UST
Code Sprint 2018

Contest Proper
Problems

Problem Letter	Problem
A	Team
B	Choosing Teams
C	Fox and Two Dots
D	Queue
E	Pangram
F	Vanya and Lanterns
G	Almost Prime
H	Chocolate
I	Boredom
J	Carrot Cakes

A. Team

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day three best friends Petya, Vasya and Tonya decided to form a team and take part in programming contests. Participants are usually offered several problems during programming contests. Long before the start the friends decided that they will implement a problem if at least two of them are sure about the solution. Otherwise, the friends won't write the problem's solution.

This contest offers n problems to the participants. For each problem we know, which friend is sure about the solution. Help the friends find the number of problems for which they will write a solution.

Input

The first input line contains a single integer n $(1 \leq n \leq 1000)$ — the number of problems in the contest. Then n lines contain three integers each, each integer is either 0 or 1. If the first number in the line equals 1, then Petya is sure about the problem's solution, otherwise he isn't sure. The second number shows Vasya's view on the solution, the third number shows Tonya's view. The numbers on the lines are separated by spaces.

Output

Print a single integer -— the number of problems the friends will implement on the contest.

Examples

Input	Output
3 1 1 0 1 1 1 1 0 0	2
2 1 0 0 0 1 1	1

Note

In the first sample Petya and Vasya are sure that they know how to solve the first problem and all three of them know how to solve the second problem. That means that they will write solutions for these problems. Only Petya is sure about the solution for the third problem, but that isn't enough, so the friends won't take it.

In the second sample the friends will only implement the second problem, as Vasya and Tonya are sure about the solution.

Input first
↳ check length
for 0–5
↳ Check Ones

B. Choosing Teams

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Saratov State University Olympiad Programmers Training Center (SSU OPTC) has $n$students. For each student you know the number of times he/she has participated in the ACM ICPC world programming championship. According to the ACM ICPC rules, each person can participate in the world championship at most 5 times.

The head of the SSU OPTC is recently gathering teams to participate in the world championship. Each team must consist of exactly three people, at that, any person cannot be a member of two or more teams. What maximum number of teams can the head make if he wants each team to participate in the world championship with the same members at least $k$times?

Input

The first line contains two integers, n and k $(1 \leq n \leq 2000;\ 1 \leq k \leq 5)$. The next line contains n integers: $y_1, y_2, ..., y_n$ $(0 \leq y_i \leq 5)$, where y_i shows the number of times the i-th person participated in the ACM ICPC world championship.

Output

Print a single number — the answer to the problem.

5-2 = 3

Examples

Input	Output
5 2 0 4 5 1 0	1 0 0 1 4 5
6 4 0 1 2 3 4 5	0
6 5 0 0 0 0 0 0	2

Note

In the first sample only one team could be made: the first, the fourth and the fifth participants.

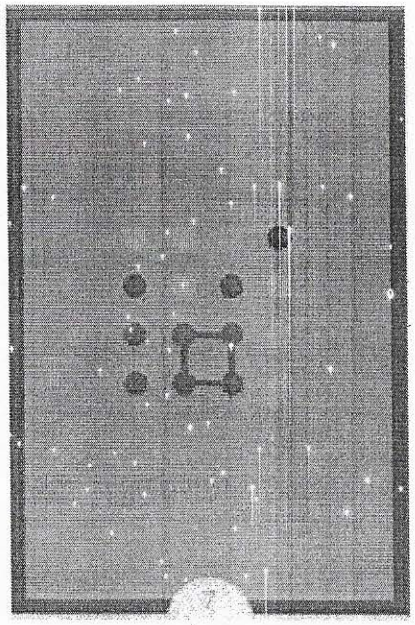
In the second sample no teams could be created.

In the third sample two teams could be created. Any partition into two teams fits.

C. Fox And Two Dots

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel is playing a mobile puzzle game called "Two Dots". The basic levels are played on a board of size $n \times m$ cells, like this:



Each cell contains a dot that has some color. We will use different uppercase Latin characters to express different colors.

The key of this game is to find a cycle that contain dots of same color. Consider 4 blue dots on the picture forming a circle as an example. Formally, we call a sequence of dots $d_1, d_2, ..., d_k$ a *cycle* if and only if it meets the following condition:

1. These k dots are different: if $i \neq j$ then d_i is different from d_j.
2. k is at least 4.
3. All dots belong to the same color.
4. For all $1 \leq i \leq k - 1$: d_i and d_{i+1} are adjacent. Also, d_k and d_1 should also be adjacent. Cells x and y are called adjacent if they share an edge.

