

Started on	Wednesday, 19 June 2024, 2:24 PM
State	Finished
Completed on	Wednesday, 19 June 2024, 8:23 PM
Time taken	5 hours 59 mins
Marks	3.00/5.00
Grade	60.00 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

```
1 def is_prime_digit(digit):
2     #Check if the digit is a prime number (2, 3, 5, or 7)
3     return digit in {'2', '3', '5', '7'}
4 def christmasDiscount(n):
5     # Convert the total bill amount to a string
6     orderValue_str=str(n)
7
8     discount=0
9
10    #Iterate through each digit of the total bill amount
11
12    for digit in orderValue_str:
13        #Check if the digit is a prime number
14        if is_prime_digit(digit):
15            discount+=int(digit)
16
17    return discount
```

	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

```
1 def coinChange(target):
2     coins = [1, 2, 3, 4]
3     dp = [float('inf')] * (target + 1)
4     dp[0] = 0
5     for i in range(1, target + 1):
6         for coin in coins:
7             if coin <= i:
8                 dp[i] = min(dp[i], dp[i - coin] + 1)
9     return dp[target]
```

	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

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.

For example:

Test	Result
print(abundant(12))	Yes
print(abundant(13))	No

Answer: (penalty regime: 0 %)

Reset answer

```
1 def abundant(number):
2     divisor_sum = sum([divisor for divisor in range(1, number)
3     if divisor_sum > number:
4         return "Yes"
5     else:
6         return "No"
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**
Incorrect
Mark 0.00 out of 1.00

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

Answer: (penalty regime: 0 %)

Reset answer

```
1 def difference_between_odd_even_positions(num_str):
2     odd_sum = 0
3     even_sum = 0
4
5     # Traverse through the digits of the number
6     for i in range(len(num_str)):
7         digit = int(num_str[i])
8         if (i + 1) % 2 == 0:
9             even_sum += digit
10        else:
11            odd_sum += digit
12
13        # Calculate the absolute difference
14        difference = abs(even_sum - odd_sum)
15
16        return difference
17
18 num_str = input().split()
19
20 # Calculate and print the difference
21 result = difference_between_odd_even_positions(num_str)
22 print(result)
```

	Test	Expected	Got	
✖	print(differenceSum(1453))	1	***Run error*** Traceback (most recent call last): File "__tester__.python3", line 18, in <module> num_str = input().split() ^^^^^^^ EOFError: EOF when reading a line	✖

Testing was aborted due to error.
Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.

Question **5**
Incorrect
Mark 0.00 out of 1.00

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

For example:

Test	Result
print(productDigits(1256))	True
print(productDigits(1595))	False

Answer: (penalty regime: 0 %)

Reset answer

```
1 def check_divisibility(num):
2     num_str = str(num)
3     odd_sum = 0
4     even_product = 1
5
6     for i, digit in enumerate(num_str):
7         digit = int(digit)
8         if (i + 1) % 2 == 0:
9             even_product *= digit
10        else:
11            odd_sum += digit
12
13    if odd_sum == 0:
14        return False
15
16    return even_product % odd_sum == 0
17
18 # Read input from studio
19 num_str = sys.stdin.read()
20
21 # Print the result
22 print(check_divisibility(num))
```

	Test	Expected	Got	
✖	print(productDigits(1256))	True	***Run error*** Traceback (most recent call last): File "__tester__.python3", line 19, in <module> num_str = sys.stdin.read() ^^^ NameError: name 'sys' is not defined	✖

Testing was aborted due to error.
Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect
Marks for this submission: 0.00/1.00.