**WEEK-9 CODING**

1. 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.

**PROGRAM:**

def checkUgly(n):

if n <= 0:

return "not ugly"

while n % 2 == 0:

n //= 2

while n % 3 == 0:

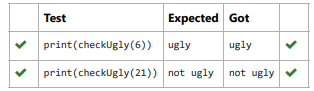
n //= 3

while n % 5 == 0:

n //= 5

return "ugly" if n == 1 else "not ugly"

**OUTPUT:**

****

2. 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.

**PROGRAM:**

def abundant(number):

s=sum([divisor for divisor in range(1,number) if number % divisor == 0])

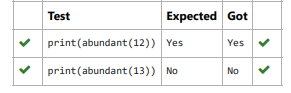
if d\_s>number:

return"Yes"

else:

return "No"

**OUTPUT:**



3.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

**PROGRAM:**

def productDigits(n):

n\_str = str(n)

sum\_odd = 0

product\_even = 1

has\_even\_digits = False

for i in range(len(n\_str)):

digit = int(n\_str[i])

if (i + 1) % 2 == 0:

product\_even \*= digit

has\_even\_digits = True

else:

sum\_odd += digit

if not has\_even\_digits:

return False

return product\_even % sum\_odd == 0

if \_\_name\_\_ == "\_\_main\_\_":

import sys

input = sys.stdin.read

try:

n = int(input().strip())

if productDigits(n):

print("TRUE")

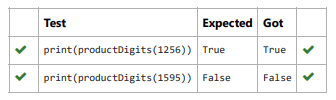
else:

print("FALSE")

except (EOFError, ValueError):

# No need to print anything if input is not provided or invalid

pass

**OUTPUT:**

4. 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

**PROGRAM:**

def automorphic(n):

if(n<0):

return "Invalid input"

square = n \* n

n\_s=str(n)

s\_s=str(square)

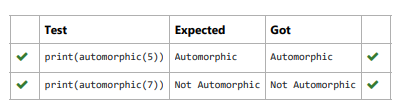
if s\_s.endswith(n\_s):

return "Automorphic"

else:

return "Not Automorphic"

**OUTPUT:**

****

5. Given a number with maximum of 100 digits as input, find the difference between the sum

of odd and even position digits.

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

**PROGRAM:**

def differenceSum(number\_str):

number\_str=str(number\_str)

odd\_sum = 0

even\_sum = 0

for index, char in enumerate(number\_str):

digit = int(char)

if (index + 1) % 2 == 0:

even\_sum += digit

else:

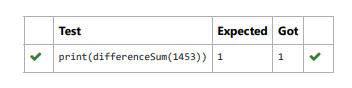
odd\_sum += digit

difference = abs(even\_sum - odd\_sum)

return difference

print(differenceSum(number\_str))

**OUTPUT:**

****