Started on	Thursday, 19 September 2024, 11:07 AM
State	Finished
Completed on	Thursday, 19 September 2024, 11:43 AM
Time taken	35 mins 55 secs
Marks	5.00/5.00
Grade	**100.00** out of 100.00

Question **1**

Correct

Mark 1.00 out of 1.00

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(578))`	12

Answer: (penalty regime: 0 %)

Reset answer

```
def christmasDiscount(n):
    d=0
    p=[2,3,5,7]
    for dig in str(n):
        dig=int(dig)
        if primeno(dig):
            d+=dig
    return d
def primeno(dig):
    return dig in[2,3,5,7]

```

	Test	Expected	Got	
✔	`print(christmasDiscount(578))`	12	12	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

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 %)

Reset answer

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

```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Write a function that returns the value of a+aa+aaa+aaaa with a given digit as the value of a.

Suppose the following input is supplied to the program:

9

Then, the output should be:

9+99+999+9999=11106

Sample Input Format:

9

Sample Output format:

11106

For example:

Test	Result
`print(Summation(8))`	`9872`

Answer: (penalty regime: 0 %)

Reset answer

```
def Summation(n):
    s=0
    j=n
    for i in range(1,5):
        if(n>=10):
            s+=j
            j=(j*100)+n
        elif(n<10):
            s+=j
            j=(j*10)+n
    return s
```

	Test	Expected	Got	
✔	`print(Summation(8))`	`9872`	`9872`	✔
✔	`print(Summation(10))`	`10203040`	`10203040`	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

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`

Answer: (penalty regime: 0 %)

Reset answer

```
def checkUgly(n):
    if n<=0:
        return "not ugly"
    while n%2==0:
        n=n//2
    while n%3==0:
        n=n//3
    while n%5==0:
        n=n//5
    if n==1:
        return "ugly"
    else:
        return "not ugly"
    n=6
    print(checkUgly(n))

```

	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.

Question **5**

Correct

Mark 1.00 out of 1.00

A strobogrammatic number is a number that looks the same when rotated 180 degrees (looked at upside down).

Write a program to determine if a number is strobogrammatic. The number is represented as a string.

Example 1:

Input:

69

Output:

true

Example 2:

Input:

88

Output:

true

Example 3:

Input:

962

Output:

false

Example 4:

Input:

1

Output:

true

For example:

Test	Result
`print(Strobogrammatic(69))`	`true`
`print(Strobogrammatic(962))`	`false`

Answer: (penalty regime: 0 %)

Reset answer

```
def Strobogrammatic(n):
    pair={'0':'0','1':'1','6':'9','9':'6','8':'8'}
    rotate=''
    for digit in reversed(str(n)):
        if digit not in pair:
            return "false"
        rotate+=pair[digit]
    return "true"

```

	Test	Expected	Got	
✔	`print(Strobogrammatic(69))`	`true`	`true`	✔
✔	`print(Strobogrammatic(88))`	`true`	`true`	✔
✔	`print(Strobogrammatic(962))`	`false`	`false`	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.