Solution,3feb,2022

1.Write a Python program to find x that minimizes mean squared deviation from a given a list of numbers.

Solution:

```
>>> def test(nums):
>>>     return sum(nums) / len(nums)
>>> nums = [4, -5, 17, -9, 14, 108, -9]
>>> print("Original list:")
>>> print(nums)
>>> print("Minimizes mean squared deviation from the said list of numbers:")
>>> print(test(nums))
>>> nums = [12, -2, 14, 3, -15, 10, -45, 3, 30]
>>> print("Original list:")
>>> print(nums)
>>> print("Minimizes mean squared deviation from the said list of numbers:")
>>> print(test(nums))
```

2.Write a Python program to select a string from a given list of strings with the most unique characters.

Solution:

```
def test(strs):
    return max(strs, key=lambda x: len(set(x)))
strs = ['cat', 'catatatatctsa', 'abcdefhijklmnop', '124259239185125', '', 'foo', 'unique']
print("Original list:")
print(strs)
print("Select a string from the said list of strings with the most unique characters:")
print(test(strs))
strs = ['Green', 'Red', 'Orange', 'Yellow', '', 'White']
print("\nOriginal list:")
print(strs)
print("Select a string from the said list of strings with the most unique characters:")
print(test(strs))
```

```
return largest
strs = ['cat', 'catatatatctsa', 'abcdefhijklmnop', '124259239185125', '', 'foo', 'unique']
print("Original list:")
print(strs)
print("Select a string from the said list of strings with the most unique characters:")
print(test(strs))
strs = ['Green', 'Red', 'Orange', 'Yellow', '', 'White']
print("\nOriginal list:")
print(strs)
print("Select a string from the said list of strings with the most unique characters:")
print(test(strs))
```

3.Write a Python program to find the indices of two numbers that sum to 0 in a given list of numbers.

Solution:

```
def test(nums):
    s = set(nums)
    for i in s:
        if -i in s:
            return [nums.index(i), nums.index(-i)]

nums = [1, -4, 6, 7, 4]
print("Original List:")
print(nums)
print("Indices of two numbers that sum to 0 in the said list:")
print(test(nums))
nums=[1232, -20352, 12547, 12440, 741, 341, 525, 20352, 91, 20]
print("\nOriginal List:")
print(nums)
print("Indices of two numbers that sum to 0 in the said list:")
print(test(nums))
```

4.Write a Python program to find the list of strings that has fewer total characters (including repetitions).

Solution:

```
def test(strs):
    return min(strs, key=lambda x: sum(len(i) for i in x))
strs = [['this', 'list', 'is', 'narrow'], ['I', 'am', 'shorter but wider']]
print("Original List:")
print(strs)
print("\nFind the given list of strings that has fewer total characters:")
print(test(strs))
strs = [['Red', 'Black', 'Pink'], ['Green', 'Red', 'White']]
print("\nOriginal List:")
print(strs)
print("\nFind the given list of strings that has fewer total characters:")
print(test(strs))
```

5.Write a Python program to find the coordinates of a triangle with the given side lengths.

Solution:

```
def test(sides):
   a, b, c = sorted(sides)
   s = sum(sides) / 2 # semi-perimeter
   area = (s * (s - a) * (s - b) * (s - c)) ** 0.5 # Heron's formula
   y = 2 * area / a # height
   x = (c ** 2 - y ** 2) ** 0.5
   return [[0.0, 0.0], [a, 0.0], [x, y]]
sides = [3, 4, 5]
print("Sides of the triangle:",sides)
print("Coordinates of a triangle with the said side lengths:")
print(test(sides))
sides = [5, 6, 7]
print("\nSides of the triangle:",sides)
print("Coordinates of a triangle with the said side lengths:")
print(test(sides))
```

6.Write a Python program to rescale and shift numbers of a given list, so that they cover the range [0, 1].

