

Ex. No. : 7.1

Date:

Register No.:

Name:

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

```
def is_abundant(number):  
    divisors_sum = sum(i for i in range(1, number) if number % i == 0)  
    return divisors_sum > number
```

Ex. No. : 7.2

Date:

Register No.:

Name:

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 $5 \times 5 = 25$. 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".

def is_automorphic(number):

square = number ** 2

return str(square).endswith(str(number))

Ex. No. : 7.3

Date:

Register No.:

Name:

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.

```
def productDigits(n):  
    sum_odd = 0  
    product_even = 1  
  
    digits = str(n)  
  
    for i in range(len(digits)):  
        digit = int(digits[i])  
        if (i + 1) % 2 == 0:  
            product_even *= digit  
        else:  
            sum_odd += digit  
  
    if sum_odd == 0:  
        return "FALSE"  
  
    return "True" if product_even % sum_odd == 0 else "False"
```

Ex. No. : 7.4

Date:

Register No.:

Name:

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 <= orderValue < 10e100000

```
def christmasDiscount(n):  
    s=[]  
    while n>0:  
        c=int(n%10)  
        flag=1  
        for i in range(2,c):  
            if c%i==0:  
                flag=0  
                break  
        if flag==1:  
            s.append(c)  
        n/=10  
    return sum(s)
```

Ex. No. : 7.5

Date:

Register No.:

Name:

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

```
def coinChange(n):  
    coins = [1, 2, 3, 4]  
    dp=[float('inf')]*(n+1)  
    dp[0]=0  
    for i in range(1,n+1):  
        for j in coins:  
            if i-j>=0:  
                dp[i]=min(dp[i],dp[i-j]+1)  
    return dp[n]
```

Ex. No. : 7.6

Date:

Register No.:

Name:

Difference Sum

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

```
def differenceSum(n):  
    sum_even = 0  
    sum_odd = 0  
  
    digits = str(n)  
  
    for i in range(len(digits)):  
        digit = int(digits[i])  
        if (i + 1) % 2 == 0:  
            sum_even += digit  
        else:  
            sum_odd += digit  
  
    difference = abs(sum_even - sum_odd)  
    return difference
```

Ex. No. : 7.7

Date:

Register No.:

Name:

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.

```
def checkUgly(n):
```

```
    if n <= 0:
```

```
        return "not ugly"
```

```
    while n % 2 == 0:
```

```
        n /= 2
```

```
    while n % 3 == 0:
```

```
        n /= 3
```

```
    while n % 5 == 0:
```

```
        n /= 5
```

```
    if n == 1:
```

```
        return "ugly"
```

```
    else:
```

```
        return "not ugly"
```