Training 4 - Divide and Conquer -20211

A. 04. PIE

1 second, 256 megabytes

My birthday is coming up and traditionally I'm serving pie. Not just one pie, no, I have a number N of them, of various tastes and of various sizes. F of my friends are coming to my party and each of them gets a piece of pie. This should be one piece of one pie, not several small pieces since that looks messy. This piece can be one whole pie though. My friends are very annoying and if one of them gets a bigger piece than the others, they start complaining. Therefore all of them should get equally sized (but not necessarily equally shaped) pieces, even if this leads to some pie getting spoiled (which is better than spoiling the party). Of course, I want a piece of pie for myself too, and that piece should also be of the same size. What is the largest possible piece size all of us can get? All the pies are cylindrical in shape and they all have the same height 1, but the radii of the pies can be different.

Input

One line with a positive integer: the number of test cases. Then for each test case:

- One line with two integers N and F with $1 \leq N, F \leq 10000$: the number of pies and the number of friends. 25% of tests has $N, F \leq 10$ and 25% of test has solution ≤ 0.1.
- One line with N integers r_i with $1 \leq r_i \leq 10000$: the radii of the pies.

Output

For each test case, output one line with the largest possible volume V such that me and my friends can all get a pie piece of size V. The answer should be given as a floating point number rounding to 6 digits after the floating point (accept error $\leq 10^{-6}$).

input
1 1 10000 3
output
0.002827

Statement is not available on English language

C. 04. EKO

1 second, 256 megabytes

Lumberjack Mirko needs to chop down M metres of wood. It is an easy job for him since he has a nifty new woodcutting machine that can take down forests like wildfire. However, Mirko is only allowed to cut a single row of trees.

Mirko's machine works as follows: Mirko sets a height parameter H (in metres), and the machine raises a giant sawblade to that height and cuts off all tree parts higher than H (of course, trees not higher than H meters remain intact). Mirko then takes the parts that were cut off. For example, if the tree row contains trees with heights of 20, 15, 10, and 17 metres, and Mirko raises his sawblade to 15 metres, the remaining tree heights after cutting will be 15, 15, 10, and 15 metres, respectively, while Mirko will take 5 metres off the first tree and 2 metres off the fourth tree (7 metres of wood in total).

Mirko is ecologically minded, so he doesn't want to cut off more wood than necessary. That's why he wants to set his sawblade as high as possible. Help Mirko find the maximum integer height of the sawblade that still allows him to cut off at least M metres of wood.

Input

The first line of input contains two space-separated positive integers, N (the number of trees, $1 \leq N \leq 1000000$) and M (Mirko's required wood amount, $1 \leq M \leq 2000000000$).

The second line of input contains N space-separated positive integers less than 1 000 000 000, the heights of each tree (in metres). The sum of all heights will exceed M, thus Mirko will always be able to obtain the required amount of wood.

Output

The first and only line of output must contain the required height setting.

input
4 7 20 15 10 17
output
15

input
5 20 4 42 40 26 46
output
36

D. 04. BOOKS1

1 second, 256 megabytes

Before the invention of book-printing, it was very hard to make a copy of a book. All the contents had to be re-written by hand by so called scribers. The scriber had been given a book and after several months he finished its copy. One of the most famous scribers lived in the 15th century and his name was Xaverius Endricus Remius Ontius Xendrianus (Xerox). Anyway, the work was very annoying and boring. And the only way to speed it up was to hire more scribers.

Once upon a time, there was a theater ensemble that wanted to play famous Antique Tragedies. The scripts of these plays were divided into many books and actors needed more copies of them, of course. So they hired many scribers to make copies of these books. Imagine you have m books (numbered $1, 2, \ldots, m$) that may have different number of pages ($p_1, p_2, \ldots, p_m$) and you want to make one copy of each of them. Your task is to divide these books among k scribes, $k \leq m$. Each book can be assigned to a single scriber only, and every scriber must get a continuous sequence of books. That means, there exists an increasing succession of numbers $0 = b_0 < b_1 < b_2, \ldots < b_{k-1} \leq b_k = m$ such that i-th scriber gets a sequence of books with numbers between $b_{i-1} + 1$ and b_i. The time needed to make a copy of all the books is determined by the scriber who was assigned the most work. Therefore, our goal is to minimize the maximum number of pages assigned to a single scriber. Your task is to find the optimal assignment.

Input

The input consists of N cases (equal to about 200). The first line of the input contains only positive integer N. Then follow the cases. Each case consists of exactly two lines. At the first line, there are two integers m and k, $1 \leq k \leq m \leq 500$. At the second line, there are integers $p_1, p_2, \ldots, p_m$ separated by spaces. All these values are positive and less than 10000000.

Output
For each case, print exactly one line. The line must contain the input succession $p_1, p_2, \ldots, p_m$ divided into exactly k parts such that the maximum sum of a single part should be as small as possible. Use the slash character ('/') to separate the parts. There must be exactly one space character between any two successive numbers and between the number and the slash.

If there is more than one solution, print the one that minimizes the work assigned to the first scriber, then to the second scriber etc. But each scriber must be assigned at least one book.

input
2 9 3 100 200 300 400 500 600 700 800 900 5 4 100 100 100 100 100
output
100 200 300 400 500 / 600 700 / 800 900 100 / 100 / 100 / 100 100

E. 04. AGGRCOW

4.0 s, 256 megabytes

Farmer John has built a new long barn, with N ($2 \leq N \leq 100,000$) stalls. The stalls are located along a straight line at positions $x_1, \ldots, x_N$ ($0 \leq x_i \leq 1,000,000,000$).

His C ($2 \leq C \leq N$) cows don't like this barn layout and become aggressive towards each other once put into a stall. To prevent the cows from hurting each other, FJ wants to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?

