Python Programming Basic Assignment - 1

Question1.

Write a Python program to print "Hello Python" ?

Answer:

In [1]:

```
print("Hello Python")
```

```
Hello Python
```

Question2.

Write a Python program to do arithmetical operations addition and division.?

Answer:

In [2]:

```
First_Number = float(input("Enter the first number:  "))
Second_Number = float(input("Enter the second number:  "))

Summation = First_Number + Second_Number
Division = First_Number / Second_Number

print("The sum of ", First_Number, " and ", Second_Number, " is ", Summation)
print("When ", First_Number, " is divided by ", Second_Number,", we get: ", Division)
```

```
Enter the first number:  4.5
Enter the second number:  1.5
The sum of  4.5  and  1.5  is  6.0
When  4.5  is divided by  1.5 , we get:  3.0
```

Question3.

Write a Python program to find the area of a triangle?

Answer:

In [3]:

```
a = float(input("Enter the length of the first side of the triangle:  "))
b = float(input("Enter the length of the second side of the triangle:  "))
c = float(input("Enter the length of the third side of the triangle:  "))

if (a+b)>c and (a+c)>b and (b+c)>a:
    s = ((a + b + c)/2)
    Area = (s*(s - a)*(s - b)*(s - c))**(0.5)
    print("The area of the triangle for the given length of the sides = ", Area)

else:
    print("WARNING!!!   Enter the length of the sides of the triangle such that the sum \
nof the length of any two sides of the triangle is greater than the third side.")
```

```
Enter the length of the first side of the triangle:  3
Enter the length of the second side of the triangle:  4
Enter the length of the third side of the triangle:  5
The area of the triangle for the given length of the sides =  6.0
```

Question4.

Write a Python program to swap two variables?

Answer:

In [4]:

```
Variable1 = input("Enter the first variable: ")
Variable2 = input("Enter the second variable: ")

Temporary_variable = Variable1
Variable1 = Variable2
Variable2 = Temporary_variable

print("First variable after swapping = ", Variable1)
print("Second variable after swapping = ", Variable2)
```

```
Enter the first variable: Onenuron
Enter the second variable: Ineuron
First variable after swapping =  Ineuron
Second variable after swapping =  Onenuron
```

Question5.

Write a Python program to generate a random number?

Answer:

In [5]:

```
import random

Random_Number = random.random()
print("Random Number =", Random_Number)
```

```
Random Number = 0.18376040940155192
```

Python Programming Basic Assignment - 10

Question1.

Write a Python program to find sum of elements in list?

Answer:

In [1]:

```
def List_Elements_Sum(x):
    """
    It is a function to find the sum of all the elements in the given list.
    It takes a list as argument.
    The datatype of elements in the list can be integer, floating-point or complex number
.
    """

    if type(x) == list:
        SUM = 0
        for i in range(len(x)):
            SUM = SUM + x[i]
        print("The sum of all the elements in the given list is ",SUM)

    else:
        print("WARNING!!!!! Please give input in the form of a list")

List_Elements_Sum([1234,23345,456,7])
```

```
The sum of all the elements in the given list is  25042
```

Question2.

Write a Python program to Multiply all numbers in the list?

Answer:

In [2]:

```
def List_Elements_Product(x):

    """
    It is a function to find the product of all the elements in the given list.
    It takes a list as argument.
    The datatype of elements in the list can be integer, floating-point or complex number
.
    """

    if type(x) == list:
        PRODUCT = 1
        for i in range(len(x)):
            PRODUCT = PRODUCT * x[i]
        print("The product of all the elements in the given list is ",PRODUCT)

    else:
        print("WARNING!!!!! Please give input in the form of a list")

List_Elements_Product([1234,23345,456,7])
```

```
The product of all the elements in the given list is  91954274160
```

Question3.

Write a Python program to find smallest number in a list?

Answer:

In [3]:

```
def List_Elements_Smallest(x):

    """
    It is a function to find the smallest number of all the elements in the given list.
    It takes a list as argument.
    The datatype of elements in the list can be integer or floating-point.
    """

    if type(x) == list:
        SMALLEST = x[1]
        for i in range(len(x)):
            if x[1] <= x[i]:
                SMALLEST = x[1]
            else:
                SMALLEST = x[i]
                x[1] = SMALLEST
        print("The smallest of all the elements in the given list is ",SMALLEST)

    else:
        print("WARNING!!!!! Please give input in the form of a list")

List_Elements_Smallest([1234,23345,456,7])
```

```
The smallest of all the elements in the given list is  7
```

Question4.

Write a Python program to find largest number in a list?

Answer:

In [4]:

```
def List_Elements_Largest(x):

    """
    It is a function to find the largest number of all the elements in the given list.
    It takes a list as argument.
    The datatype of elements in the list can be integer or floating-point.
    """

    if type(x) == list:
        LARGEST = x[1]
        for i in range(len(x)):
            if x[1] >= x[i]:
                LARGEST = x[1]
            else:
                LARGEST = x[i]
                x[1] = LARGEST
        print("The largest of all the elements in the given list is ",LARGEST)

    else:
        print("WARNING!!!!! Please give input in the form of a list")

List_Elements_Largest([1234,23345,456,7])
```

```
The smallest of all the elements in the given list is  23345
```

Question5.

Write a Python program to find second largest number in a list?

Answer:

In [12]:

```
def List_Elements_Second_Largest(x):
    """
    It is a function to find the second largest number of all the elements in the given list.
    It takes a list as argument.
    The datatype of elements in the list can be integer or floating-point.
    """

    if type(x) == list:
        Sorted_List = sorted(x)
        Reversed_Sorted_List = sorted(x)[::-1]
        print("The second largest number in the given list is ", Reversed_Sorted_List[1])


List_Elements_Second_Largest([1234,23345,456,7])
```

```
The second largest number in the given list is  1234
```

Question6.

Write a Python program to find N largest elements from a list?

Answer:

In [24]:

```
def List_Elements_N_Largest(x):

    """
    It is a function to find the list of N largest number of all the elements in the given list.
    The value of N is given by the user, it can be any positive number less than or equal to
    the number of elements in the given list.
    It takes a list as argument.
    The datatype of elements in the list can be integer or floating-point.
    """

    if type(x) == list:
        Sorted_List = sorted(x)
        Reversed_Sorted_List = sorted(x)[::-1]

        N = int(input())
        print(Reversed_Sorted_List[:N:])



List_Elements_N_Largest([1234,23345,456,7])
```

```
2
[23345, 1234]
```

Question7.

Write a Python program to print even numbers in a list?

Answer:

In [26]:

```
def List_Elements_Even(x):
    """
    It is a function to find the list of even numbers from the given list.
    It takes a list as argument.
    The datatype of elements in the list can be integer or floating-point.
    """

    if type(x) == list:
        List = []
        for i in range(len(x)):
            if x[i]%2 == 0:
                List.append(x[i])
        print("The even numbers in the given list are: ",List)


List_Elements_Even([1234,23345,456,7])
```

```
The even numbers in the given list are:  [1234, 456]
```

Question8.

