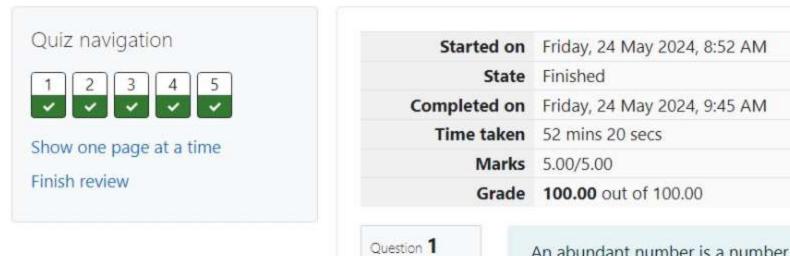
DIVYADHARSHINI V 2022-BIOMED-A D2 ~ REC-PS

## GE19211 / GE23233 / GE23231 - PSPP/PUP

Dashboard / My courses / PSPP/PUP / Functions: Built-in functions, User-defined functions, Recursive functions / Week9\_Coding



An abundant number is a number for which the sum of its proper divisors is greater than Correct the number itself. Proper divisors of the number are those that are strictly lesser than the number. Mark 1.00 out of Input Format: 1.00 Take input an integer from stdin Flag question 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(n): if n < 1: return "No" 3 4 proper\_divisors\_sum = sum(divisor for divisor in range(1, n) if n % divisor == 0) 5 6 return "Yes" if proper\_divisors\_sum > n else "No" 7

**Expected Got** 

return "Not Automorphic"

print(automorphic(5)) Automorphic

Expected

print(automorphic(7)) Not Automorphic Not Automorphic ✓

Got

Automorphic

17 18

Test

Passed all tests! <

Question 3

Mark 1.00 out of

Flag question

Question 4

Mark 1.00 out of

P Flag question

Output Format:

Example Input:

Test

Passed all tests! <

Correct

18

19

20 21

22 23

24 25

26 27

28 29

Correct

Test

Passed all tests! <

Marks for this submission: 1.00/1.00.

# Iterate over each digit in the order value

# Convert the digit back to an integer

# Add the prime digit to the discount

**Expected Got** 

Jump to ...

12

\$

Finish review

Searching -

# Check if the digit is prime

for digit in order\_str:

num = int(digit)

if is\_prime(num):

return discount

print(christmasDiscount(578)) 12

discount += num

Question 5

Mark 1.00 out of

Flag question

Correct

1.00

print(coinChange(16)) 4

16

Output:

return the minimum number of coins required to meet the given target.

Expected Got

Correct

1.00

Correct

Marks for this submission: 1.00/1.00.

