

Ex. No. : **6.10**

Date:

Register No: **231501049**

Name: **GNAANESH B B**

Username Domain Extension

Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

Input Format:

The first line contains S.

Output Format:

The first line contains EXTENSION.

The second line contains DOMAIN.

The third line contains USERNAME.

Boundary Condition:

$1 \leq \text{Length of } S \leq 100$

Example Input/Output 1:

Input:

vijayakumar.r@rajalakshmi.edu.in

Output:

edu.in
rajalakshmi
vijayakumar.r

PROGRAM:

```
a = input()  
ext = a.split('@')[0]  
dom = a.split('@')[1].split('.')[0]  
userno = a.find('.')  
user = a[userno+1:]  
print(user)  
print(dom, end='\n')
```

```
print(ext,end='\n')
```

	Input	Expected	Got	
✓	abcd@gmail.com	com gmail abcd	com gmail abcd	✓

Passed all tests! ✓

07 - Functions

Ex. No. : 7.1

Date:

Register No: 231501049

Name: GNAANESH B B

Abundant Number

An abundant number is a number for which the sum of its proper divisors is greater than the number itself. Proper divisors of the number are those that are strictly lesser than the number.

Input Format:

Take input an integer from stdin

Output Format:

Return Yes if given number is Abundant. Otherwise, print No

Example input:

12

Output:

Yes

Explanation

The proper divisors of 12 are: 1, 2, 3, 4, 6, whose sum is $1 + 2 + 3 + 4 + 6 = 16$. Since sum of proper divisors is greater than the given number, 12 is an abundant number.

Example input:

13

Output:

No

Explanation

The proper divisors of 13 is: 1, whose sum is 1. Since sum of proper divisors is not greater than the given number, 13 is not an abundant number.

For example:

Test	Result
print(abundant(12))	Yes
print(abundant(13))	No

PROGRAM:

```
def abundant(n):  
    l,s=[],0  
    for i in range(1,int(n//2)+1):  
        if(n%i==0):  
            l.append(i)  
    for i in l:  
        s+=i  
    if(s>n):  
        return("Yes")  
    else:  
        return("No")
```

	Test	Expected	Got	
✓	print(abundant(12))	Yes	Yes	✓
✓	print(abundant(13))	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : **7.2**

Date:

Register No: **231501049**

Name: **GNAANESH B B**

Automorphic number or not

An automorphic number is a number whose square ends with the number itself. For example, 5 is an automorphic number because $5*5 = 25$. The last digit is 5 which same as the given number.

If the number is not valid, it should display “Invalid input”.

If it is an automorphic number display “Automorphic” else display “Not Automorphic”.

Input Format:

Take a Integer from Stdin

Output Format:

Print Automorphic if given number is Automorphic number, otherwise Not Automorphic

Example input: 5 Output: Automorphic Example input: 25 Output: Automorphic

Example input: 7 Output: Not Automorphic

For example:

Test	Result
print(automorphic(5))	Automorphic

PROGRAM:

```
def automorphic(n):  
    a=str(n*n)  
    if(int(a[-1])==n):  
        return("Automorphic")  
    else:  
        return("Not Automorphic")
```

	Test	Expected	Got	
✓	print(automorphic(5))	Automorphic	Automorphic	✓
✓	print(automorphic(7))	Not Automorphic	Not Automorphic	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 7.3

Date:

Register No: 231501049

Name: GNAANESH B B

Check Product of Digits

Write a code to check whether product of digits at even places is divisible by sum of digits at odd place of a positive integer.

Input Format:

Take an input integer from stdin.

Output Format:

Print TRUE or FALSE.

Example Input:

1256

Output:

TRUE

Example Input:

1595

Output:

FALSE

For example:

Test	Result
print(productDigits(1256))	True
print(productDigits(1595))	False

```
def productDigits(n):
```