Solution,25 Jan,2022

1.Write a Python program to calculate the hypotenuse of a right angled triangle.

Solution:

```
>>> from math import sqrt
>>> print("Input lengths of shorter triangle sides:")
>>> a = float(input("a: "))
>>> b = float(input("b: "))
>>> c = sqrt(a**2 + b**2)
>>> print("The length of the hypotenuse is:", c )
```

2.Write a Python program to convert the distance (in feet) to inches, yards, and miles.

1 foot = 12 inches
1 yard = 3 feet
1 mile = 5,280 feet

Solution::

```
>>> d_ft = int(input("Input distance in feet: "))
>>> d_inches = d_ft * 12
>>> d_yards = d_ft / 3.0
>>> d_miles = d_ft / 5280.0

>>> print("The distance in inches is %i inches." % d_inches)
>>> print("The distance in yards is %.2f yards." % d_yards)
>>> print("The distance in miles is %.2f miles." % d_miles)
```

3.Write a Python program to convert pressure in kilopascals to pounds per square inch,a millimeter of mercury (mmHg) and atmosphere pressure

Solution:

```
>>> kpa = float(input("Input pressure in in >>> kilopascals> "))
>>> psi = kpa / 6.89475729
>>> mmhg = kpa * 760 / 101.325
>>> atm = kpa / 101.325
>>> print("The pressure in pounds per square inch: %.2f psi"  % (psi))
>>> print("The pressure in millimeter of mercury: %.2f mmHg" % (mmhg))
>>> print("Atmosphere pressure: %.2f atm." % (atm))
```

4.write a program to Create a 5X2 integer array from a range between 100 to 200 such that the difference between each element is 10?

Solution:

```
>>> import numpy
>>>
>>> print("Creating 5X2 array using numpy.arange")
>>> sampleArray = numpy.arange(100, 200, 10)
>>> sampleArray = sampleArray.reshape(5,2)
>>> print (sampleArray)
```

5.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))
```

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

Solutions:

```
>>> 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("Original 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))
```

7.Write a Python program that accepts a string and calculate the number of digits and letters

Solution:

```
>>> s = input("Input a string")
>>> d=l=0
>>> for c in s:
>>>     if c.isdigit():
>>>         d=d+1
>>>     elif c.isalpha():
>>>         l=l+1
>>>     else:
>>>         pass
>>> print("Letters", l)
>>> print("Digits", d)
```

8.Write a NumPy program to get the indices of the sorted elements of a given array.

Solution:

```
>>> import numpy as np
>>> student_id = np.array([1023, 5202, 6230, 1671, 1682, 5241, 4532])
>>> print("Original array:")
>>> print(student_id)
>>> i = np.argsort(student_id)
>>> print("Indices of the sorted elements of a given array:")
>>> print(i)
```

9.Write a NumPy program to sort a given complex array using the real part first, then the imaginary part.

Solution:

```
>>> import numpy  as np
>>> complex_num = [1 + 2j, 3 - 1j, 3 - 2j, 4 - 3j, 3 + 5j]
>>> print("Original array:")
>>> print(complex_num)
>>> print("\nSorted a given complex array using the real part first, then the imaginary part.")
>>> print(np.sort_complex(complex_num))
```

10.Write a NumPy program to partition a given array in a specified position and move all the smaller elements values to the left of the partition, and the remaining values to the right, in arbitrary order (based on random choice).

Solution:

```
>>> nums = np.array([70, 50, 20, 30, -11, 60, 50, 40])
>>> print("Original array:")
>>> print(nums)
>>> print("\nAfter partitioning on 4 the position:")
>>> print(np.partition(nums, 4))
```