The current selected programming language is **Python**. We emphasize the submission of a fully working code over partially correct but efficient code. Use of certain header files is restricted. Once **submitted,** you cannot review this problem again. You can use *print* to debug your code. The *print* may not work in case of syntax/runtime error. The version of Python being used is **2.7.**

A company Dictory is launching a new dictionary application for mobile users. Initially, the dictionary will not have any words. Instead it will be an auto-learning application that will learn according to a user's given text. When a user types text, the application auto-detects the words that appear more than once. The application then stores these words in the dictionary and uses them as suggestions in future typing sessions.

Write an algorithm to identify which words will be saved in the dictionary.

Input

The input consists of a string *textInput,* representing the text that is given as an input to the application by the user.

Output

Print space-separated strings in the lexicographically sorted order representing the number of repeated words detected in the input text and if no word is repeated print "NA".

Note

A word is an alphabetic sequence of characters having no whitespace and there is no punctuation in the input text.

textInput is case-sensitive (i.e. cat and CAT are considered as different words not same).

It contains lowercase and uppercase English alphabets[i.e. a-z, A-Z].

Example

Input:

cat batman latt matter cat matter cat

Output:

cat matter

Explanation:

The word "cat" is repeated three times and the word "matter" is repeated two times in the text.

So, the dictionary will store ["cat", "matter"].

```python
def dictWords(textInput):
    #Write you code here
    WordString = textInput.split(" ")
    s=[]
    for i in WordString:
        if WordString.count(i)>1:
            #This condition will only true if the word is present more than one
            s.append(i)
    #s list will only contain those words which is occured more than once in tex
    s=set(s)

    s = list(s)

    s.sort()
    if len(s)==0:
        #This means there are no duplicate words present in textInput
        return ["NA"]

    return s




def main():
    #input for text input
    textInput =input()

    result = dictWords(textInput)

    print(" ".join([str(res) for res in result]))


if __name__ == "__main__":
    main()
```

agency's computer (P) and sends

transmission, noise in the network causes some bits of the data message P to get distorted. However, we know that Roger always matches the binary values of both messages and checks whether he can convert the message P to message Q by flipping the minimum number of bits.

Write an algorithm to help Roger find the minimum number of bits that must be flipped to convert message P to message Q.

Input
The first line of the input consists of an integer *num1*, representing the secret message sent to the agency's computer (P). The second line consists of an integer *num2*, representing the message sent to the technical head of the agency (Q).

Output
Print an integer representing the minimum number of bits that must be flipped to convert message P to message Q.

Constraints
$-10^4 \leq num1, num2 \leq 10^9$

Example
Input:

7

10

Output:

3

Explanation:
Binary representation of P is 00000111.
Binary representation of Q is 00001010.
Three bits of P at position 5, 6 and 8 must be flipped.
So, the number of bits that must be flipped is 3.

```python
def Filpped(num1,num2):
    #Converting decimal value to binary number
    binaryNum1 = bin(num1)
    binaryNum1 = str(binaryNum1)[2:]

    #Converting decimal value to binary number
    binaryNum2 = bin(num2)
    binaryNum2 = str(binaryNum2)[2:]
    #This will count how many times we have flip
    count=0

    if len(binaryNum2)==len(binaryNum1):
        for i,j in zip(binaryNum1,binaryNum2):
            if i!=j :
                #This means we have to flipped binaryNum1
                count+=1
        return count
    else:
        #if length of binaryNum2 is greater than length of binaryNum2 then do th
        binaryNum1 = "0"*(len(binaryNum2)-len(binaryNum1)) + binaryNum1
        for i,j in zip(binaryNum1,binaryNum2):
            if i!=j :
                #This means we have to flipped binaryNum1
                count+=1
        return count


def main():
    num1 = int(input())
    num2 = int(input())

    result = Filpped(num1,num2)

    print(result)

if __name__ == "__main__":
    main()
```