09 – Functions

Ex. No.: 9.1 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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:

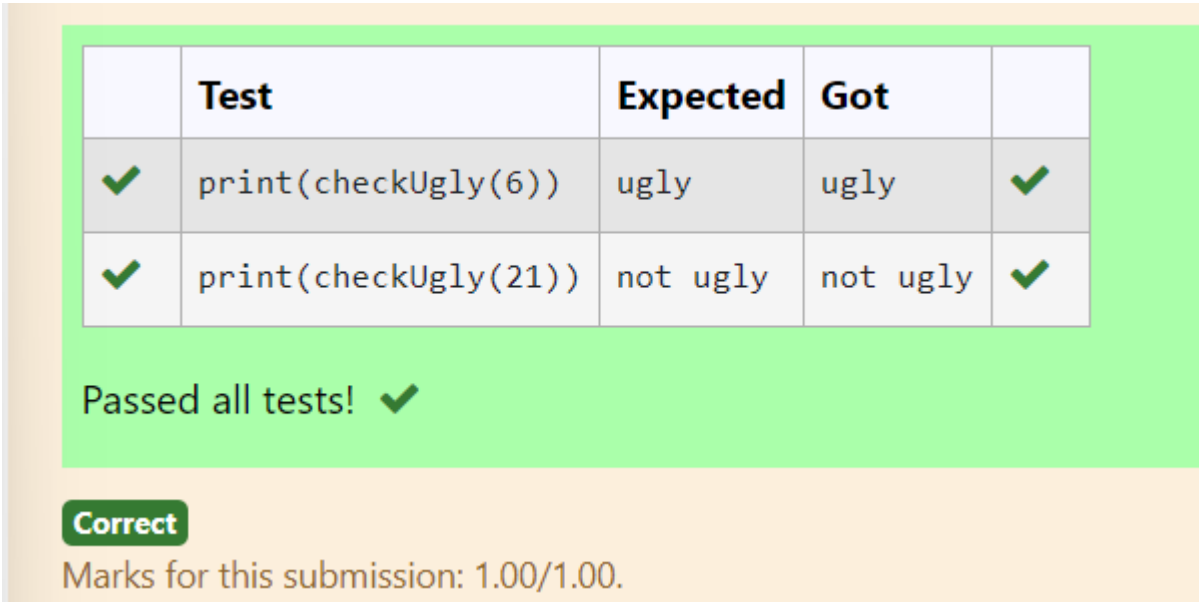
An ugly number U can be expressed as: U = 2^a * 3^b * 5^c, where a, b and c are nonnegative integers.

For example:

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

```
def checkUgly(n):
    f = 0
    if((n%2 == 0 and n%3 == 0)or(n%2 == 0 and n%5 == 0)):
        f = 1
    elif((n%3 == 0 and n%5 == 0)):
        f = 1
    elif(n%5==0):
        f = 1
    elif(n%2 == 0 and n%3 == 0 and n%5 == 0):
        f=1
    if(f == 1):
        return "ugly"
    else:
        return "not ugly"
```

Output:



	Test	Expected	Got	
✔	`print(checkUgly(6))`	`ugly`	`ugly`	✔
✔	`print(checkUgly(21))`	`not ugly`	`not ugly`	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 9.2 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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

```
def coinChange(n):
    c = 0
    while(n>=4):
        c = c + 1
        n = n - 4
    if(n != 0):
        c = c + 1
    return c
```

Output:

	Test	Expected	Got	
✔	print(coinChange(16))	4	4	✔

Passed all tests! ✔

Ex. No.: 9.3 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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

Program:

```
def differenceSum(n):
    n = str(n)
    ev = 0
    od = 0
    for i in range(len(n)):
        if i%2==0:
            ev += int(n[i])
        else:
            od += int(n[i])
    return abs(ev-od)
```

Output:

	Test	Expected	Got	
✔	print(differenceSum(1453))	1	1	✔

Passed all tests! ✔

Ex. No.: 9.4 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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

```
def productDigits(n):
    n = str(n)
    ev = 1
    od = 0
    for i in range(len(n)):
        if((i+1) % 2 == 0):
            ev = ev * int(n[i])
        else:
            od = od + int(n[i])
    if(ev%od == 0):
        return True
    else:
        return False
```

Output:

	Test	Expected	Got	
✔	print(productDigits(1256))	True	True	✔
✔	print(productDigits(1595))	False	False	✔

Passed all tests! ✔

Correct

Ex. No.: 9.5 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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

Program:

```
def abundant(n):
    s = 0
    for i in range(1, 11):
        if(n%i == 0):
            s = s + i
    if s>n:
        return "Yes"
    else:
        return "No"
```

Output:

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

Passed all tests! ✔

Correct