Problem A. Standings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Algorithms and data structures course is over. All exams have passed, and MSAI teachers are going to analyse the result. The number of students in MSAI program this year was $3 \le N \le 1000$. Teachers have an excel table with the following information about each student:

- unique student id;
- student score for the course;
- Student nickname.

Unfortunately, teachers don't know how to sort tables in excel, and they decided to ask you to write program to do that.

You need to provide students ids in non-increasing score order. If two students have identical score, one who have less id should go first (see Sample 3).

Also, teachers want your program to print nicknames of three students who took 1st, 2nd and 3rd place in sorted table to give them a prize.

Please, use any Quadratic sorting algorithm covered in lecture in this problem solution.

Input

First line contains single integer number: $3 \le N \le 1000$. The following N lines contain information about students. i-th line contains the following:

1. Integer number $0 \le id < N$ (unique student id);
2. Space character;
3. Integer number $0 \le score \le 10^9$ (student score for the course);
4. Space character;
5. String which consists of $1 - 20$ latin letters (A-Z, a-z): (student nickname).
6. Line break character (`\n`);

Output

First three lines should contain nicknames of 3 prize-winners (students who took 1st, 2nd and 3rd place in sorted table). First line should contain N integer numbers divided by space character — ids of students in non-increasing score order. If two students have identical score, one who have less id should go first (see Sample 3).

Examples

standard input	standard output
5 0 100 Kermit 1 0 Pepe 2 999999999 Knuth 3 1000000000 Dijkstra 4 100000000 Kormen	Dijkstra Knuth Kormen 3 2 4 0 1
4 3 10 McDonald 1 50 Ivan 0 1000 AdaLovelace 2 100 Radoslav	AdaLovelace Radoslav Ivan 0 2 1 3
3 2 1 a 1 1 b 0 1 c	c b a 0 1 2

Problem B. QSort Partition

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

This year number of students in MSAI program is enormous: $0 \leq N \leq 100000$. Quadratic sorting algorithm you used in previous problem now is too slow. MSAI teachers ask you for help again. Now you don't need to sort tables. You are asked just to implement fast sorting algorithm.

You can try to submit quadratic sorting algorithm here to make sure it's too slow.

You are asked to implement QSort algorithm. To simplify your task, teachers prepared code sample for you (link to github repo shortened for convenience): `https://bit.ly/3lWoFnA`

It already has input and output parts. Also it has qsort main function implemented. You need to implement `partition` function which divides an array into 3 parts ($p_<, p_=, p_>$). This function should give you correct QSort algorithm.

Please, find partition function signature and description to follow in the sample file.

Solution will be tested as sorting algorithm.

Input

First line contains single integer number: $0 \leq N \leq 100000$. Next line contains N integer numbers divided by space character: $0 \leq x_i \leq 10^9$.

Output

On single line you should print N integer numbers, divided by space character: x_i in sorted (non-decreasing) order.

Examples

standard input	standard output
3 0 2 1	0 1 2
3 800000000 1000000000 900000000	800000000 900000000 1000000000

Note

Your partition implementation should have $O(N)$ complexity. Refer Lecture 2 video for detailed explanation.

Problem C. Ali Baba 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Ali Baba successfully brought treasures you helped him to choose in Ali Baba 1 Problem. Now he returned back to cave with treasures again and this time he took a very big bag to be able to take more treasures.

When he entered the cave, he found a room in it that contains $0 \le N \le 100000$ bunches of golden sand.

For each bunch of sand, Ali Baba estimated it's weight (in grams) and it's cost (he counts cost in dollars, because he loves green color).

His bag can hold not more than $0 \le W \le 10^9$ grams of sand in total.

Also, Ali Baba counts cost of each bunch or it's part to be an integer number of dollars. He do not take cents into account. He doesn't like cents, because they are not green.

Help Ali Baba to take out as many golden sand as he can, and maximize total cost.

Input

First line contains two integer numbers divided by space character: $0 \le N \le 10^5$ and $0 \le W \le 10^9$ — number of bunches of golden sand and carrying capacity of a bag in grams respectively.

Next N lines contain information about bunches of golden sand. Each line contains two integer numbers: $0 \le c_i \le 10^9$ and w_i — cost ($) and weight (kg) of i-th bunch respectively.

Output

Single integer number — maximum total cost in dollars of golden sand that Ali Baba can take out of the cave.

Examples

standard input	standard output

3 2
10 1
5 1
1000 2
	1000

1 100 400 300	133

4 3
10 3
4 1
4 1
4 1
	12

Note

Ali Baba counts cost of each bunch (or it's part) to be an integer number. He uses the following formula: $c = c_i * w // w_i$, where c_i, w_i — cost and weight of i-th bunch, w — weight of gold sand taken from this bunch, c — obtained cost, and $//$ — integer division.

E.g. if he takes 100g from bunch with cost=300$ and weight=200g, the cost of this 100g will be $(300 * 100//200) = 133$.