Yes

No

Test

Passed all tests! <

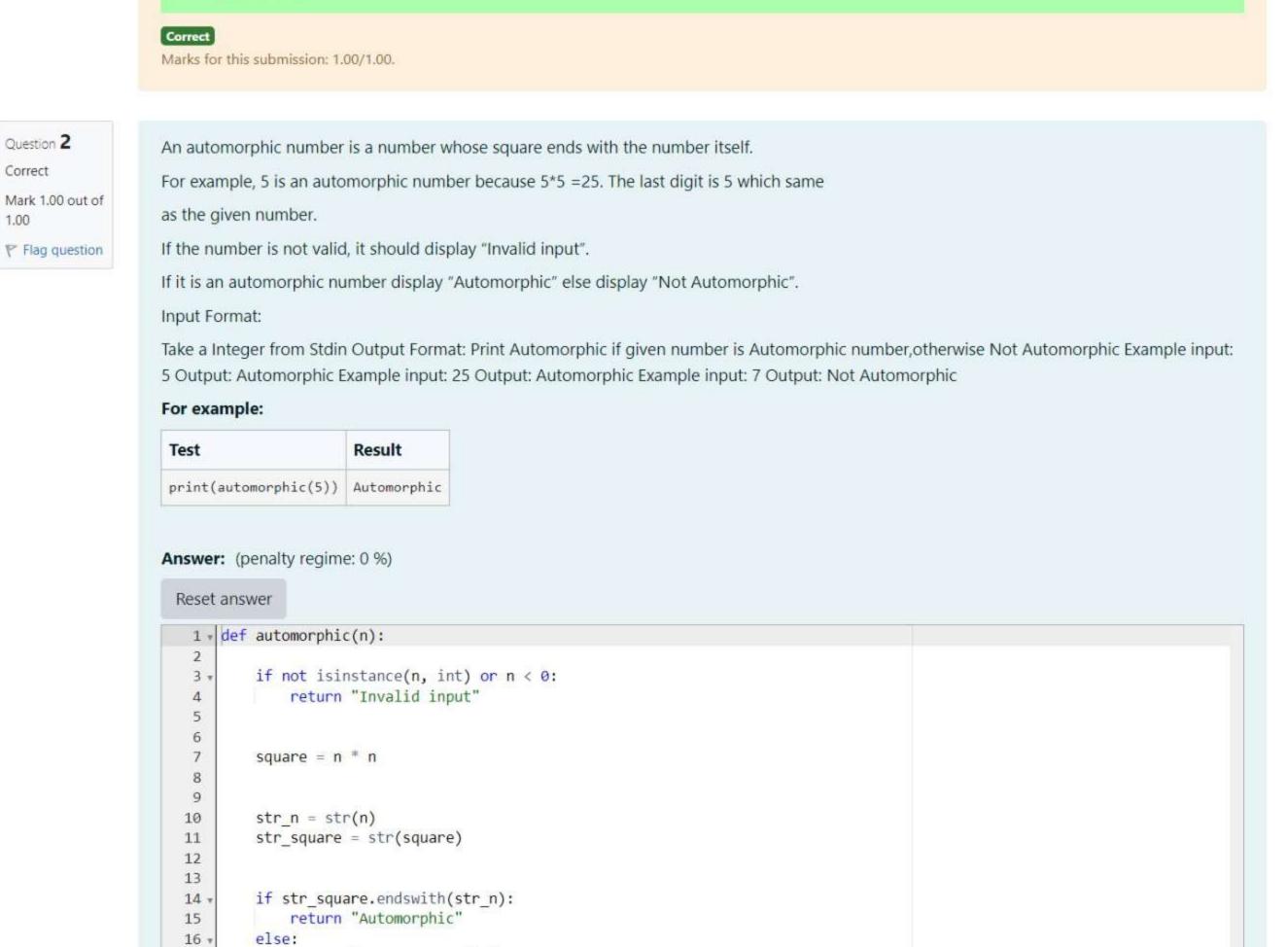
Question 2

Correct

1.00

print(abundant(12)) Yes

print(abundant(13)) No



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 1 def checkUgly(n): if n <= 0: 2 1 return "not ugly" 3 4 5 1 for factor in [2, 3, 5]: while n % factor == 0: 6 1 n //= factor 8 return "ugly" if n == 1 else "not ugly" 9

**Expected Got** Test print(checkUgly(6)) ugly ugly print(checkUgly(21)) not ugly ont ugly Passed all tests! < Correct Marks for this submission: 1.00/1.00. 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.

Explanation: We need only 4 coins of value 4 each Example Input: 25 Output: Explanation: We need 6 coins of 4 value, and 1 coin of 1 value Answer: (penalty regime: 0 %) Reset answer 1 - def coinChange(target): # Available coin denominations 3 coins = [1, 2, 3, 4]4 # Create a list to store the minimum coins needed for each amount up to the target 5 dp = [float('inf')] \* (target + 1) # Base case: 0 coins are needed to make the amount 0 9 dp[0] = 010 # Fill the dp array 11 for i in range(1, target + 1): 12 1 13 , for coin in coins: if i - coin >= 0: 14 1 dp[i] = min(dp[i], dp[i - coin] + 1)15 16 # The answer will be in dp[target] 17 18 return dp[target]

Marks for this submission: 1.00/1.00. 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 v def is\_prime(n): # Check if n is less than 2 if n < 2: 3 + return False 4 5 # Check if n is divisible by any number from 2 to sqrt(n) for i in range(2, int(n\*\*0.5) + 1): if n % i == 0: 8 return False 9 return True 10 11 + def christmasDiscount(orderValue): # Convert the order value to a string to iterate over its digits 12 13 order\_str = str(orderValue) 14 # Initialize discount value 15 16 discount = 0 17

■ Week9\_MCQ You are logged in as DIVYADHARSHINI V 2022-BIOMED-A (Log out)

PSPP/PUP Data retention summary