Take a string `str` as input and print all its **characters** such that each character is printed in the **same line** by giving a **tab** after printing each character.

Input Format

Input contains a string str.

Constraints

```
1 <= str.length <= 10^5
```

Output Format

print the string taken by the user.

Sample Input 0

```
geekster
```

Sample Output 0

```
g	e	e	k	s	t	e	r
```

Submitted Code

Language: Java 15

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

        for(int i =0;i<str.length();i++){
            System.out.print(str.charAt(i)+"\t");
        }
    }
}
```

Lucas was a computer science student who enjoyed coding challenges. One day, he was given a programming problem that required him to take **two input strings** and determine if the **first string** was a **toggle** of the **second string.**

Note : toggling means changing uppercase to lowercase and vice-versa

Input Format

First line contains a string input s1.

Second line contains a string input s2.

Constraints

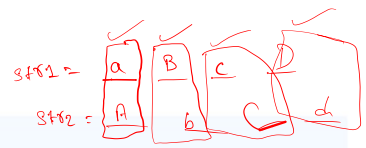
- 1<=str1.length()<=100000
- 1<=str2.length()<=100000

Output Format

Return a boolean value

Sample Input 0

```
aBcD
AbCd
```



Sample Output 0

```
true
```

Explanation 0

At each index of string 1 we are its toggled character in string 2

Submitted Code

Language: Java 15

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str1 = sc.nextLine();
        String str2 = sc.nextLine();

        if(str1.length() !=str2.length()){
            System.out.print(false);
            return;
        }

        for(int i =0;i<str1.length();i++){
            char ch1 = str1.charAt(i);
            char ch2 = str2.charAt(i);
            if(!(ch1==ch2-32 || ch1 == ch2+32)){
                System.out.print(false);return;
            }
        }
        System.out.print(true);
    }
}
```

Given a string. Print the indices of consonants in the string.

Input Format

A String

Constraints

1<=str.length()<=100000

Output Format

An integer series in single line

Sample Input 0

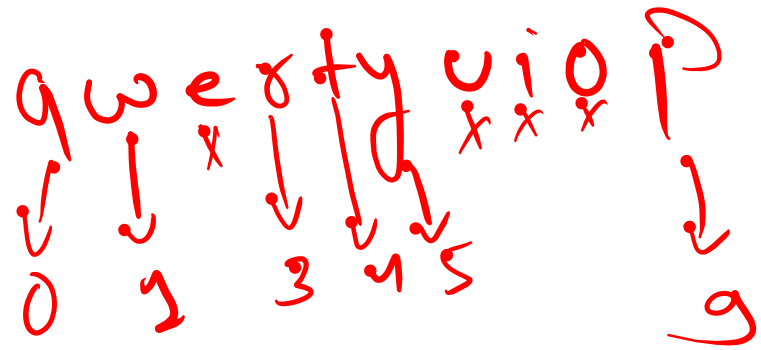
```
qwertyuiop
```

Sample Output 0

```
0 1 3 4 5 9
```

Explanation 0

print the indices of character which are consonants



Submitted Code

Language: Java 15

Open in editor

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

        for(int i =0;i<str.length();i++){
            char ch = str.charAt(i);
            if(ch != 'a' && ch != 'A' && ch != 'e' && ch != 'E' && ch != 'i' && ch != 'I' && ch != 'o' && ch != 'O' && ch != 'u' && ch != 'U')System.out.print(i+" ");
        }
    }
}
```

Given a string. Generate all rotations of a string.

Input Format

A String

Constraints

1<=string.length()<=10^3

Output Format

An series of String in differnt lines.

Sample Input 0

```
geeks
```

Sample Output 0

```
geeks
sgeek
ksgee
eksge
eeksg
```

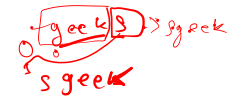
Explanation 0

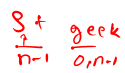
after each right rotation we print the string.

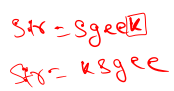
Submitted Code

Language: Java 15

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        int n =str.length();
        int i = n ;
        while(i-->0){
           System.out.println(str);
           str = str.charAt(n-1) + str.substring(0,n-1);
        }
    }
```







Submitted Code

Language: Java 15

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        String str = sc.next();

        int n =str.length();
        int i = n ;

        while(i>0){
            System.out.println(str);
            str = str.charAt(n-1) + str.substring(0,n-1);
            i--;
        }
    }
}
```

Take Two Strings as input. First string as **"str"** and second string as a **"Target"** string.

You are allowed to **rotate** the original string "str" **multiple** times.

Print **"True"** if **"Target"** string can be achieved by rotating the **"str"** any number of times else print **"False".**

Note: String **"bcda"** is a rotation of **"abcd"** but **"bdca"** is not a rotation of String **"abcd"**.

Input Format

- String **"str"**
- String **"Target"**

Constraints

1. 1<=str.length()<=1000
2. 1<=Target.length()<=1000

Output Format

A String **"True"** or **"False"**.

Sample Input 0

```
abcde
cdeab
```

Sample Output 0

```
True
```

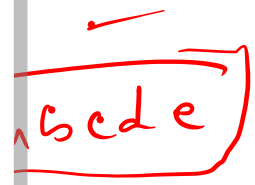
Solved: 2
Attempted: 2

Sample Input 1

```
abcde
abced
```

Sample Output 1

```
False
```



Submitted Code

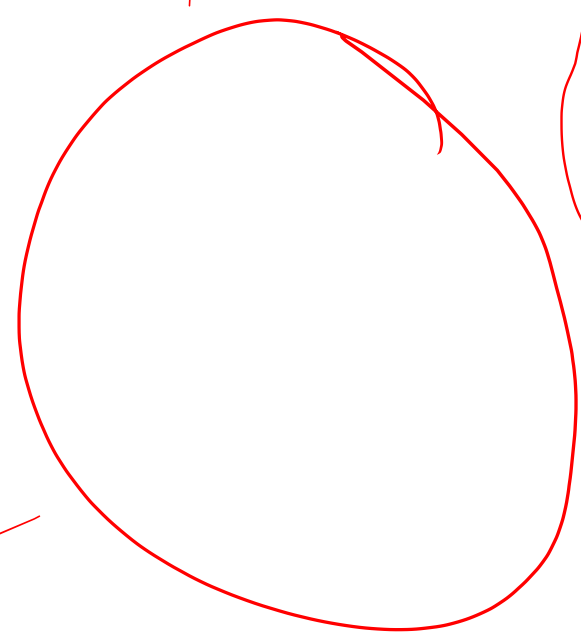
Language: Java 15

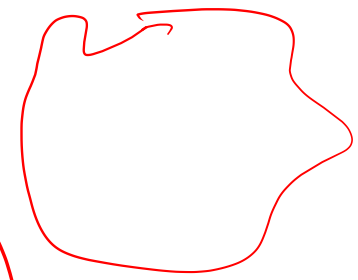
```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        String str = sc.next();
        String target = sc.next();

        if(str.length()!=target.length()){
            System.out.print("False");
            return;
        }

        int n =str.length();
        int i = n ;

        while(i>0){
            if(target.equals(str)){
                System.out.print("True");
                return;
            }
            str = str.charAt(n-1) + str.substring(0,n-1);
            i--;
        }
        System.out.print("False");
    }
}
```

Str = abc

= qbc

Pool





-> Str = abc

-> Mutty = abc

Str = abc + d

Str = abcd

mut

-> String -> Immutable

c *

Str = abc

= abd

-> -> StringBuilder();

- StringBuffer();