Determine if there exists a *cycle* on the field.

Input

The first line contains two integers n and m ($2 \leq n, m \leq 50$): the number of rows and columns of the board.

Then n lines follow, each line contains a string consisting of m characters, expressing colors of dots in each line. Each character is an uppercase Latin letter.

Output

Output "Yes" if there exists a *cycle*, and "No" otherwise.

Examples

Input	Output
3 4 AAAA ABCA AAAA	Yes
3 4 AAAA ABCA AADA	No
4 4 YYYR BYBY BBBY BBBY	Yes
7 6 AAAAAB ABBBAB ABAAAB ABABBB ABAAAB ABBBAB AAAAAB	Yes
2 13 ABCDEFGHIJKLM NOPQRSTUVWXYZ	No

Note

In first sample test all 'A' form a cycle.

In second sample there is no such cycle.

The third sample is displayed on the picture above ('Y' = Yellow, 'B' = Blue, 'R' = Red).

D. Queue

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little girl Susie went shopping with her mom and she wondered how to improve service quality.

There are n people in the queue. For each person we know time t_i needed to serve him. A person will be disappointed if the time he waits is more than the time needed to serve him. The time a person waits is the total time when all the people who stand in the queue in front of him are served. Susie thought that if we swap some people in the queue, then we can decrease the number of people who are disappointed.

Help Susie find out what is the maximum number of not disappointed people can be achieved by swapping people in the queue.

Input

The first line contains integer n ($1 \leq n \leq 10^5$).

The next line contains n integers t_i ($1 \leq t_i \leq 10^9$), separated by spaces.

Output

Print a single number — the maximum number of not disappointed people in the queue.

sum count := 0

Examples

sum=0

Input	Output
5 15 2 1 5 3	4

1 2 3 5 15

0

Note

Value 4 is achieved at such an arrangement, for example: $1, 2, 3, 5, 15$.
Thus, you can make everything feel not disappointed except for the person with time 5.

i=0

i	input	sum	
	1	0	true 1
1	2	1	true 2
2	3	3	true 3
3	5	6	false 3
4	15	11	true 4
5			

E. Pangram

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A word or a sentence in some language is called a *pangram* if all the characters of the alphabet of this language appear in it *at least once*. Pangrams are often used to demonstrate fonts in printing or test the output devices.

You are given a string consisting of lowercase and uppercase Latin letters. Check whether this string is a pangram. We say that the string contains a letter of the Latin alphabet if this letter occurs in the string in uppercase or lowercase.

Input

The first line contains a single integer n $(1 \le n \le 100)$ — the number of characters in the string.

The second line contains the string. The string consists only of uppercase and lowercase Latin letters.

Output

Output "`YES`", if the string is a pangram and "`NO`" otherwise.

Examples

Input	Output
12 toosmallword	NO
35 TheQuickBrownFoxJumpsOverTheLazyDog	YES

F. Vanya and Lanterns

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

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}.

Examples

$\frac{15}{7} = 2.5$

lanterns length

Input	Output
7 15 15 5 3 7 9 14 0	2.5000000000
2 5 2 5	2.0000000000

Note

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.

G. Almost Prime

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A number is called almost prime if it has exactly two distinct prime divisors. For example, numbers 6, 18, 24 are almost prime, while 4, 8, 9, 42 are not. Find the amount of almost prime numbers which are between 1 and n, inclusive.

Input

Input contains one integer number n ($1 \le n \le 3000$).

Output

Output the amount of almost prime numbers between 1 and n, inclusive.

Examples

Input	Output
10	2
21	8

n% n% n% n%
n 7 1 2 3 5 7

2 | ~~14~~ 19
~~14~~ 7
1

uh

10 , 3
7

for i=0 -> 5
5%0=0
5%1=0
5%2=1
5%3=2
5%4=1
5%5=0

H. Chocolate

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bob loves everything sweet. His favorite chocolate bar consists of pieces, each piece may contain a nut. Bob wants to break the bar of chocolate into multiple pieces so that each part would contain **exactly** one nut and any break line goes between two adjacent pieces.

