Problem 1: Fraction Action

Many advanced calculators have a fraction feature that will simplify fractions for you.

You are to write a program that will accept for input a non-negative integer as a numerator and a positive integer as a denominator, and output the fraction in simplest form. That is, the fraction cannot be reduced any further, and the numerator will be less than the denominator. You can assume that all input numerators and denominators will produce valid fractions.

Input	Input	Input	Input
28 7	13 5	0 7	55 10
Output	**Output**	**Output**	**Output**
4	2 3/5	0	5 1/2

Problem 2: Multiple Choice

Your teacher likes to give multiple choice tests. One benefit of giving these tests is that they are easy to mark, given an answer key. The other benefit is that students believe they have a one-in-five chance of getting the correct answer, assuming the multiple choice possibilities are A, B, C, D or E. Write a program that your teacher can use to grade one multiple choice test.

The input will contain the number N ($0 < N < 10000$) followed by $2N$ lines. The $2N$ lines are composed of N lines of student responses (with one of A, B, C, D or E on each line), followed by N lines of correct answers (with one of A, B, C, D or E on each line), in the same order as the student answered the questions (that is, if line i is the student response, then line $N + i$ contains the correct answer to that question).

Output the number of questions the student answered correctly.

Input	Input
3 A B C A C B	3 A A A A B A
Output	**Output**
1	2

Problem 3: Pattern Generator

Write a program that repeatedly reads two numbers n and k and prints all bit patterns of length n with k ones in descending order (when the bit patterns are considered as binary numbers). You may assume that $30 \geq n > 0$, $8 > k \geq 0$, and $n \geq k$. The first number in the input gives the number of pairs n and k. The numbers n and k are separated by a single space. Leading zeroes in a bit pattern should be included. See the example below.

Sample Input	Sample Output
3 2 1 2 0 4 2	The bit patterns are 10 01 The bit patterns are 00 The bit patterns are 1100 1010 1001 0110 0101 0011

Problem 4: Absolutely Acidic

You are gathering readings of acidity level in a very long river in order to determine the health of the river. You have placed N sensors in the river, and each sensor gives an integer reading R. For the purposes of your research, you would like to know the frequency of each reading, and find the absolute difference between the two most frequent readings.

If there are more than two readings that have the highest frequency, the difference computed should be the *largest* such absolute difference between two readings with this frequency. If there is only one reading with the largest frequency, but more than one reading with the second largest frequency, the difference computed should be the *largest* absolute difference between the most frequently occurring reading and any of the readings which occur with second-highest frequency.

The first line of input will be the integer N ($2 \leq N \leq 2 \times 10^6$), the number of sensors. The next N lines each contain the reading for that sensor, which is an integer R ($1 \leq R \leq 1000$). You should assume that there are at least two different readings in the input.

Output the positive integer value representing the absolute difference between the two most frequently occurring readings, subject to the tie-breaking rules outlined above.

Sample Input 1

```
5
1
1
1
4
3
```

Sample Output 1

```
3
```

Sample Input 2

```
4
10
6
1
8
```

Sample Output 2

```
9
```