TASK	JACK	PROZORI	ZAGRADE	REZ	PASTELE	KOŠARE

source code	`jack.pas`
	`jack.c`
	`jack.cpp`
		`prozori.pas`
		`prozori.c`
		`prozori.cpp`
			`zagrade.pas`
			`zagrade.c`
			`zagrade.cpp`
				`rez.pas`
				`rez.c`
				`rez.cpp`
					`pastele.pas`
					`pastele.c`
					`pastele.cpp`
						`kosare.pas`
						`kosare.c`
						`kosare.cpp`

input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 second	1 second	1 second	1 second	5 seconds	2 second
memory limit	32 MB	32 MB	32 MB	32 MB	256 MB	256 MB
point value	**50**	**80**	**100**	**120**	**140**	**160**
	650					

In "Blackjack", a popular card game, the goal is to have cards which sum up to largest number not exceeding 21. Mirko came up with his own version of this game.

In Mirko's game, cards have positive integers written on them. The player is given a set of cards and an integer **M**. He must choose **three** cards from this set so that their sum comes as close as possible to **M** without exceeding it. This is not always easy since there can be a hundred of cards in the given set.

Help Mirko by writing a program that finds the best possible outcome of given game.

INPUT

The first line of input contains an integer **N** (3 ≤ **N** ≤ 100), the number of cards, and **M** (10 ≤ **M** ≤ 300 000), the number that we must not exceed.

The following line contains numbers written on Mirko's cards: **N** distinct space-separated positive integers less than 100 000.

There will always exist some three cards whose sum is not greater than **M**.

OUTPUT

The first and only line of output should contain the largest possible sum we can obtain.

SAMPLE TESTS

input	**input**
5 21 5 6 7 8 9	10 500 93 181 245 214 315 36 185 138 216 295
output	**output**
21	497

Now that spring is here and the sun is shining bright, people are starting to lower their blinds. Štefica is an elderly woman who likes to keep track of what other people in the neighbourhood are doing and then talk about it behind their backs. This year, she is particularly interested in who is lowering blinds in the building across the street, and how low are they lowering them.

We will represent each window with a 4 x 4 grid, with asteriskes representing lowered blinds. Štefica can see a window in one of the following 5 states:

`....`
`....`
`....`
`....`
	`****`
	`....`
	`....`
	`....`
		`****`
		`****`
		`....`
		`....`
			`****`
			`****`
			`****`
			`....`
				`****`
				`****`
				`****`
				`****`

The building across the street has **N** windows at each of the **M** floors. Given the current building state, find out how many windows are in each of the 5 states shown above.

INPUT

The first line of input contains space separated integers **M** and **N** (1 ≤ **M**, **N** ≤ 100).

The following lines describe the current building state. Each window is represented with one of the 4 x 4 grids shown above, and windows are separated using character '#'. See the example input for clarification. Building description will have exactly 5**M** + 1 lines each having 5**N** + 1 characters.

OUTPUT

Output should contain 5 space separated integers, number of windows for each type in order shown above. Sum of these numbers is **M*N**.

SAMPLE TESTS

input

```
1 2
###########
#....#****#
#....#****#
#....#....#
#....#....#
###########
```

output

```
1 0 1 0 0
```

input

```
2 3
################
#****#****#****#
#****#****#****#
#****#....#****#
#....#....#****#
################
#....#****#****#
#....#****#....#
#....#....#....#
#....#....#....#
################
```

output

```
1 1 2 1 1
```

Mirko was bored at his chemistry class, so he played Bomb Switcher on his cell phone. Unfortunately, he was spotted and was given a ridiculously heavy assignment for homework. For a given valid math expression with brackets, he must find all **different** expressions that can be obtained by removing valid pairs of brackets from the original expression. Two expressions are different if there is a character at which they differ.

For example, given (2+(2*2)+2), one can get (2+2*2+2), 2+(2*2)+2, and 2+2*2+2. (2+2*2)+2 and 2+(2*2+2) can't be reached, since we would have to remove pairs of brackets that are not valid. More than one pairs of brackets can surround the same part of the expression.

INPUT

The first and only line of input contains one valid mathematical expression composed of nonnegative integers, basic arithmetic operations denoted with characters '+', '*', '-' and '/', and brackets '(' and ')'.

Given expression won't have more than 200 characters, and will have at least one, and no more than 10 pairs of brackets. Each expression is guaranteed to have at least one pair of brackets.

OUTPUT

Output all different expressions that can be obtained by removing valid pairs of brackets, sorted lexicographically.

