1.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

SOLUTION:
```
def abundant(n):
    c=0
    for i in range (1,n):
        if n%i==0:
            c=c+i
    if c>n:
        return 'Yes'
    else:
        return 'No'
```

2.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`

SOLUTION:

```
def automorphic(n):
    d=n*n
    s=len(str(n))
    if d%10==n:
        return 'Automorphic'
    elif d%10!=n:
        return 'Not Automorphic'
    else :
        return 'Invalid input'
```

3.An e-commerce company plans to give their customers a special discount for Christmas. They are planning to offer a flat discount. The discount value is calculated as the sum of all the prime digits in the total bill amount.

Write an algorithm to find the discount value for the given total bill amount.

Constraints

1 <= orderValue< 10e100000

Input

The input consists of an integer orderValue, representing the total bill amount.

Output

Print an integer representing the discount value for the given total bill amount.

Example Input

578

Output

12

For example:

Test	Result
`print(christmasDiscount(57 8))`	12

SOLUTION:
```
def christmasDiscount(n):
    r=0
    c=0
    su=0
    while n>0:
        r=n%10
        c=0
        for i in range (2,r+1):
            if r%i==0:
                c=c+1
        if c==1:
            su=su+r
        n=n//10
    return su
```

4.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(125 6))	True
print(productDigits(159 5))	False

```
SOLUTION:
def productDigits(n):
    d=len(str(n))
    r,i,o=0,0,0
    e=1
    while n>0:
        r=n%10
        if d%2==0:
            e=e*r
        else:
            o=o+r
        d=d-1
```

```
        n=n//10
    if e%o==0:
        return 'True'
    else:
        return 'False'
```

5.complete function to implement coin change making problem i.e. finding the minimum number of coins of certain denominations that add up to given amount of money.

The only available coins are of values 1, 2, 3, 4

Input Format:

Integer input from stdin.

Output Format:

return the minimum number of coins required to meet the given target.

Example Input:

16

Output:

4

Explanation:

We need only 4 coins of value 4 each

Example Input:

25

Output:

7

Explanation:

We need 6 coins of 4 value, and 1 coin of 1 value

Answer:(penalty regime: 0 %)

```
SOLUTION:
def coinChange(n):
    c=0
    if n>=4:
        #print(4)
        c=c+(n//4)
        n=n%4
```

```python
    if n>=3:
        #print(3)
        c=c+(n//3)
        n=n%3
    if n>=2:
        #print(2)
        c=c+(n//2)
        n=n%2
    if n>=1:
        #print(1)
        c=c+(n//1)
        n=n%1
return c
```