Input
t – the number of test cases, then t test cases follows.

- Line 1: Two space-separated integers: N and C
- Lines $2 \ldots N+1$: Line $i+1$ contains an integer stall location, x_i

Output
For each test case output one integer: the largest minimum distance.

input
1 5 3 1 2 8 4 9
output
3

FJ can put his 3 cows in the stalls at positions 1, 4 and 8, resulting in a minimum distance of 3.

G. Lost Numbers

1 second, 256 megabytes

This is an interactive problem. Remember to flush your output while communicating with the testing program. You may use `fflush(stdout)` in C++, `system.out.flush()` in Java, `stdout.flush()` in Python or `flush(output)` in Pascal to flush the output. If you use some other programming language, consult its documentation. You may also refer to the guide on interactive problems: https://codeforces.com/blog/entry/45307.

The jury guessed some array a consisting of 6 integers. There are 6 special numbers — 4, 8, 15, 16, 23, 42 — and each of these numbers occurs in a exactly once (so, a is some permutation of these numbers).

You don't know anything about their order, but you are allowed to ask **up to 4 queries**. In each query, you may choose two indices i and j ($1 \le i, j \le 6$, i and j are not necessarily distinct), and you will get the value of $a_i \cdot a_j$ in return.

Can you guess the array a?

The array a is fixed beforehand in each test, the interaction program doesn't try to adapt to your queries.

Interaction

Before submitting the answer, you may ask up to 4 queries. To ask a query, print one line in the following format: ? i j, where i and j should be two integers such that $1 \le i, j \le 6$. *The line should be ended with a line break character*. After submitting a query, flush the output and read the answer to your query — one line containing one integer $a_i \cdot a_j$. If you submit an incorrect query (or ask more than 4 queries), the answer to it will be one string `0`. After receiving such an answer, your program should terminate immediately — otherwise you may receive verdict "Runtime error", "Time limit exceeded" or some other verdict instead of "Wrong answer".

To give the answer, your program should print one line ! a_1 a_2 a_3 a_4 a_5 a_6 *with a line break in the end*. After that, it should flush the output and terminate gracefully.

input
16 64 345 672
output
? 1 1 ? 2 2 ? 3 5 ? 4 6 ! 4 8 15 16 23 42

If you want to submit a hack for this problem, your test should contain exactly six space-separated integers a_1, a_2, ..., a_6. Each of 6 special numbers should occur exactly once in the test. The test should be ended with a line break character.

H. Vanya and Lanterns

1 second, 256 megabytes

Vanya walks late at night along a straight street of length l, lit by n lanterns. Consider the coordinate system with the beginning of the street corresponding to the point 0, and its end corresponding to the point l. Then the i-th lantern is at the point a_i. The lantern lights all points of the street that are at the distance of at most d from it, where d is some positive number, common for all lanterns.

Vanya wonders: what is the minimum light radius d should the lanterns have to light the whole street?

Input

The first line contains two integers n, l ($1 \le n \le 1000$, $1 \le l \le 10^9$) — the number of lanterns and the length of the street respectively.

The next line contains n integers a_i ($0 \le a_i \le l$). Multiple lanterns can be located at the same point. The lanterns may be located at the ends of the street.

Output

Print the minimum light radius d, needed to light the whole street. The answer will be considered correct if its absolute or relative error doesn't exceed 10^{-9}.

input
7 15 15 5 3 7 9 14 0
output
2.5000000000

input
2 5 2 5
output
2.0000000000

Consider the second sample. At $d = 2$ the first lantern will light the segment $[0, 4]$ of the street, and the second lantern will light segment $[3, 5]$. Thus, the whole street will be lit.

I. 04.TRIPLE

1 second, 512 megabytes

Given a sequence of positive integers $a = (a_1, \ldots, a_N)$ and a positive integer K. Compute the number Q of triple (i, j, k) such that $1 \le i < j < k \le N$ and $a_i + a_j + a_k = K$.

Input

The input consists of following lines:

- Line 1 contains N and K ($1 \le N \le 5000, 1 \le K \le 10^5$).
- Line 2 contains $a_1, \ldots, a_N$ ($1 \le a_i \le 10^5$)

Output

Write the value Q.

input
6 12 2 3 4 1 6 7
output
3

J. 04. CLOPAIR

1 second, 256 megabytes

Cho N điểm trên mặt phẳng, hãy tìm một cặp điểm với khoảng cách euclid nhỏ nhất giữa chúng. Biết rằng không có hai điểm nào trùng nhau và có duy nhất một cặt có khoảng cách nhỏ nhất.

Input

Dòng đầu tiên chứa một số nguyên N ($2 \le N \le 50000$). N dòng tiếp theo mỗi dòng chứa hai số nguyên là tọa độ X và Y của một điểm. Giá trị tuyệt đối của X, Y không vượt quá 10^6.

Output

Ghi ra 3 số abc, trong đó a, b ($a < b$) là các chỉ số của cặp điểm tìm được trong dữ liệu vào (chỉ số bắt đầu từ 0) và c là khoảng cách giữa chúng. Làm tròn c đến 6 chữ số sau dấu phẩy động.

input
5 0 0 0 1 100 45 2 3 9 9
output
0 1 1.000000

input
5 0 0 -4 1 -7 -2 4 5 1 1
output
0 4 1.414214

K. EXPMOD

1 second, 256 megabytes

Given two positive integers a and b. Compute a^b mod $(10^9 + 7)$

Input

One line contains two integers a and b ($1 \leq a, b \leq 18446744073709551614$)

Output

The value a^b mod $(10^9 + 7)$

input
2 3
output
8