Write a Python program to print odd numbers in a List?

Answer:

In [27]:

```
def List_Elements_Odd(x):

    """
    It is a function to find the list of odd numbers from the given list.
    It takes a list as argument.
    The datatype of elements in the list can be integer or floating-point.
    """

    if type(x) == list:
        List = []
        for i in range(len(x)):
            if x[i]%2 != 0:
                List.append(x[i])
        print("The odd numbers in the given list are: ",List)


List_Elements_Odd([1234,23345,456,7])
```

```
The odd numbers in the given list are:  [23345, 7]
```

Question9.

Write a Python program to Remove empty List from List?

Answer:

In [44]:

```
def List_Elements_Remove(x):

    """
    It is a function to remove empty list from the given list.
    It takes a list as argument.
    It returns the remaining list, i.e. the list after removing the empty list, as an output .
    """
    if type(x) == list:
        Modified_List = []
        for i in range(len(x)):
            if x[i] != []:
                Modified_List.append(x[i])
        print("The modified list is as: ", Modified_List)


List_Elements_Remove([1234,23345,456,[],[],7])
```

```
The modified list is as:  [1234, 23345, 456, 7]
```

Question10.

Write a Python program to Cloning or Copying a list?

Answer:

In [48]:

```
def List_Cloning(x):

    """
    It is a unction to copy or clone a given list.
    It takes a list as argument.
    """
    if type(x) == list:
        Cloned_List = x[:]
    print("The cloned list is: ","\n",Cloned_List)

List_Cloning([1234,23345,456,[],[3245,345,34],7])
```

```
The cloned list is:
 [1234, 23345, 456, [], [3245, 345, 34], 7]
```

Question11.

Write a Python program to Count occurrences of an element in a list?

Answer:

In [51]:

```
def List_Count_Occurrence(x, element):

    """
    It is a function to count the number of occurences of a given element in the given li
st.
    It takes two arguments.
    First argument is the list.
    Second argument is the element for which you want to find its number of occurences in
the provided list.
    """

    if type(x) == list:
        Count = 0
        for i in x:
            if i == element:
                Count = Count + 1
        print(element, "has occurred", Count, "number of times in the given list")

List_Count_Occurrence([31,2,3,12,3,3,45,34,5,5,6,67,68,3,3,2,4,44,54,65,67,56,23,4], 3)
```

```
3 has occurred 5 number of times in the given list
```

```
3 has occurred 3 number of times in the given list
```

Python Programming Basic Assignment - 11

Question1.

Write a Python program to find words which are greater than given length k?

Answer:

In [1]:

```
def WORD_FILTER(s,k):

    """
    It is a function to find words which are greater than given length k.
    It takes two arguments.
    First argument should be a string .
    Second arguments should be the value of k in the form of an integer.
    It returns a list as an output.
    """


    List_Of_Required_Words = []
    Store_Room = s.split(" ")
    for i in Store_Room:
        if len(i) > k:
            List_Of_Required_Words.append(i)
    return List_Of_Required_Words

s = input("Enter the string: ")
k = int(input("Enter the value of k: "))
WORD_FILTER(s,k)
```

```
Enter the string: I am solving the Python Programming Basic Assignment Number 11
Enter the value of k: 5
```

Out[1]:

```
['solving', 'Python', 'Programming', 'Assignment', 'Number']
```

Question2.

Write a Python program for removing i-th character from a string?

Answer:

In [2]:

```
def REMOVE_CHARACTER(s,c):

    """
    It is a function to remove i-th character from a string.
    It takes two arguments.
    First argument should be a string .
    Second arguments should be the value of i in the form of an integer.
    The value of i is given by the user.
    It returns a string as an output.
    """

    for i in range(len(s)):
        if i == c:
            s = s.replace(s[i], "", 1)
    return s

s = input("Enter the string: ")
c = int(input("Enter the index value of the character you want to remove: "))
REMOVE_CHARACTER(s,c)
```

```
Enter the string: 0123456789
Enter the index value of the character you want to remove: 5
```

Out[2]:

```
'012346789'
```

Question3.

Write a Python program to split and join a string?

Answer:

In [3]:

```
def JOIN_AFTER_SPLITTING_STRING(string,s,j):

    """
    It is a function to split agiven string and then join the splitted string.
    It takes three arguments.
    First argument should be the string.
    Second argument should be the delimeter on the basis of which you want to split the string.
    Third argument should be the delimeter on the basis of which you want to join the list of splitted string.
    It gives string as an output.
    """

    List_Of_Splitted_String = string.split(s)
    Joined_String = j.join(List_Of_Splitted_String)

    return Joined_String

string = input("Enter the string:   ")
s = input("Enter the delimeter on the basis of which you want to split the string: ")
j = input("Enter the delimeter on the basis of which you want to join the list of splitted string: ")

JOIN_AFTER_SPLITTING_STRING(string,s,j)
```

```
Enter the string:   I am solving the Python Programming Basic Assignment Number 11
Enter the delimeter on the basis of which you want to split the string:
Enter the delimeter on the basis of which you want to join the list of splitted string: ___
```

Out[3]:

```
'I___am___solving___the___Python___Programming___Basic___Assignment___Number___11'
```

Question4.

Write a Python to check if a given string is binary string or not?

Answer:

In [4]:

```
def CHECK_STING_FOR_BINARY(s):
```

```
def CHECK_STING_FOR_BINARY(s):

    """
    It is a function to check if a given string is binary string or not.
    It takes a string as an input.
    """

    STORE_ROOM = set(s)
    BINARAY_SET ={"0", "1"}

    if STORE_ROOM == BINARAY_SET or STORE_ROOM == {"0"} or STORE_ROOM == {"1"}:
        print ("The given string is a binary string.")

    else:
        print("The given string is not a binary string.")

s = input("Enter the string you want to check if it is binary or not:    ")
CHECK_STING_FOR_BINARY(s)
```

```
Enter the string you want to check if it is binary or not:    10001000020000309
The given string is not a binary string.
```

Question5.

Write a Python program to find uncommon words from two Strings?

Answer:

In [5]:

```
def UNCOMMON_FINDER(s1, s2):

    """
    It is a function to find uncommon words from two Strings.
    It takes two arguments, both of which can be a string.
    It returns a list of uncommon words as an output.
    """

    List_Of_Uncommon_Words = []

    List1 = s1.split()
    List2 = s2.split()

    for i in List1:
        if  i not in List2:
            List_Of_Uncommon_Words.append(i)

    for j in List2:
        if  j not in List1:
            List_Of_Uncommon_Words.append(j)

    return List_Of_Uncommon_Words

s1 = input("Enter the first string:  ")
s2 = input("Enter the second string:  ")

print("The uncommon words found from the given two strings is given below in the list.")
UNCOMMON_FINDER(s1, s2)
```

```
Enter the first string:  I am solving the Python Programming Basic Assignment Number 11
Enter the second string:  I am going to solve the Python Programming Basic Assignment Num
ber 12
The uncommon words found from the given two strings is given below in the list.
```

Out[5]:

```
['solving', '11', 'going', 'to', 'solve', '12']
```

Question6.

