Started on	Sunday, 2 June 2024, 7:51 AM
State	Finished
Completed on	Sunday, 2 June 2024, 10:06 AM
Time taken	2 hours 15 mins
Marks	5.00/5.00
Grade	**100.00** out of 100.00

Question **1**

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=0
    while n>0:
        max_val=max([coin for coin in coins if coin<=n])
        n-=max_val
        count+=1
    return count

n=16
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 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

```
from math import log10
def productDigits(n):
    digit_count = int(log10(n))+1
    total = 0
    prod = 1
    while n > 0:
        if digit_count % 2 == 0 :
            prod *= n % 10
        else:
            total += n % 10

        n = n // 10
        digit_count -= 1

    if prod % total == 0:
        return True
    else:
        return False
n = 1256


```

	Test	Expected	Got	
✓	`print(productDigits(1256))`	`True`	`True`	✓
✓	`print(productDigits(1595))`	`False`	`False`	✓

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

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

number=12

```

	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**

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):
    x=str(n)
    tot=0
    for num in x:
        y=int(num)
        if y==2 or y==3 or y==5 or y==7:
            tot+=y
    return tot

christmasDiscount(578)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **5**

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"
    for prime in [2,3,5]:
        while n%prime==0:
            n//=prime
    return "ugly" if n==1 else "not ugly"

n=6
```

	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.

◄ Week9_MCQ

Jump to...

Searching ►