GE19211 / GE23233 / GE23231 - PSPP/PUP

Quiz navigation

Show one page at a time

Finish review

Started on	Friday, 24 May 2024, 8:15 AM
State	Finished
Completed on	Friday, 24 May 2024, 9:06 PM
Time taken	12 hours 50 mins
Marks	5.00/5.00
Grade	**100.00** out of 100.00

Question **1**

Correct

Mark 1.00 out of 1.00

Flag question

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

```
def differenceSum(num):
    num_str = str(num)
    even_sum = 0
    odd_sum = 0
    for i in range(len(num_str)):
        digit = int(num_str[i])
        if (i + 1) % 2 == 0:
            even_sum += digit
        else:
            odd_sum += digit
    difference = abs(even_sum - odd_sum)
    return difference
```

	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

Flag question

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(productOfDigits(1256))`	True
`print(productOfDigits(1595))`	False

Answer: (penalty regime: 0 %)

Reset answer

```
def productOfDigits(number):
    num_str = str(number)
    product_even = 1
    sum_odd = 0
    for i in range(len(num_str)):
        digit = int(num_str[i])
        if (i + 1) % 2 == 0:
            product_even *= digit
        else:
            sum_odd += digit
    if sum_odd == 0:
        return False
    return product_even % sum_odd == 0

if __name__ == "__main__":
    try:
        number = int(input())
        if productOfDigits(number):
            print("TRUE")
        else:
            print("FALSE")
    except EOFError:
        pass
    except ValueError:
        print("ValueError: Invalid input. Please enter a positive integer.")
```

	Test	Expected	Got	
✓	`print(productOfDigits(12562))`	True	True	✓
✓	`print(productOfDigits(15955))`	False	False	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

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 **4**

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.

Question **5**

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(x):
    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.

Finish review