Solutions,27jan,2022

1.Write a Python program to calculate the distance between two points using latitude and longitude.

Solution:

```
>>> from math import radians, sin, cos, acos

>>> print("Input coordinates of two points:")
>>> slat = radians(float(input("Starting latitude: ")))
>>> slon = radians(float(input("Ending longitude: ")))
>>> elat = radians(float(input("Starting latitude: ")))
>>> elon = radians(float(input("Ending longitude: ")))
>>>
>>> dist = 6371.01 * acos(sin(slat)*sin(elat) + cos(slat)*cos(elat)*cos(slon - elon))
>>> print("The distance is %.2fkm." % dist)
```

2.Write a Python program for nth Catalan Number.

Solution:

```
>>> def catalan_number(num)
>>>     if num <=1:
>>>          return 1
>>>
>>>     res_num = 0
>>>     for i in range(num):
>>>         res_num += catalan_number(i) * catalan_number(num-i-1)
>>>     return res_num
>>>
>>> for n in range(10):
>>>     print(catalan_number(n))
```

3.. Write a Python program to add, subtract, multiply and divide two fractions.

Solution:

```
>>> import fractions
>>>
>>> f1 = fractions.Fraction(2, 3)
>>> f2 = fractions.Fraction(3, 7)
>>>
>>> print('{} + {} = {}'.format(f1, f2, f1 + f2))
>>> print('{} - {} = {}'.format(f1, f2, f1 - f2))
>>> print('{} * {} = {}'.format(f1, f2, f1 * f2))
>>> print('{} / {} = {}'.format(f1, f2, f1 / f2))
```

4.Write a Python program to convert a floating point number (PI) to an approximate rational value on the various denominator.

Solution:

```
>>> import fractions
>>> import math
>>>
>>> print('PI       =', math.pi)

>>> f_pi = fractions.Fraction(str(math.pi))
>>> print('No limit =', f_pi)

>>> for d in [1, 5,  50, 90, 100, 500, 1000000]:
>>>     limited = f_pi.limit_denominator(d)
>>>     print('{0:8} = {1}'.format(d, limited))
```

5. Write a Python program to create a dot string

Solution:

```
>>>
>>> from math import sin, cos, radians
>>> import sys
>>> for i in range(1000):
>>>     print(' '*int(10*cos(radians(i))+10) + '.')
```

6. Write a Python program to get the nth tetrahedral number from a given integer(n) value

Solution:

```
>>> def test(n):
>>>     return (n * (n + 1) * (n + 2)) / 6
>>>
>>> n = 1
>>> print("\nOriginal Number:",n)
>>> print("Tetrahedral number:",test(n))
>>> n = 2
>>> print("\nOriginal Number:",n)
>>> print("Tetrahedral number:",test(n))
>>> n = 6
>>> print("\nOriginal Number:",n)
>>> print("Tetrahedral number:",test(n))
```

7. Write a Python program to compute Euclidean distance.

Solution:

```
>>>
>>> import math
>>> # Example points in 3-dimensional >>>
>>> x = (5, 6, 7)
>>> y = (8, 9, 9)
>>> distance = math.sqrt(sum([(a - b) ** 2 for a, b in zip(x, y)]))
>>> print("Euclidean distance from x to y: ",distance)
```

8.Write a Python program to get the sum of the powers of all the numbers from start to end (both inclusive).

Solution:

```
>>>> def sum_of_powers(end, power = 2, start = 1):
>>>   return sum([(i) ** power for i in range(start, end + 1)])
>>>
>>> print(sum_of_powers(12))
>>> print(sum_of_powers(12, 3))
>>> print(sum_of_powers(12, 5, 7))
```

9.. Write a NumPy program to test equal, not equal, greater equal, greater and less test of all the elements of two given arrays.

Solution:

```
>>> import numpy as np
>>> x1 = np.array(['Hello', 'PHP', 'JS', 'examples', 'html'], dtype=np.str)
>>> x2 = np.array(['Hello', 'php', 'Java', 'examples', 'html'], dtype=np.str)
>>> print("\nArray1:")
>>> print(x1)
>>> print("Array2:")
>>> print(x2)
>>> print("\nEqual test:")
>>> r = np.char.equal(x1, x2)
>>> print(r)
>>> print("\nNot equal test:")
>>> r = np.char.not_equal(x1, x2)
>>> print(r)
>>> print("\nLess equal test:")
>>> r = np.char.less(x1, x2)
>>> print(r)
```

10.Write a NumPy program to count the number of "P" in a given array, element-wise.

Solution:

```
>>> import numpy as np
>>> x1 = np.array(['Python', 'PHP', 'JS', 'examples', 'html'], dtype=np.str)
>>> print("\nOriginal Array:")
>>> print(x1)
>>> print("Number of ‘P’:")
>>> r = np.char.count(x1, "P")
>>> print(r)
```