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

Started on	Sunday, 9 June 2024, 11:13 PM
State	Finished
Completed on	Sunday, 9 June 2024, 11:29 PM
Time taken	15 mins 54 secs
Marks	5.00/5.00
Grade	**100.00** out of 100.00

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

Question **1**

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(n):
 2      n=str(n)
 3      e=1
 4      o=0
 5      for i, digit in enumerate(n):
 6          if i%2==0:
 7              o+=int(digit)
 8          else:
 9              e=e*int(digit)
10      return ("True") if e%o==0 else ("False")
11
```

	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 **2**

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(n):
    x=[]
    for i in range (1,n):
        if n%i==0:
            x.append(i)
    return ("Yes") if sum(x)>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.

	Test	Expected	Got	
✓	`print(abundant(12))`	Yes	Yes	✓
✓	`print(abundant(13))`	No	No	✓

Question **3**

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      n=str(n)
 3      a=0
 4      b=0
 5      for i, digit in enumerate(n):
 6          if i%2==0:
 7              a=a+int(digit)
 8          else:
 9              b=b+int(digit)
10      return (abs(a-b))
```

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

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

```python
def coinChange(amount):
    coins = [1, 2, 3, 4]
    dp = [float('inf')] * (amount + 1)
    dp[0] = 0

    for i in range(1, amount + 1):
        for coin in coins:
            if i - coin >= 0:
                dp[i] = min(dp[i], dp[i - coin] + 1)

    return dp[amount]


    print(coinChange(16))  # Output: 4
```

	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

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
1  def checkUgly(n):
2      while n%2==0:
3          n//=2
4      while n%3==0:
5          n//=3
6      while n%5==0:
7          n//=5
8      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.

◄ Week9_MCQ

[Jump to...]

Searching ►