

[Dashboard](#) / [My courses](#) / [PSP/PUP](#) / [Functions: Built-in functions, User-defined functions, Recursive functions](#) / [Week9 Coding](#)

Started on	Friday, 24 May 2024, 3:24 PM
State	Finished
Completed on	Friday, 24 May 2024, 3:35 PM
Time taken	11 mins 23 secs
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

```

1 def differenceSum(n):
2     nu=str(n)
3     even = sum(int(nu[i]) for i in range(1, len(str(n)), 2))
4     odd = sum(int(nu[i]) for i in range(0, len(str(n)), 2))
5     return abs(even - odd)
6
7 differenceSum(1453)
8
9

```

	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

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

```

1 def productDigits(num):
2     str1 = str(num)
3     even = 1
4     odd = 0
5     for i in str1:
6
7         if int(i)%2==0:
8             even=even*int(i)
9
10        else:
11            odd=odd+int(i)
12
13        return int(even) // int(odd) >1
14
15
16 productDigits(1256)
17

```

	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 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

$1 \leq \text{orderValue} < 10^6$

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
<code>print(christmasDiscount(578))</code>	12

Answer: (penalty regime: 0 %)

Reset answer

```

1 def is_prime(num):
2     if num < 2:
3         return False
4     for i in range(2, int(num**0.5) + 1):
5         if num % i == 0:
6             return False
7
8     return True
9
10 def christmasDiscount(orderValue):
11     discount = 0
12     order_str = str(orderValue)
13     for digit in order_str:
14         num = int(digit)
15         if is_prime(num):
16             discount += num
17
18     return discount
19
20

```

	Test	Expected	Got	
✓	<code>print(christmasDiscount(578))</code>	12	12	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

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

```

1 def coinChange(n):
2     coins = [1, 2, 3, 4]
3     count = 0
4
5     while n > 0:
6         max2 = max([coin for coin in coins if coin <= n])
7         n -= max2
8         count += 1
9
10    return count
11 n=16
12

```

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

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
<code>print(abundant(12))</code>	Yes
<code>print(abundant(13))</code>	No

Answer: (penalty regime: 0 %)

Reset answer

```
1 def abundant(n):
2     divisor=0
3     for i in range(1,n):
4         if n % i == 0:
5             divisor +=i
6     if divisor > n:
7         return "Yes"
8     else:
9         return "No"
10
11 n=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.

◀ [Week9_MCQ](#)

Jump to...

Searching ►