SAMPLE TESTS

input	**input**	**input**
(0/(0))	(2+(2*2)+2)	(1+(2*(3+4)))
output	**output**	**output**
(0/0) 0/(0) 0/0	(2+2*2+2) 2+(2*2)+2 2+2*2+2	(1+(2*3+4)) (1+2*(3+4)) (1+2*3+4) 1+(2*(3+4)) 1+(2*3+4) 1+2*(3+4) 1+2*3+4

Let's say that there exists a huge cake made from blueberries, strawberries and chocolate. It's shaped like a square, and has area of 100 square meters. Proffesionals strongly advise that cake is being cut with wet knife and eaten with dry spoon. Also:

- Every cut begins and ends on the cake's perimeter
- A cut cannot lie completely on one of the sides
- No two cuts have the same starting and ending points, i.e. all cuts are different

Parts obtained by these cuts are separated and counted only after last cut has been made. During cutting, the cake keeps its square form.

At least how many cuts need to be made in order to obtain **at least K** parts? Exactly what cuts to make?

INPUT

The first and only line of input contains an integer **K** (1 ≤ **K** ≤ 1 000 000), minimum number of parts that we must have after cutting is done.

OUTPUT

The first line of output should contain the requested number of cuts, **N**.

The following **N** lines should have four integers each, coordinates of starting and ending point for each cut made. Coordinates are represented in millimeters, and opposing corners of the cake have coordinates (-5000, -5000) and (5000, 5000). So for each point (**x**, **y**) lying on the side of the square, the following will hold:

$$\max(|\mathbf{x}|, |\mathbf{y}|) = 5000.$$

SCORING

If only the number of cuts **N** is correct, you will get 50% of the points for that test case.

SAMPLE TESTS

input	**input**	**input**
1	4	7
output	**output**	**output**

0	2
	-5000 -5000 5000 5000
	5000 -5000 -5000 5000
		3
		-5000 5000 0 -5000
		-2000 -5000 5000 5000
		-5000 0 5000 0

Mirko recently got **N** crayons as a gift. The color of each crayon is a combination of three primary colors: red, green and blue. The color of the **i**th crayon is represented with three integers: $\mathbf{R}_i$ for the red, $\mathbf{G}_i$ for the green and $\mathbf{B}_i$ for the blue component.

The **difference** between the **i**th and the **j**th crayon is max(|$\mathbf{R}_i$ - $\mathbf{R}_j$|, |$\mathbf{G}_i$ - $\mathbf{G}_j$|, |$\mathbf{B}_i$ - $\mathbf{B}_j$|). The **colorfulness** of a subsequence of crayons is equal to the largest difference between any two crayons in the subsequence.

Mirko needs a subsequence with **K** crayons with the smallest colorfulness for his drawing. The subsequence does not have to be consecutive. Find it!

INPUT

The first line of input contains integers **N** and **K** ($2 \le \mathbf{K} \le \mathbf{N} \le 100\,000$).

The **i**th of the folowing **N** lines contains three integers $\mathbf{R}_i$, $\mathbf{G}_i$ and $\mathbf{B}_i$ ($0 \le \mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i \le 255$).

OUTPUT

The first line of output should contain the smallest colorfulness of a subsequence with **K** crayons.

The following **K** lines should contain the **R**, **G** and **B** values of the colors of the crayons in the subsequence, in any order. Any subsequence that yields the smallest colorfulness will be accepted.

SCORING

In test cases worth 50% of total points, $0 \le \mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i \le 20$ will hold.

In test cases worth additional 30% of total points, $0 \le \mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i \le 50$ will hold.

SAMPLE TESTS

input 2 2 1 3 2 2 6 4 **output** 3 1 3 2 2 6 4	**input** 3 2 3 3 4 1 6 4 1 1 2 **output** 2 3 3 4 1 1 2	**input** 5 3 6 6 4 6 2 7 3 1 3 4 1 5 6 2 6 **output** 2 6 2 7 4 1 5 6 2 6

Mirko found **N** boxes with various forgotten toys at his attic. There are **M** different toys, numbered 1 through **M**, but each of those can appear multiple times across various boxes.

Mirko decided that he will **choose some boxes** in a way that there is **at least one toy of each kind** present, and throw the rest of the boxes away.

Determine the number of ways in which Mirko can do this.

INPUT

The first line of input contains two integers **N** and **M** ($1 \le \mathbf{N} \le 1\ 000\ 000$, $1 \le \mathbf{M} \le 20$).

Each of the following **N** lines contains an integer $\mathbf{K}_i$ ($0 \le \mathbf{K}_i \le \mathbf{M}$) followed by $\mathbf{K}_i$ distinct integers from interval [1, **M**], representing the toys in that box.

OUTPUT

The first and only line of output should contain the requested number of ways modulo 1 000 000 007.

SCORING

In test cases worth 50% of total points, $\mathbf{N} \le 100$ and $\mathbf{M} \le 15$ will hold.

In test cases worth 70% of total points, $\mathbf{N} \le 1\ 000\ 000$ and $\mathbf{M} \le 15$ will hold.

SAMPLE TESTS

input 3 3 3 1 2 3 3 1 2 3 3 1 2 3 **output** 7	**input** 3 3 1 1 1 2 1 3 **output** 1	**input** 4 5 2 2 3 2 1 2 4 1 2 3 5 4 1 2 4 5 **output** 6