Write a Python to find all duplicate characters in string?

Answer:

In [6]:

```
def DUPLICATE_FINDER(s):

    """
    It is a function  to find all duplicate characters in string.
    It takes a string as argument.
    It gives a list of all duplicate characters in the given string as an ouput.
    """

    List_Of_Duplicates = []


    for i in s:
        if  s.count(i) > 1:
            List_Of_Duplicates.append(i)


    return set(List_Of_Duplicates)

s = input("Enter the string:  ")


print("The duplicate characters found in the given string is given below in the set.")
DUPLICATE_FINDER(s)
```

```
Enter the string:  MmmmmmAaaaaNnnnGggOo
The duplicate characters found in the given string is given below in the set.
```

Out[6]:

```
{'a', 'g', 'm', 'n'}
```

Question7.

Write a Python Program to check if a string contains any special character?

Answer:

In [8]:

```
def CHECK_SPECIAL_CHAR(s):

    """
    It is a function to check if a string contains any special character or not.
    It takes a string as an input.
    """

    import string as Imported_Module

    Set_Of_Special_Characters = set(Imported_Module.punctuation)

    if any(i in Set_Of_Special_Characters for i in s):
        print("The given string contains special character.")
    else:
        print("The given string does not contain any special character.")
```

```
s = input("Enter the string to check if it contains any special character or not:  ")
CHECK_SPECIAL_CHAR(s)
```

```
Enter the string to check if it contains any special character or not:  I am solving the
Python Programming Basic Assignment Number 11
The given string does not contain any special character.
```

Python Programming Basic Assignment - 12

Question1.

Write a Python program to Extract Unique values dictionary values?

Answer:

In [1]:

```
def Extractor(d):

    """
    It is a function to extract to extract Unique values from dictionary values.
    It takes a dictionary as an argument.
    It gives a set of unique values as an output.
    """

    Store_Room = []
    for i in d.values():
        for j in i:
            Store_Room.append(j)
    return set(Store_Room)

print("The unique  values from the values of the given dictionary are:  ")
Extractor({'a' : [0,1,2,3,4,5], 'b' : [3,4,5,6,7,8], 'c' : [6,7,8,9,10,11], 'd' : [9,10
,11,12,13,14]})
```

```
The unique  values from the values of the given dictionary are:
```

Out[1]:

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}
```

Question2.

Write a Python program to find the sum of all items in a dictionary?

Answer:

In [2]:

```
def SUMMER(d):

    """
    It is a function to find the sum of all values in a dictionary.
    It takes dictionary as argument.
    """

    SUM = 0
    for i in d.values():
        SUM = SUM + i
    return SUM
```

```
print("The summation of all the values of the given dictionary is given below: ")
SUMMER({'a' : 1234, 'b' : 3456, 'c' : 5678, 'd' : 7890})
```

```
The summation of all the values of the given dictionary is given below:
```

Out[2]:

```
18258
```

Question3.

Write a Python program to Merging two Dictionaries?

Answer:

In [3]:

```
def DICTIONARY_MERGER(d1, d2):

    """
    It is a function to to merge two Dictionaries.
    It takes two dictionaries, both of which can be a dictionay.
    It returns a dictionary as an output.
    """

    d = d1.update(d2)
    return d


d1 = {"a" : 123,  "b" : 234, "c" : 345}
d2 = {"d" : 456,  "e" : 567, "f" : 678}

DICTIONARY_MERGER(d1, d2)
print("The Merged dictionary is as:  ", d1)
```

```
The Merged dictionary is as:   {'a': 123, 'b': 234, 'c': 345, 'd': 456, 'e': 567, 'f': 67
8}
```

Question4.

Write a Python program to convert key-values list to flat dictionary?

Answer:

In [4]:

```
def DICT_FLATTERER(d):

    """
    It is a function to convert key-values list to flat dictionary.
    It takes dictionary as an argument.
    It returns a dictionary as an output.

    """

    Flattened_Dictionary = dict(zip(d["Subject"], d["Marks Obtained out of 100"]))
    return Flattened_Dictionary


d = {"Subject" : ["Physics", "Chemistry", "Mathematics"],  "Marks Obtained out of 100" :
[80, 85, 90]}


print("The Flattened dictionary is as:  ", DICT_FLATTERER(d))
```

```
The Flattened dictionary is as:   {'Physics': 80, 'Chemistry': 85, 'Mathematics': 90}
```

Question5.

Write a Python program to insertion at the beginning in OrderedDict?

Answer:

In [5]:

```
def INSERT(x,y):

    """
     It is a function to insert an item at the beginning of a dictionary in its OrderedDi
ct form.
     It takes two arguments.
     First argument should be a dictionary in which the item is to be added.
     Second argument should be a dictionary containing the item to be added.
     It gives an orderedDict as an output
    """

    from collections import OrderedDict as OD

    Previous_Dictionary = OD(x)
    Item_To_Be_Inserted = OD(y)

    Final_Dictionary = OD(list(Item_To_Be_Inserted.items()) + list(Previous_Dictionary.i
tems()))

    return Final_Dictionary

x ={'Physics': 80, 'Chemistry': 85,  'Mathematics': 90}
y = {'Sanskrit': 100}

print("Final Dictionary :", INSERT(x,y))
```

```
Final Dictionary : OrderedDict([('Sanskrit', 100), ('Physics', 80), ('Chemistry', 85), ('
Mathematics', 90)])
```

Question6.

Write a Python program to check order of character in string using OrderedDict()?

Answer:

In [6]:

```
def CHECK_ORDER(s, p):

    """
    It is a function to check order of character in string using OrderedDict().
    It takes two arguments.
    First argument should be the string in which a particular order may be checked if req
uired.
    Second argument is also a string of characters for which the order is to be checked.
    """

    from collections import OrderedDict

    Dictionary_Created = OrderedDict.fromkeys(s)

    Pattern_Length = 0
    for i,j in Dictionary_Created.items():

        if i == p[Pattern_Length]:
            Pattern_Length = Pattern_Length + 1

        if Pattern_Length == len(p):
            return "The order of pattern is correct"

    return "The order of pattern is incorrect"

s = input("Enter the string:  ")
p = input("Enter the pattern for you which you want to check its order in the string:  ")

CHECK_ORDER(s,p)
```

```
Enter the string:  prerajulisation
Enter the pattern for you which you want to check its order in the string:  juli
```

Out[6]:

```
'The order of pattern is correct'
```

Question7.

Write a Python program to sort Python Dictionaries by Key or Value?

Answer:

SORTING BY KEY:

In [7]:

```
def SORT_BY_KEY(d):

    """
    It is a function to sort a dictionary by its keys.
    It takes a dictionary as argument.
    It gives a dictionary as an output.
    """

    from collections import OrderedDict as OD

    print("The original dictionary is :")
    print(d)

    print("The modified dictionary when sorted by key is: ")
    print(sorted(list(d.items())))


d = {'Physics': 80, 'Chemistry': 85, 'Mathematics': 90, 'Sanskrit': 100}
SORT_BY_KEY(d)
```

