GE19211 / GE23233 / GE23231 - PSPP/PUP

Dashboard / My courses / PSPP/PUP / Functions: Built-in functions, User-defined functions, Recursive functions / Week9_Coding

Quiz navigation

1 ✓ 2 ✓ 3 ✓ 4 ✓ 5 ✓

Show one page at a time

Finish review

Started on	Friday, 24 May 2024, 8:59 AM
State	Finished
Completed on	Friday, 24 May 2024, 9:02 AM
Time taken	2 mins 19 secs
Marks	5.00/5.00
Grade	**100.00** out of 100.00

Question **1**
Correct
Mark 1.00 out of 1.00
Flag question

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
 return "Automorphic" if str(A).endswith(str(n)) else "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 **2**
Correct
Mark 1.00 out of 1.00
Flag question

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):
    A = {'2', '3', '5', '7'}
    return sum(int(digit) for digit in str(n) if digit in A)
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**
Correct
Mark 1.00 out of 1.00
Flag question

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]
    dp = [0] + [float('inf')] * n
    for amount in range(1, n + 1):
        dp[amount] = min(dp[amount - coin] + 1 for coin in coins if amount >= coin)
    return dp[n]
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**
Correct
Mark 1.00 out of 1.00
Flag question

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(n):
    A = sum(i for i in range(1, n) if n % i == 0)
    return "Yes" if A > n else "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 **5**
Correct
Mark 1.00 out of 1.00
Flag question

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.

Finish review

◄ Week9_MCQ

Jump to...

Searching ►