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

Started on	Monday, 10 June 2024, 9:59 PM
State	Finished
Completed on	Monday, 10 June 2024, 11:51 PM
Time taken	1 hour 52 mins
Marks	5.00/5.00
Grade	**100.00** out of 100.00

Question **1**

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

```python
1  def differenceSum(n):
2      s=str(n)
3      e=0;o=0
4      for i in range(len(s)):
5          if(i%2!=0):
6              e+=int(s[i])
7          else:
8              o+=int(s[i])
9          return abs(e-o)
```

	Test	Expected	Got	
✓	print(differenceSum(1453))	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

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

```python
def checkUgly(n):
    while(n%2==0):
        n//=2
    while(n%3==0):
        n//=3
    while(n%5==0):
        n//=5
    if (n==1):
     return "ugly"
    else:
     return "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 **3**

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

```python
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 **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

```
1  def christmasDiscount(n):
2      def is_prime(digit):
3          if digit in {2,3,5,7}:
4              return True
5          return False
6      str_n=str(n)
7      discount=sum(int(digit) for digit in str_n if is_prime(int(digit)))
8      return discount
9
```

	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

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]

```python
def abundant(number):
    divisor_sum = sum([divisor for divisor in range(1,number) if number % divisor ==0])
    if divisor_sum > number:
        return "Yes"
    else:
        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.

◄ Week9_MCQ

Jump to...

Searching ►