```
The original dictionary is :
{'Physics': 80, 'Chemistry': 85, 'Mathematics': 90, 'Sanskrit': 100}
The modified dictionary when sorted by key is:
[('Chemistry', 85), ('Mathematics', 90), ('Physics', 80), ('Sanskrit', 100)]
```

SORTING BY VALUE:

In [8]:

```
def SORT_BY_VALUE(d):
    """
    It is a function to sort a dictionary by its values.
    It takes a dictionary as argument.
    It gives a dictionary as an output.
    """

    print("The original dictionary is :")
    print(d)

    sorted_values = sorted(d.values())
    sorted_dict_by_value = {}

    for i in sorted_values:
        for k in d.keys():
            if d[k] == i:
                sorted_dict_by_value[k] = d[k]
                break
    print("The modified dictionary when sorted by value is: ")
    print(sorted_dict_by_value)


d = {'Chemistry': 85, 'Physics': 80, 'Mathematics': 90, 'Sanskrit': 100}
SORT_BY_VALUE(d)
```

```
The original dictionary is :
{'Chemistry': 85, 'Physics': 80, 'Mathematics': 90, 'Sanskrit': 100}
The modified dictionary when sorted by value is:
{'Physics': 80, 'Chemistry': 85, 'Mathematics': 90, 'Sanskrit': 100}
```

Python Programming Basic Assignment - 13

Question 1:

Write a program that calculates and prints the value according to the given formula:
Q = Square root of [(2 *C* D)/H]
Following are the fixed values of C and H:
C is 50. H is 30.
D is the variable whose values should be input to your program in a comma-separated sequence.

Example
Let us assume the following comma separated input sequence is given to the program:
100,150,180
The output of the program should be:
18,22,24

Answer:

In [1]:

```
def EOQ(C,D,H):

    """
    It is a function that calculates and prints the value of Q according to the given for
mula:
    Q = Square root of [(2 * C * D)/H]
    It takes thre arguments, out of which the value of first argument,c and
    the third argument, H has been predefined respectively as 50 and 30.
    You can give different values for the third argument, separated with a comma to get d
ifferent values of Q.
    It gives
    """

    import math
    D = D.split(",")


    Store_Room = []
    for i in D:
        Q = int(round(math.sqrt(2*C*float(i)/H)))
        Store_Room.append(str(Q))

    print("The respective values of Q are:  ")
    print(','.join(Store_Room))

C = 50
H = 30

D = input("Enter different values of D:  ")


EOQ(C,D,H)
```

```
Enter different values of D:  100,150,180
The respective values of Q are:
18,22,24
```

Question 2:

Write a program which takes 2 digits, X,Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column of the array should be i*j. Note: i=0,1.., X-1; j=0,1,¡Y-1. Example Suppose the following inputs are given to the program: 3,5 Then, the output of the program should be: [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

Answer:

In [2]:

```
def TwoD_Array_generator():
    user_input = input("Enter values for row and column number. (Ex.- 4,5): ")
    rows, cols = user_input.split(",")
    rows = int(rows)
    cols = int(cols)

    grid_list = []
    for i in range(rows):
        row = []
        for j in range(cols):
            row.append(i * j)
        grid_list.append(row)
    print(grid_list)

TwoD_Array_generator()
```

```
Enter values for row and column number. (Ex.- 4,5): 7,5
[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8], [0, 3, 6, 9, 12], [0, 4, 8, 12, 16], [
0, 5, 10, 15, 20], [0, 6, 12, 18, 24]]
```

Question 3:

Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically. Suppose the following input is supplied to the program: without,hello,bag,world Then, the output should be: bag,hello,without,world

Answer:

In [3]:

```
def sorter():
    user_input = input("Input words: ")
    list_of_words = user_input.split(",")
    list_of_words.sort()
    return (',').join(list_of_words)

sorter()
```

```
Input words: aserg,dsgh,drth,rtf,awer,dfgh,drtywe,rdfsg
```

Out[3]:

```
'aserg,awer,dfgh,drth,drtywe,dsgh,rdfsg,rtf'
```

Question 4:

Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. Suppose the following input is supplied to the program: hello world and practice makes perfect and hello world again Then, the output should be: again and hello makes perfect practice world

Answer:

In [4]:

```
user_input = input("Input words: ")
list_of_words = user_input.split(" ")
sett = set(list_of_words)
listt = list(sett)
listt.sort()


print((' ').join(listt))
```

```
Input words: eag sdfg drtfghty uer tsdfgh ghrd awert rtdy
awert drtfghty eag ghrd rtdy sdfg tsdfgh uer
```

Question 5:

Write a program that accepts a sentence and calculate the number of letters and digits. Suppose the following input is supplied to the program: hello world! 123 Then, the output should be: LETTERS 10

DIGITS 3

Answer:

In [5]:

```
user_input = input("Input sentence: ")
letters = 0
digits = 0


for i in user_input:

    if i.isalpha():
        letters += 1

    elif i.isdigit():
        digits += 1
    else:
        pass

print("Number of letters in the given sentence is ", letters)
print("Number of digits in the given sentence is ", digits)
```

```
Input sentence:  hello world! 123
Number of letters in the given sentence is  10
Number of digits in the given sentence is  3
```

Question 6:

A website requires the users to input username and password to register. Write a program to check the validity of password input by users. Following are the criteria for checking the password:

1. **At least 1 letter between [a-z]**
2. **At least 1 number between [0-9]**
3. **At least 1 letter between [A-Z]**
4. **At least 1 character from [$#@]**
5. **Minimum length of transaction password: 6**
6. **Maximum length of transaction password: 12 Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma. Example If the following passwords are given as input to the program: ABd1234@1,a F1#,2w3E*,2We3345 Then, the output of the program should be: ABd1234@1**

Answer:

In [6]:

```
user_input = input("Enter your password: ")

List_Of_passwords = user_input.split(",")

for i in List_Of_passwords:
    if len(i) >=6 and len(i) <= 13:
        if any(j.islower() for j in i):
            if any(j.isupper() for j in i):
                if any(j.isdigit() for j in i):
                    if any(j in "$,#,@" for j in i):
                        print(i, " is a valid password. You may proceed ahead.")
```

```
Enter your password: ABd1234@1,a F1#,2w3E*,2We3345
ABd1234@1  is a valid password. You may proceed ahead.
```

Python Programming Basic Assignment - 14

Question 1:

Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

Answer:

In [1]:

```
class Multiples_Of_7:
    def __init__(self):
        number = int(input("Enter the value of n: "))
        self.select_Upper_bound_range = number
    def generator(self):
        for i in range(0, self.select_Upper_bound_range+1):
            if i%7 == 0:
                yield i

result = Multiples_Of_7()
for j in result.generator():
    print(j,end=' ')
```

```
Enter the value of n: 100
0 7 14 21 28 35 42 49 56 63 70 77 84 91 98
```

Question 2:

Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically.

Suppose the following input is supplied to the program:

New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.

Then, the output should be:

2:2

3.:1

3?:1

New:1

Python:5

Read:1

and:1

between:1

choosing:1

or:2

to:1

Answer:

In [2]:

```
user_input = input("Enter your sentence: ")
print()
words_in_the_sentence = user_input.split(" ")

sorted_words = sorted(set(words_in_the_sentence))

sorted_words_list = list(sorted_words)

for i in sorted_words_list:
    number = sorted_words_list.count(i)
    print(f'{i}:{number}',end="\n\n")
```

```
Enter your sentence: New to Python or choosing between Python 2 and Python 3? Read Python
2 or Python 3.

