Solutions 20 jan,2022

1. Write a Python program to execute a string containing Python code. Go to the editor?
Solution:

```
>>> mycode = 'print("hello world")'
>>> code = """
>>> def mutiply(x,y):
>>>     return x*y

>>> print('Multiply of 2 and 3 is: ',mutiply(2,3))
"""
>>> exec(mycode)
>>> exec(code)
```

2.Write a NumPy program to multiply a 5x3 matrix by a 3x2 matrix and create a real matrix product.?

Solution:

```
>>> import numpy as np
>>> x = np.random.random((5,3))
>>> print("First array:")
>>> print(x)
>>> y = np.random.random((3,2))
>>> print("Second array:")
>>> print(y)
>>> z = np.dot(x, y)
>>> print("Dot product of two arrays:")
>>> print(z)
```

3.Write a Python program to combine two dictionary adding values for common keys

Solution:

```
>>> from collections import Counter
>>> d1 = {'a': 100, 'b': 200, 'c':300}
>>> d2 = {'a': 300, 'b': 200, 'd':400}
>>> d = Counter(d1) + Counter(d2)
>>> print(d)
```

4.Write a Python program to find the pairs of maximum and minimum product from a given list. Use itertools module.

Solution:

```
>>> import itertools as it
>>> def list_max_min_pair(nums):
>>>     result_max = max(it.combinations(nums, 2), key = lambda sub: sub[0] * sub[1])
>>>     result_min = min(it.combinations(nums, 2), key = lambda sub: sub[0] * sub[1])
>>>     return result_max, result_min

>>> nums = [2,5,8,7,4,3,1,9,10,1]
>>> print("The original list: ")
>>>>print(nums)
>>> print("\nPairs of maximum and minimum product from the said list:")
>>> print(list_max_min_pair(nums))
```

5.write a program to count the number of days of specific month?

Solution:

```
>>> import numpy as np
>>> print("Number of days, February, 2016: ")
>>> print(np.datetime64('2016-03-01') - >>> np.datetime64('2016-02-01'))
>>> print("Number of days, February, 2017: ")
>>> print(np.datetime64('2017-03-01') - >>> np.datetime64('2017-02-01'))
>>> print("Number of days, February, 2018: ")
>>> print(np.datetime64('2018-03-01') - >>> np.datetime64('2018-02-01'))
```

6.Write a Python program to find the greatest common divisor (gcd) of two integers.?

Solution:

```
>>> def Recurgcd(a, b):
>>>     low = min(a, b)
>>>     high = max(a, b)
>>>
>>>      if low == 0:
>>>                 return high
>>>      elif low == 1:
```

```
>>>                 return 1
>>>      else:
>>>                 return Recurgcd(low, high%low)
>>> print(Recurgcd(12,14))
```

7.write a program to print alphabet M ?

Solution:

```
>>> for row in range(0,7):
>>>     for column in range(0,7):
>>>        if (column == 1 or column == 5 or (row == 2 and (column == 2 or column == 4)) or
(row == 3 and column == 3)):
>>>            result_str=result_str+"* "
>>>        else:
>>>            result_str=result_str+"  "
>>>    result_str=result_str+"\n"
>>> print(result_str);
```

8.Write a Python program to check the validity of a password (input from users).
At least 1 letter between [a-z] and 1 letter between [A-Z].
At least 1 number between [0-9].
At least 1 character from [$#@].
Minimum length 6 characters.
Maximum length 16 characters.

Solution:

```
>>> import re
>>> p= input("Input your password")
>>> x = True
>>> while x:
>>>    if (len(p)<6 or len(p)>12):
>>>       break
>>>    elif not re.search("[a-z]",p):
>>>          break
>>>     elif not re.search("[0-9]",p):
>>>        break
>>>     elif not re.search("[A-Z]",p):
>>>        break
>>>    elif not re.search("[$#@]",p):
>>>        break
>>>     elif re.search("\s",p):
```

```
>>>        break
>>>     else:
>>>         print("Valid Password")
>>>         x=False
>>>        break
>>>
>>>  if x:
>>>    print("Not a Valid password")
```

9 . Write a Python program to calculate a dog's age in dog's years.
######For the first two years, a dog year is equal to 10.5 human years .after that each dog year equals to 4human years

Solution:

```
>>> h_age = int(input("Input a dog's age in human years: "))

>>> if h_age < 0:
>>>      print("Age must be positive number.")
>>>      exit()
>>> elif h_age <= 2:
>>>      d_age = h_age * 10.5
>>> else:
>>>      d_age = 21 + (h_age - 2)*4

>>> print("The dog's age in dog's years is", d_age)
  "
```

10.. Write a Python program to rotate a Deque Object specified number (positive) of times.?.

Solution:

```
>>> import collections
>>> # declare an empty deque object
>>> dq_object = collections.deque()
>>> # Add elements to the deque - left to right
>>> dq_object.append(2)
>>> dq_object.append(4)
>>> dq_object.append(6)
>>> dq_object.append(8)
>>> dq_object.append(10)
>>> print("Deque before rotation:")
>>>>print(dq_object)
>>> # Rotate once in positive direction
>>> dq_object.rotate()
```

```
>>> print("\nDeque after 1 positive rotation:")
>>> print(dq_object)
>>> # Rotate twice in positive direction
>>> dq_object.rotate(2)
>>> print("\nDeque after 2 positive rotations:")
>>> print(dq_object)
```