You are asked to calculate the number of ways he can do it. Two ways to break chocolate are considered distinct if one of them contains a break between some two adjacent pieces and the other one doesn't.

Please note, that if Bob doesn't make any breaks, all the bar will form one piece and it still has to have exactly one nut.

Input

The first line of the input contains integer n $(1 \le n \le 100)$ — the number of pieces in the chocolate bar.

The second line contains n integers a_i $(0 \le a_i \le 1)$, where 0 represents a piece without the nut and 1 stands for a piece with the nut.

Output

Print the number of ways to break the chocolate into multiple parts so that each part would contain exactly one nut.

Examples

Input	Output
3 0 1 0	1
5 1 0 1 0 1	4

Note

In the first sample there is exactly one nut, so the number of ways equals 1 — Bob shouldn't make any breaks.
In the second sample you can break the bar in four ways:

```
10|10|1

1|010|1

10|1|01

1|01|01
```

1|1 0|1
1/|0 1
1 1 0 1

1|1 0|1 00|1
1 0|1 0|0 1
1|1|0|00 1
~~1|1|0|1 0 0|1~~

I. Boredom

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alex doesn't like boredom. That's why whenever he gets bored, he comes up with games. One long winter evening he came up with a game and decided to play it.

Given a sequence a consisting of n integers. The player can make several steps. In a single step he can choose an element of the sequence (let's denote it a_k) and delete it, at that all elements equal to $a_k + 1$ and $a_k - 1$ also must be deleted from the sequence. That step brings a_k points to the player.

Alex is a perfectionist, so he decided to get as many points as possible. Help him.

Input

The first line contains integer n ($1 \leq n \leq 10^5$) that shows how many numbers are in Alex's sequence.

The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq 10^5$).

Output

Print a single integer — the maximum number of points that Alex can earn.

Examples

Input	Output
2 1 2	2
3 1 2 3	4
9 1 2 1 3 2 2 2 2 3	10

Note

Consider the third test example. At first step we need to choose any element equal to 2. After that step our sequence looks like this $[2, 2, 2, 2]$. Then we do 4 steps, on each step we choose any element equals to 2. In total we earn 10 points.

J. Carrot Cakes

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In some game by Playrix it takes t minutes for an oven to bake k carrot cakes, all cakes are ready at the same moment t minutes after they started baking. Arkady needs at least n cakes to complete a task, but he currently don't have any. However, he has infinitely many ingredients and one oven. Moreover, Arkady can build one more similar oven to make the process faster, it would take d minutes to build the oven. While the new oven is being built, only old one can bake cakes, after the new oven is built, both ovens bake simultaneously. Arkady can't build more than one oven.

Determine if it is reasonable to build the second oven, i.e. will it decrease the minimum time needed to get n cakes or not. If the time needed with the second oven is the same as with one oven, then it is unreasonable.

Input

The only line contains four integers n, t, k, d $(1 \le n, t, k, d \le 1\,000)$ — the number of cakes needed, the time needed for one oven to bake k cakes, the number of cakes baked at the same time, the time needed to build the second oven.

4 1 1 3 No
5 2 1 3 Yes

if (k >= n) Yes
else if ($k \geq \frac{n}{2}$ & $d \leq t$) ~~No~~ Yes
else if ($k \geq \frac{n}{2}$ & $d \geq t$) No
else if ($k < \frac{n}{2}$ & $d < t$) Yes
else if ($k < \frac{n}{2}$ & $d \geq t$) No

Output

If it is reasonable to build the second oven, print "YES". Otherwise print "NO".

Examples

int batches = $\lceil \frac{n}{k} \rceil$

Input	Output
8 6 4 5	YES
8 6 4 6	NO
10 3 11 4	NO
4 2 1 4	YES

n k

Note

In the first example it is possible to get 8 cakes in 12 minutes using one oven. The second oven can be built in 5 minutes, so after 6 minutes the first oven bakes 4 cakes, the second oven bakes 4 more ovens after 11 minutes. Thus, it is reasonable to build the second oven.

In the second example it doesn't matter whether we build the second oven or not, thus it takes 12 minutes to bake 8 cakes in both cases. Thus, it is unreasonable to build the second oven.

In the third example the first oven bakes 11 cakes in 3 minutes, that is more than needed 10. It is unreasonable to build the second oven, because its building takes more time that baking the needed number of cakes using the only oven.