Solution:

```
def test(nums):
    a = min(nums)
    b = max(nums)
    if b - a == 0:
        return [0.0] + [1.0] * (len(nums) - 1)
    for i in range(len(nums)):
        nums[i] = (nums[i] - a) / (b - a)
    return nums

nums = [18.5, 17.0, 18.0, 19.0, 18.0]
print("Original list:")
print(nums)
print("Rescale and shift the numbers of the said list so that they cover the range [0, 1]:")
print(test(nums))
nums = [13.0, 17.0, 17.0, 15.5, 2.94]
print("\nOriginal list:")
print(nums)
print("Rescale and shift the numbers of the said list so that they cover the range [0, 1]:")
print(test(nums))
```

7.Write a Python program to find the sum of the numbers of a given list among the first k with more than 2 digits.

Solution:

```
def test(nums, k):
    s = 0
    for i in range(len(nums))[:k]:
        if len(str(abs(nums[i])))>2:
            s = s + nums[i]
    return s

nums = [4, 5, 17, 9, 14, 108, -9, 12 ,76]
print("Original list:",nums)
K = 4
print("Value of K:",K)
print("sum of the numbers among the first k with more than 2 digits")
print(test(nums, K))
nums = [4, 5, 17, 9, 14, 108, -9, 12 ,76]
print("\nOriginal list:",nums)
K = 6
print("Value of K:",K)
print("sum of the numbers among the first k with more than 2 digits")
print(test(nums, K))
nums = [114, 215, -117, 119, 14, 108, -9, 12 ,76]
```

```
print("\nOriginal list:",nums)
K = 5
print("Value of K:",K)
print("sum of the numbers among the first k with more than 2 digits")
print(test(nums, K))
print("\nOriginal list:",nums)
K = 1
print("Value of K:",K)
print("sum of the numbers among the first k with more than 2 digits")
print(test(nums, K))
```

8.Write a Python program to compute the product of the odd digits in a given number, or 0 if there aren't any.

Solution:

```
def test(n):
    if any(int(c) % 2 for c in str(n)):
        prod = 1
        for c in str(n):
            if int(c) % 2 == 1:
                prod *= int(c)
        return prod
    return 0

n = 123456789
print("Original Number:",n)
print("Product of the odd digits in the said number, or 0 if there aren't any")
print(test(n))
n = 2468
print("\nOriginal Number:",n)
print("Product of the odd digits in the said number, or 0 if there aren't any")
print(test(n))
n = 13579
print("\nOriginal Number:",n)
print("Product of the odd digits in the said number, or 0 if there aren't any")
print(test(n))
```

9.Write a Python program to find the largest integer divisor of a number n that is less than n.

Solution:

```
def test(n):
    return next(d for d in range(n - 1, 0, -1) if n % d == 0)

n = 18
print("Original number:",n)
print("Largest integer divisor of a number n that is less than n:")
print(test(n))
n = 100
print("Original number:",n)
print("Largest integer divisor of a number n that is less than n:")
print(test(n))
n = 102
print("Original number:",n)
print("Largest integer divisor of a number n that is less than n:")
print(test(n))
n = 500
print("Original number:",n)
print("Largest integer divisor of a number n that is less than n:")
print(test(n))
n = 1000
print("Original number:",n)
print("Largest integer divisor of a number n that is less than n:")
print(test(n))
n = 6500
print("Original number:",n)
print("Largest integer divisor of a number n that is less than n:")
print(test(n)
```

10.Write a Python program to sort the numbers of a given list by the sum of their digits.

Solution:

```
def test(nums):
    return sorted(nums, key=lambda n: sum(int(c) for c in str(n) if c != "-"))

nums = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
print("Original list of numbers:",nums)
print("Sort the numbers of the said list by the sum of their digits:")
print(test(nums))
nums = [23,2,9,34,8,9,10,74]
print("\nOriginal list of numbers:",nums)
print("Sort the numbers of the said list by the sum of their digits:")
print(test(nums))
```