car. Electrical cars, however, are not very common in his country yet. They can only be charged inside a city; there are no charging stations along the highways in between the cities. Moreover, all electrical cars are identical except for one thing: the size of the battery. As batteries are very expensive, Charles would like to buy a car with battery that is as small as possible.

Picture by Frank Hebbert via Flickr

However, this greatly increases the time it takes for him to get home, much to the distaste of his wife, Charlotte. This has spawned an argument, and after much discussion they have decided to compromise: Charlotte is fine with Charles taking a longer route, as long as it its length is at most $X\%$ longer than the length of shortest route that Charles could have taken to get home from work by using a regular car. Charles has agreed with this, and he now wants to find a route that minimizes the size of the car battery that he needs, i.e. the route that minimizes the maximum distance that Charles has to drive on a highway without passing through a city.

The amount of time Charles spends to charge his car can be neglected.

Input

The input starts with integers $2 \leq N \leq 10\,000$, $1 \leq M \leq 100\,000$ and $0 \leq X \leq 10\,000$: the number of cities, the number of highways connecting the cities and the aforementioned percentage X. City 1 is the place where Charles lives and city N is where he works.

Then follow M lines with on each line three integers: $1 \leq C_1 \leq N$, $1 \leq C_2 \leq N$, $1 \leq T \leq 10^9$. This means that there is a highway of length T connecting cities C_1 and C_2 (Charles can traverse the highway in both directions) *without* passing through any other cities. You may assume that there exists a path from city 1 to city N.

Output

The output is a single integer: the smallest maximum distance that Charles has to travel on a highway without passing through a city, such that the route he takes is at most $X\%$ longer than the shortest route.

Sample Input 1	Sample Output 1
2 1 100 1 2 5	5

Sample Input 2	Sample Output 2
9 8 15 1 9 16 1 4 4 4 5 4 5 6 4 6 8 4 4 7 5 7 8 5 8 9 4	5

Sample Input 3	Sample Output 3
9 8 30 1 9 16 1 4 4 4 5 4 5 6 4 6 8 4 4 7 5 7 8 5 8 9 4	4

Explanation of samples

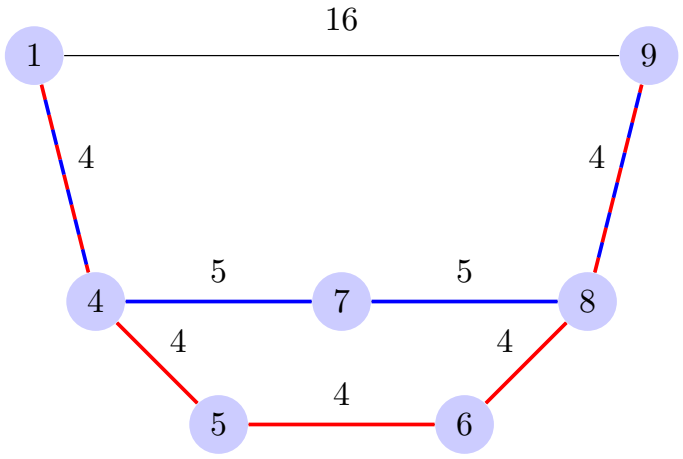


Figure 2: The graph of the second and third test case. The shortest path has length 16. In the second testcase, Charles's travel distance may not exceed $16 \cdot 1.15 = 18.4$ distance units. As a result, he cannot use a battery with which he can travel 4 distance units, as the red path 1–4–5–6–8–9 has length 20. Therefore, he uses the blue path 1–4–7–8–9, which has length 18, and the longest edge has length 5. In the third test case, he is allowed to travel $16 \cdot 1.30 = 20.8$ distance units, so he can follow the red path, where the longest edge has length 4.

Problem E. Twisting the Number

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	64 megabytes

Professor Dull invented a direction in number theory. This is the theory of "twisting generators". Consider a positive integer number p. Let it contain b bits (the highest bit should be 1). Consider all possible b cycle shifts of the binary notation of the number p. Let's make a set from all this shifts and call it $W(p)$. Note, some of the shifts can start with zero. Let's say that the number p twistingly generates the set $W(p)$. For example, $W(11) = \{7, 11, 13, 14\}$.

The number p is called a twisting generator of the number n, if the union of all sets $W(1), W(2), \ldots, W(p)$ contains $\{1, 2, \ldots, n\}$ as a subset. Your task is to find the minimum generator of the given number n.

Input

The first line of the input contains the integer number n ($1 \le n \le 10^{18}$).

Output

Write to the output the minimum twisting generator of the number n.

Example

stdin	stdout
6	5