Funnel

Given two strings of letters, determine whether the second can be made from the first by removing one letter. The remaining letters must stay in the same order.

Examples

funnel("leave", "eave") => true

funnel("reset", "rest") => true

funnel("dragoon", "dragon") => true

**sample input1**

leave eave

**Sample output1**

true

**sample input2**

reset rest

**Sample output2**

true

## 

## Tally Programhttps://www.redditstatic.com/desktop2x/img/renderTimingPixel.png

# Description

5 Friends (let's call them a, b, c, d and e) are playing a game and need to keep track of the scores. Each time someone scores a point, the letter of his name is typed in lowercase. If someone loses a point, the letter of his name is typed in uppercase. Give the resulting score from highest to lowest.

# Input Description

A series of characters indicating who scored a point. Examples:

abcde

# Output Description

The score of every player, sorted from highest to lowest. Examples:

a:1, b:1, c:1, d:1, e:1

**Sample input1**

abcde

**Sample output1**

a:1, b:1, c:1, d:1, e:1

**Sample input2**

dbbaCEDbdAacCEAadcB

**Sample output2**

b:2, d:2, a:1, c:0, e:-2

**Angry Professor(Medium)**

A Discrete Mathematics professor has a class of  students. Frustrated with their lack of discipline, he decides to cancel class if fewer than  students are present when class starts.

Given the arrival time of each student, determine if the class is canceled.

**Input Format**

Each test case consists of two lines. The first line has two space-separated integers,  (students in the class) and  (the cancelation threshold). The second line contains  space-separated integers () describing the arrival times for each student.

**Note:** Non-positive arrival times () indicate the student arrived early or on time; positive arrival times () indicate the student arrived  minutes late.

**Output Format**

For each test case, print the word YES if the class is canceled or NO if it is not.

**Note**

If a student arrives exactly on time(a**i =0)**  the student is considered to have entered before the class started.

**Sample Input1**

4 3

-1 -3 4 2

**Sample Output1**

YES

**Sample Input2**

4 2

0 -1 2 1

**Sample Output2**

NO

**Explanation**

For the first test case, . The professor wants at least  students in attendance, but only  have arrived on time ( and ). Thus, the class is canceled.

For the second test case, . The professor wants at least  students in attendance, and there are  who have arrived on time ( and ). Thus, the class is *not* canceled.

**Angry Children 1(medium)**

Bill Gates is on one of his philanthropic journeys to a village in Utopia. He has N packets of candies and

would like to distribute one packet to each of the K children in the village (each packet may contain

different number of candies). To avoid any fighting among the children, he would like to pick K out of N

packets, such that unfairness is minimized.

Suppose the K packets have (x1, x2, x3,....xk) candies in them, where xi denotes the number of candies in

the ith packet, then we define unfairness as

max(x1,x2,...xk) - min(x1,x2,...xk) where max denotes the highest value amongst the elements, and min

denotes the least value amongst the elements. Can you figure out the minimum unfairness and print it?

**Input Format**

contains an integer N(no of packets) , contains an integer K(no of children), candies in the each packet **.**

All integers are space separated.

**Output Format**

An integer that denotes the minimum possible value of unfairness.

**Sample Input1**

7 3 10 100 300 200 1000 20 30

**Sample Output1**

20

**Explanation**

Here K = 3. We can choose packets that contain 10,20,30 candies. The unfairness is

max(10,20,30) - min(10,20,30) = 30 - 10 = 20

**Sample Input2**

10 4 1 2 3 4 10 20 30 40 100 200

**Sample Output2**

3

Explanation-Here we have to choose the packets that contains 1,2,3,4 candies