

## **07 - Functions**



**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



**Ex. No.** : 7.1

**Date:**

**Register No.:** 230701101

**Name:** Harini M

---

### **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

**Solution:**

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



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





Ex. No. : 7.2

Date:

Register No.: 230701101

Name: Harini M

---

### **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".

### **Solution:**

```
def automorphic(n):    if not
isinstance(n,int) or n<0:
return "Invalid input"    s=n*n
string=str(n)    square=str(s)
if square.endswith(string):
return "Automorphic"    else:
return "Not Automorphic"
```



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



Ex. No. : 7.3

Date:

Register No.: 230701101

Name: Harini M

---

### **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.

**Solution:**

```
def productDigits(n):
    odd_sum = 0
    even_product = 1

    n = str(n)

    for i in range(len(n)):
        digit = int(n[i])

        if i % 2 == 0:
            odd_sum += digit
        else:
            even_product *= digit

    if odd_sum == 0:
        return False
    return even_product % odd_sum == 0
```



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(578))	12





Ex. No. : 7.4

Date:

Register No.: 230701101

Name: Harini M

---

## Christmas Discount

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 python code to find the discount value for the given total bill amount.

### Constraints

$1 \leq \text{orderValue} < 10e^{100000}$

### Solution:

```
def christmasDiscount(n):
    sum=0
    while(n>0):
        rem=n%10
        n=n//10
        num=rem
        if num > 1:
            flag=0
            for i in range(2,
                (num//2)+1):
                if (num % i)
                    == 0:
                    flag=1
            if flag==0:
                sum+=num
    return sum
```



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



Ex. No. : 7.5

Date:

Register No.: 230701101

Name: Harini M

---

### Coin Change

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

Solution:

```
def
coinChange(n):
c=0  i=0  while
c<n:
    c+=4
i+=1  return i
```



Input Format:

Take a number in the form of String from stdin. Output

Format:

Print the difference between sum of even and odd digits Example

input:

1453

Output:

1

Explanation:

Here, sum of even digits is  $4 + 3 = 7$  sum of odd  
digits is  $1 + 5 = 6$ .

Difference is 1.

Note that we are always taking absolute difference





**Ex. No.** : 7.6

**Date:**

**Register No.:** 230701101

**Name:** Harini M

---

### **Difference Sum**

Given a number with maximum of 100 digits as input, find the difference between the sum of odd and even position digits.

**Solution:**

```
def differenceSum(n):
    s=str(n)  e=0  o=0
    for i in range(len(s)):
        d=int(s[i])
        if (i+1)%2==0:
            e+=d
        else:
            o+=d
    diff=abs(e-o)  return
    diff
```



**For example:**

Test	Result
print(checkUgly(6))	ugly
print(checkUgly(21))	not ugly



Ex. No. : 7.7

Date:

Register No.: 230701101

Name: Harini M

---

### Ugly number

A number is considered to be ugly if its only prime factors are 2, 3 or 5. [1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ...] is the sequence of ugly numbers.

Task:

complete the function which takes a number n as input and checks if it's an ugly number.  
return ugly if it is ugly, else return not ugly

Hint:

An ugly number U can be expressed as:  $U = 2^a * 3^b * 5^c$ , where a, b and c are nonnegative integers.

Solution:

```
def checkUgly(n):
    if n<0: return
    "not ugly" for i in
    [2,3,5]: while
    n%i==0:
        n//=i if
    n==1: return
    "ugly" else:
        return "not ugly"
```





