| | |
|---|---|
| **Started on** | Wednesday, 29 May 2024, 7:35 PM |
| **State** | Finished |
| **Completed on** | Saturday, 1 June 2024, 2:43 PM |
| **Time taken** | 2 days 19 hours |
| **Marks** | 5.00/5.00 |
| **Grade** | **100.00** out of 100.00 |

Question **1**

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 /= 2
    while n % 3 == 0:
        n /= 3
    while n % 5 == 0:
        n /= 5
    return "ugly" if n == 1 else "not ugly"


```

| | 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 **2**

Correct

Mark 1.00 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
2
3  def differenceSum(n):
4      sum1=0
5      sum2=0
6      x=str(n)
7      for i in range(len(x)):
8          if i%2==0:
9              sum1+=int(x[i])
10         else:
11             sum2+=int(x[i])
12     d=abs(sum1-sum2)
13     return d
14
15
16
17
18
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✔ | `print(differenceSum(1453))` | 1 | 1 | ✔ |

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

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

**For example:**

| Test | Result |
|---|---|
| `print(automorphic(5))` | `Automorphic` |

**Answer:** (penalty regime: 0 %)

Reset answer

```
def automorphic(n):
    a=n*n
    if a%10==n:
        return "Automorphic"
    else:
        return "Not Automorphic"
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✔ | `print(automorphic(5))` | `Automorphic` | `Automorphic` | ✔ |
| ✔ | `print(automorphic(7))` | `Not Automorphic` | `Not Automorphic` | ✔ |

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):
    n=str(n)
    count=0
    for i in range(len(n)):
        for j in range(i,len(n)):
            if j%2==1:
                count+=int(n[i])
    return count
```

| | 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

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):
    dp = [float('inf')] * (n + 1)
    dp[0] = 0
    coins = [1, 2, 3, 4]
    for coin in coins:
        for i in range(coin, n + 1):
            dp[i] = min(dp[i], dp[i - coin] + 1)

    return dp[n]
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ Week9_MCQ

Jump to...

Week10_MCQ ►