2:1
```

```
3.:1

3?:1

New:1

Python:1

Read:1

and:1

between:1

choosing:1

or:1

to:1
```

Question 3:

Define a class Person and its two child classes: Male and Female. All classes have a method "getGender" which can print "Male" for Male class and "Female" for Female class.

Answer:

In [3]:

```python
class Person():
    def getGender(self):
        return "Use a child class object of Person class to check gender type! "

class Male(Person):
    def getGender(self):
        return "Male"

class Female(Person):
    def getGender(self):
        return "Female"

object_male = Male()
object_Person = Person()
object_female= Female()

print(object_male.getGender())
print(object_Person.getGender())
print(object_female.getGender())
```

```
Male
Use a child class object of Person class to check gender type!
Female
```

Question 4:

Please write a program to generate all sentences where subject is in ["I", "You"] and verb is in ["Play", "Love"] and the object is in ["Hockey", "Football"].

Answer:

In [4]:

```
def sentence_generator():

    Subject = ["I", "You"]
    Verb = ["Play", "Love"]
    Object = ["Hockey", "Football"]

    for s in Subject:
        for v in Verb:
            for o in Object:
                print(s,v,o, end = "\n\n")

sentence_generator()
```

```
I Play Hockey

I Play Football

I Love Hockey

I Love Football

You Play Hockey

You Play Football

You Love Hockey

You Love Football
```

Question 5:

Please write a program to compress and decompress the string "hello world!hello world!hello world!hello world!".

Answer:

In [5]:

```
def compressor():
    string = input("Enter the string you want to compress:  ")
    compressed_string_store_room = string[0]
    count = 1
    for i in range(len(string) - 1):
        if string[i] == string[i+1]:
            count = count + 1
        else:
            if count > 1:
                compressed_string_store_room =  compressed_string_store_room + str(count
)
                compressed_string_store_room = compressed_string_store_room + string[i+1
]
                count = 1
    if count > 1:
        return compressed_string_store_room + str(count)
    else:
        return compressed_string_store_room

compressor()
```

```
Enter the string you want to compress:  mmmmmaaaannnggo
```

Out[5]:

```
'm5a4n3g2o'
```

In [6]:

```
def decompressor():
    string = input("Enter the string you want to decompress:  ")
    decompressed_string = ""
    for i in range(len(string)):
        if string[i].isdigit():
            decompressed_string = decompressed_string + decompressed_string[-1]*(int(str
ing[i])-1)
        else:
            decompressed_string = decompressed_string + string[i]
    return decompressed_string

decompressor()
```

```
Enter the string you want to decompress:  m5a4n3g2o
```

Out[6]:

```
'mmmmmaaaannnggo'
```

Question 6:

Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list.

Answer:

In [7]:

```
List = eval(input("Enter any list: "))
Element = int(input("Enter the element for which you to find its index in the list given
you after sorting:  "))

def Binary_Searcher():
    Sorted_List = sorted(List)

    lowest_index = 0
    highest_index = len(Sorted_List) - 1

    while lowest_index <= highest_index:

        middle_index = (lowest_index + highest_index)//2

        if Sorted_List[middle_index] < Element:
            lowest_index = middle_index + 1

        elif Sorted_List[middle_index] > Element:
            highest_index = middle_index - 1

        else:
            return middle_index
    else:
        return "invalid"

result = Binary_Searcher()


if result != "invalid":
    print(f'{Element} is present at index {result} ')

else:
    print("The element for which you to find its index in not present in the list given b
y you")
```

```
Enter any list: [34234,56456723,345456,234,456567,78,567234,345,234,356432,3456,4567,678,
567,87,34546,67,67,78,3,34,6,9]
```

```
Enter the element for which you to find its index in the list given you after sorting:  5
67
567 is present at index 12
```

Python Programming Basic Assignment - 15

Question 1:

Please write a program using generator to print the numbers which can be divisible by 5 and 7 between 0 and n in comma separated form while n is input by console.

Example:
If the following n is given as input to the program:
100
Then, the output of the program should be:
0,35,70

Answer:

In [1]:

```python
n = int(input("Enter the value of n :       "))

def Multiples_of_Both_5_And_7(x):

    for i in range(0,x+1):
        if i%5 == 0 and i%7 == 0:
            yield i

store_room = []
for i in Multiples_of_Both_5_And_7 (n):
    store_room.append(str(i))
print(",".join(store_room))
```

```
Enter the value of n :       100
0,35,70
```

Question 2:

Please write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console.

Example:
If the following n is given as input to the program:
10
Then, the output of the program should be:
0,2,4,6,8,10

Answer:

In [2]:

```python
n = int(input("Enter the value of n :       "))

def Even_Printer(x):

    for i in range(0,x+1):
        if i%2 == 0:
            yield i

store_room = []
for i in Even_Printer (n):
    store_room.append(str(i))
print(",".join(store_room))
```

```
Enter the value of n :      20
0,2,4,6,8,10,12,14,16,18,20
```

Question 3:

The Fibonacci Sequence is computed based on the following formula:

f(n)=0 if n=0

f(n)=1 if n=1

f(n)=f(n-1)+f(n-2) if n>1

Please write a program using list comprehension to print the Fibonacci Sequence in comma separated form with a given n input by console.

Example:
If the following n is given as input to the program:
7

Then, the output of the program should be:
0,1,1,2,3,5,8,13

Answer:

In [3]:

```
n = int(input("Enter any positive value of n:   "))

def fibonaci_generator(x):

    if x >= 0:
        if x == 0:
            return 0
        elif x == 1:
            return 1
        else:
            return fibonaci_generator(x-1) + fibonaci_generator(x-2)

store_room = [str(fibonaci_generator(i)) for i in range(n + 1)]
print(",".join(store_room))
```

```
Enter any positive value of n:   10
0,1,1,2,3,5,8,13,21,34,55
```

Question 4:

Assuming that we have some email addresses in the "username@companyname.com" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only.

Example:

Example:

If the following email address is given as input to the program:
john@google.com
Then, the output of the program should be:
john

Answer:

In [4]:

```
userinput = input("Enter your email address:      ")

print(userinput[0:userinput.index("@")])
```

```
Enter your email address:      utkarshchoubey5@gmail.com
utkarshchoubey5
```

Question 5:

Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape's area is 0 by default.

Answer:

In [5]:

```
class Shape:
    def area(self):
        return 0

class Square(Shape):

    def __init__(self, side_length):
        self.side_length = side_length
    def area(self):
        print(f"The area of the Square is {(self.side_length)**2}")

Object = Square(int(input("Enter the length of each side of the Square:  ")))
Object.area()
```

```
Enter the length of each side of the Square:  44
The area of the Square is 1936
```

Python Programming Basic Assignment - 16

Question1.

Write a function that stutters a word as if someone is struggling to read it. The first two letters are repeated twice with an ellipsis ... and space after each, and then the word is pronounced with a question mark ?.

Examples

stutter("incredible") ➞ "in... in... incredible?"
stutter("enthusiastic") ➞ "en... en... enthusiastic?"
stutter("outstanding") ➞ "ou... ou... outstanding?"

Hint :- Assume all input is in lower case and at least two characters long.

Answer:

In [1]:

```
def stutterer():
    user_input = input("Enter any word:   ")
    output = (user_input[:2]+"..."+" ")*2 + user_input+ "?"
    print(output)
stutterer()
```

```
Enter any word:   Bilingual
Bi... Bi... Bilingual?
```

Question 2.

Create a function that takes an angle in radians and returns the corresponding angle in degrees rounded to one decimal place.

Examples

radians_to_degrees(1) → 57.3

radians_to_degrees(20) → 1145.9

radians_to_degrees(50) → 2864.8

Answer:

In [2]:

```
def radTOdegree():
    import math
    user_input = int(input("Enter the angle in radian:   "))
    output=round((user_input*180/math.pi), ndigits=1)
    return output
radTOdegree()
```

```
Enter the angle in radian:   10
```

Out[2]:

```
573.0
```

Question 3.

In this challenge, establish if a given integer num is a Curzon number. If 1 plus 2 elevated to num is exactly divisible by 1 plus 2 multiplied by num, then num is a Curzon number. Given a non-negative integer num, implement a function that returns True if num is a Curzon number, or False otherwise.

Examples

is_curzon(5) → True
2 * *5 + 1 = 33*
***2*5 + 1 = 11**
33 is a multiple of 11

is_curzon(10) → False
2 * *10 + 1 = 1025*
***2*10 + 1 = 21**
1025 is not a multiple of 21

is_curzon(14) → True
2 * *14 + 1 = 16385*
***2* 14 + 1 = 29**
16385 is a multiple of 29

Answer:

In [3]:

```
def curzon_checker():
    user_input = int(input("Enter the number you want to check: "))
    store1 = (2**user_input) + 1
    store2 = (2*user_input) + 1
    if store1%store2 == 0:
        print(f"True---->{user_input} is a curzon number")
    else:
        print(f"False--->{user_input} is not a curzon number")
curzon_checker()
```

```
Enter the number you want to check: 14
True---->14 is a curzon number
```

#

Question 4.Given the side length x find the area of a hexagon.

Examples

area_of_hexagon(1) ➞ 2.6
area_of_hexagon(2) ➞ 10.4
area_of_hexagon(3) ➞ 23.4

Answer:

In [4]:

```
def hexarea():
    import math
    user_input = int(input("Enter the side length of the hexagon:   "))
    output = round(6*(math.sqrt(3)/4)*(user_input**2), ndigits=1)
    return output

hexarea()
```

```
Enter the side length of the hexagon:   3
```

Out[4]:

```
23.4
```

Question 5.

Create a function that returns a base-2 (binary) representation of a base-10 (decimal) string number. To convert is simple: ((2) means base-2 and (10) means base-10)

010101001(2) = 1 + 8 + 32 + 128.

Going from right to left, the value of the most right bit is 1, now from that every bit to the left will be x2 the value, value of an 8 bit binary numbers are (256, 128, 64, 32, 16, 8, 4, 2, 1).

Examples

Examples

binary(1) ➞ "1"
1*1 = 1

binary(5) ➞ "101"
1*1 + 1*4 = 5

binary(10) ➞ "1010"
1*2 + 1*8 = 10

Answer:

```
In [5]:
```

```
def getBinary():
    user_input = int(input("Enter a Number: "))
    output = bin(user_input).replace('0b','')
    print(f'Binary of {user_input} ➞ {output}')


getBinary()
```

```
Enter a Number: 10
Binary of 10 ➞ 1010
```

Python Programming Basic Assignment - 17

Question1.

Create a function that takes three arguments a, b, c and returns the sum of the numbers that are evenly divided by c from the range a, b inclusive.

Examples:

evenly_divisible(1, 10, 20) ➞ 0
No number between 1 and 10 can be evenly divided by 20.

evenly_divisible(1, 10, 2) ➞ 30
2 + 4 + 6 + 8 + 10 = 30

evenly_divisible(1, 10, 3) ➞ 18
3 + 6 + 9 = 18

Answer:

```
In [1]:
```

```
def evenly_divisible(a,b,c):
    count = 0
    for i in range(a,b+1):
        if i%c == 0:
            count +=i

        else:
            count = count
    return count

evenly_divisible(1,10,20)
```

```
Out[1]:

0
```

Question2.

Create a function that returns True if a given inequality expression is correct and False otherwise.

Examples:

correct_signs("3 < 7 < 11") ➞ True
correct_signs("13 > 44 > 33> 1") ➞ False
correct_signs("1 < 2 < 6 < 9 > 3") ➞ True

Answer:

In [2]:

```
def correct_signs():
    user_input = input("Enter the inequality:   ")
    output = eval(user_input)
    print(f'{user_input} ➞ {output}')
    print()

for i in range(3):
    correct_signs()
```

```
Enter the inequality:   13 > 44 > 33> 1
13 > 44 > 33> 1 ➞ False

Enter the inequality:   3 < 7 < 11
3 < 7 < 11 ➞ True

Enter the inequality:   1 < 2 < 6 < 9 > 3
1 < 2 < 6 < 9 > 3 ➞ True
```

Question3.

Create a function that replaces all the vowels in a string with a specified character.

Examples:

replace_vowels("the aardvark", "#") ➞ "th# ##rdv#rk"
replace_vowels("minnie mouse", "?") ➞ "m?nn?? m??s?"
replace_vowels("shakespeare", " ") ➞ "shksp**r"

Answer:

In [3]:

```
def replace_vowels():

    user_input1 = input("Enter any string:    ")
    user_input2 = input("Enter the special character:    ")

    user_input1_copy = user_input1

    vowels = ['a','e','i','o','u','A','E','I','O','U']
    for i in user_input1:
        if i in vowels:
            user_input1 = user_input1.replace(i, user_input2)
    return user_input1
```

```
for i in range(3):
    print(replace_vowels())
```

```
Enter any string:    the aardvark
Enter the special character:    #
th# ##rdv#rk
Enter any string:    minnie mouse
Enter the special character:    ?
m?nn?? m??s?
Enter any string:    shakespeare
Enter the special character:    *
sh*k*sp**r*
```

Question4.

Write a function that calculates the factorial of a number recursively.

Examples:

factorial(5) → 120
factorial(3) → 6
factorial(1) → 1
factorial(0) → 1

Answer:

In [4]:

```
def factorial(x):

    if x == 0 or x ==1:
        output = 1
    else:
        output = (factorial(x-1))*x

    return output

for i in range(4):
    x = int(input("Enter the number for which you want to find its factorial:   "))
    print(factorial(x))
```

```
Enter the number for which you want to find its factorial:   5
120
Enter the number for which you want to find its factorial:   3
6
Enter the number for which you want to find its factorial:   1
1
Enter the number for which you want to find its factorial:   0
1
```

Question 5

Hamming distance is the number of characters that differ between two strings.

To illustrate:

String1: "abcbba"
String2: "abcbda"
Hamming Distance: 1 - "b" vs. "d" is the only difference.

Create a function that computes the hamming distance between two strings.

Examples

hamming_distance("abcde", "bcdef") ➞ 5
hamming_distance("abcde", "abcde") ➞ 0
hamming_distance("strong", "strung") ➞ 1

In [1]:

```
def hamming_distance(s1,s2):

    count = 0
    for i in range(min(len(s1),len(s2))) :
            if s1[i] != s2[i]:
                count += 1
    return count

for i in range(3):
    s1 = input("Enter string 1:    ")
    s2 = input("Enter string 2:    ")

    print(hamming_distance(s1,s2))
```

```
Enter string 1:    abcde
Enter string 2:    bcdef
5
Enter string 1:    abcde
Enter string 2:    abcde
0
Enter string 1:    strong
Enter string 2:    strung
1
```

Python Programming Basic Assignment - 18

Question 1

Create a function that takes a list of non-negative integers and strings and return a new list without the strings.

Examples:

filter_list([1, 2, "a", "b"]) ➞ [1, 2]
filter_list([1, "a", "b", 0, 15]) ➞ [1, 0, 15]
filter_list([1, 2, "aasf", "1", "123", 123]) ➞ [1, 2, 123]

Answer:

In [1]:

```
def filter_list():

    user_input = eval(input("Enter the list which you want to filter:   "))

    new_list = []

    if type(user_input) != list:
        return "Please input a list!", filter_list
    else:
        for i in user_input:
            if type(i) == str or  type(i) == int:
                if type(i) == int:
                    if i >= 0:
                        new_list.append(i)
                    else:
                        return "The list input by you cannot contain negative integers"
                else:
                    new_list = new_list
            else:
                return "The list input by you can contain only non-negative integers and strings"
    return new_list

for i in range(3):
    print(filter_list())
```

```
Enter the list which you want to filter:   [1, 2, "aasf", "1", "123", 123]
[1, 2, 123]
Enter the list which you want to filter:   [1, 2, "a", "b"]
[1, 2]
Enter the list which you want to filter:   [1, "a", "b", 0, 15]
[1, 0, 15]
```

Question 2

The "Reverser" takes a string as input and returns that string in reverse order, with the opposite case.

Examples

reverse("Hello World") ➞ "DLROw OLLEh"
reverse("ReVeRsE") ➞ "eSrEvEr"
reverse("Radar") ➞ "RADAr"

Answer:

In [2]:

```
def reverse():
    user_input = input("Enter a string:     ")
    reversed_string = user_input[::-1]
    swapped_string = reversed_string.swapcase()

    return swapped_string
reverse()
```

```
Enter a string:     ReVeRsE
```

Out[2]:

```
'eSrEvEr'
```

Question 3

You can assign variables from lists like this:

lst = [1, 2, 3, 4, 5, 6]
first = lst[0]
middle = lst[1:-1]
last = lst[-1]

print(first) ➞ outputs 1

print(first) → outputs 1
print(middle) → outputs [2, 3, 4, 5]
print(last) → outputs 6

With Python 3, you can assign variables from lists in a much more succinct way. Create variables first, middle and last from the given list using destructuring assignment (check the Resources tab for some examples), where:

first → 1
middle → [2, 3, 4, 5]
last → 6

Your task is to unpack the list writeyourcodehere into three variables, being first, middle, and last, with middle being everything in between the first and last element. Then print all three variables.

Answer:

In [3]:

```
def unpacker():
    lst = eval(input("Enter a list:     "))
    print()
    first, *middle, last = lst
    print(f'first → {first}')
    print()
    print(f'middle → {middle}')
    print()
    print(f'last → {last}')

unpacker()
```

```
Enter a list:     [1, 2, 3, 4, 5, 6]

first → 1

middle → [2, 3, 4, 5]

last → 6
```

Question 4

Write a function that calculates the factorial of a number recursively.

Examples:

factorial(5) → 120
factorial(3) → 6
factorial(1) → 1
factorial(0) → 1

Answer:

In [4]:

```
def factorial(x):

    if x == 0 :
        output = 1
    else:
        output = (factorial(x-1))*x

    return output

for i in range(4):
    x = int(input("Enter the number for which you want to find its factorial:   "))
    print(factorial(x))
```

```
Enter the number for which you want to find its factorial:   7
5040
Enter the number for which you want to find its factorial:   5
120
Enter the number for which you want to find its factorial:   0
1
Enter the number for which you want to find its factorial:   1
1
```

Question 5

Write a function that moves all elements of one type to the end of the list.

Examples:

move_to_end([1, 3, 2, 4, 4, 1], 1) ➞ [3, 2, 4, 4, 1, 1]
Move all the 1s to the end of the array.
move_to_end([7, 8, 9, 1, 2, 3, 4], 9) ➞ [7, 8, 1, 2, 3, 4, 9]
move_to_end(["a", "a", "a", "b"], "a") ➞ ["b", "a", "a", "a"]

Answer:

In [5]:

```
def move_to_end(lst, element):

    occurence = lst.count(element)

    store_room =[]
    for i in lst:
        if i != element:
            store_room.append(i)
    for i in range(occurence):
        store_room.append(element)

    return store_room


lst = eval(input("Enter the list:    "))
element = eval(input("Enter the element:    "))
move_to_end(lst, element)
```

```
Enter the list:    [7, 8, 9, 1, 2, 3, 4]
Enter the element:    9
```

Out[5]:

```
[7, 8, 1, 2, 3, 4, 9]
```

Python Programming Basic Assignment - 19

Question 1.

Create a function that takes a string and returns a string in which each character is repeated once.

Examples:

double_char("String") → "SSttrriinngg"
double_char("Hello World!") → "HHeelllloo WWoorrlldd!!"
double*char("1234! ") → "11223344!!__ "*

Answer:

In [1]:

```
def double_char():

    user_input = input("Enter a string:     ")
    output = ""
    for i in user_input:
        output += i*2

    return output

print(double_char())
```

```
Enter a string:     String
SSttrriinngg
```

Question 2.

Create a function that reverses a boolean value and returns the string "boolean expected" if another variable type is given.

Examples:

reverse(True) → False
reverse(False) → True
reverse(0) → "boolean expected"
reverse(None) → "boolean expected"

Answer:

In [2]:

```
def reverse(user_input):


    if type(user_input) != bool:
        return "boolean expected"
    else:
        return not user_input

print(f'reverse(True) → {reverse(True)}')
print(f'reverse(False) → {reverse(False)}')
print(f'reverse(0) → {reverse(0)}')
print(f'reverse(None) → {reverse(None)}')
```

```
reverse(True) → False
reverse(False) → True
reverse(0) → boolean expected
reverse(None) → boolean expected
```

Question 3.

Create a function that returns the thickness (in meters) of a piece of paper after folding it n number of times. The paper starts off with a thickness of 0.5mm.

Examples:

num_layers(1) ➞ "0.001m"
Paper folded once is 1mm (equal to 0.001m)
num_layers(4) ➞ "0.008m"
Paper folded 4 times is 8mm (equal to 0.008m)
num_layers(21) ➞ "1048.576m"
Paper folded 21 times is 1048576mm (equal to 1048.576m)

Answer:

In [3]:

```
def num_layers(x):
    output = str((2**(x-1))/1000)+"m"
    return output

user_input =int(input("Enter the number of times paper has been folded:       "))
num_layers(user_input)
```

```
Enter the number of times paper has been folded:       21
```

Out[3]:

```
'1048.576m'
```

Question 4.

Create a function that takes a single string as argument and returns an ordered list containing the indices of all capital letters in the string.

Examples:

index_of_caps("eDaBiT") ➞ [1, 3, 5]
index_of_caps("eQuINoX") ➞ [1, 3, 4, 6]
index_of_caps("determine") ➞ []
index_of_caps("STRIKE") ➞ [0, 1, 2, 3, 4, 5]
index_of_caps("sUn") ➞ [1]

Answer:

In [4]:

```
def index_of_caps(x):

    store_room = []
    for i in range(len(x)):
        if x[i].isupper():
            store_room.append(i)

    return store_room
user_input = input("Enter a string:   ")
index_of_caps(user_input)
```

```
Enter a string:   eDaBiT
```

Out[4]:

```
[1, 3, 5]
```

Question 5.

Using list comprehensions, create a function that finds all even numbers from 1 to the given number.

Examples:

find_even_nums(8) → [2, 4, 6, 8]
find_even_nums(4) → [2, 4]
find_even_nums(2) → [2]

Answer:

In [5]:

```
def find_even_nums(x):

    store_room = []

    if x >= 2:
        for i in range(2,x+1):
            if i%2 == 0:
                store_room.append(i)
    return store_room

user_input = int(input("Enter a positive integer greater than 1:   "))
find_even_nums(user_input)
```

```
Enter a positive integer greater than 1:   8
```

Out[5]:

```
[2, 4, 6, 8]
```

Python Programming Basic Assignment - 2

Question1.

Write a Python program to convert kilometers to miles?

Answer:

In [1]:

```
K = float(input("Enter number of kilometers to be converted in to miles:  "))
M = K/1.609344
print(K, " kilometers = ", M, " miles ")
```

```
Enter number of kilometers to be converted in to miles:  150
150.0  kilometers =  93.20567883560008  miles
```

Question2.

Write a Python program to convert Celsius to Fahrenheit?

Answer:

In [2]:

```
C = float(input("Enter the temperature in degree Celsius to be converted in to degree Fah
renheit:  "))
F = (C*(9/5)) + 32
```

```
print(C, " degree Celsius = ", F, " degree Fahrenheit ")
```

```
Enter the temperature in degree Celsius to be converted in to degree Fahrenheit:  -40
-40.0  degree Celsius =  -40.0  degree Fahrenheit
```

Question3.

Write a Python program to display calendar?

Answer:

In [3]:

```
import calendar

yyyy = int(input("Enter the year in the xxxx format: "))
mm = int(input("Enter the month in the xx format: "))

print(calendar.month(yyyy,mm))
```

```
Enter the year in the xxxx format: 2022
Enter the month in the xx format: 01
    January 2022
Mo Tu We Th Fr Sa Su
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

Question4.

Write a Python program to solve quadratic equation?

Answer:

In [4]:

```
a = float(input("Enter the coefficient of 2nd power of the variable in the expression of
the quadratic equation: "))
b = float(input("Enter the coefficient of 1st power of the variable in the expression of
the quadratic equation: "))
c = float(input("Enter the value of constant in the expression of the quadratic equation:
"))

i = (((b**2) - (4*a*c))**0.5) / (2*a)

x1 = (-b/(2*a)) - i
x2 = (-b/(2*a)) + i

print("The roots of the quadratic equation are: ", x1, "and ", x2)
```

```
Enter the coefficient of 2nd power of the variable in the expression of the quadratic equ
ation: 12
Enter the coefficient of 1st power of the variable in the expression of the quadratic equ
ation: 23
Enter the value of constant in the expression of the quadratic equation:34
The roots of the quadratic equation are:  (-0.9583333333333335-1.3838101587846345j) and
(-0.9583333333333333+1.3838101587846345j)
```

Question5.

Write a Python program to swap two variables without temp variable?

Answer:

In [5]:

```
Variable1 = input("Enter the first variable: ")
Variable2 = input("Enter the second variable: ")

Variable1, Variable2 = Variable2, Variable1

print("First variable after swapping = ", Variable1)
print("Second variable after swapping = ", Variable2)
```

```
Enter the first variable: Ineuron
Enter the second variable: Oneneuron
First variable after swapping =  Oneneuron
Second variable after swapping =  Ineuron
```

Python Programming Basic Assignment - 20

Question 1.

Create a function that takes a list of strings and integers, and filters out the list so that it returns a list of integers only.

Examples:

filter_list([1, 2, 3, "a", ";b", 4]) → [1, 2, 3, 4]
filter_list(["A", 0, "Edabit", 1729, "Python", "1729"]) → [0, 1729]
filter_list(["Nothing", "here"]) → []

Answer:

In [1]:

```
def filter_list():

    user_input = eval(input("Enter the list which you want to filter:   "))

    new_list = []

    if type(user_input) != list:
        return "Please input a list!", filter_list
    else:
        for i in user_input:
            if type(i) == str or  type(i) == int:
                if type(i) == int:

                    new_list.append(i)

                else:
                    new_list = new_list
            else:
                return "The list input by you can contain only non-negative integers and strings"
    return new_list

for i in range(3):
    print(filter_list())
```

```
Enter the list which you want to filter:   [1, 2, 3, "a", ";b", 4]
[1, 2, 3, 4]
Enter the list which you want to filter:   ["A", 0, "Edabit", 1729, "Python", "1729"]
[0, 1729]
Enter the list which you want to filter:   ["Nothing", "here"]
[]
```

Question 2.

Given a list of numbers, create a function which returns the list but with each element's index in the list added to itself. This means you add 0 to the number at index 0, add 1 to the number at index 1, etc...

Examples:

add_indexes([0, 0, 0, 0, 0]) ➞ [0, 1, 2, 3, 4]
add_indexes([1, 2, 3, 4, 5]) ➞ [1, 3, 5, 7, 9]
add_indexes([5, 4, 3, 2, 1]) ➞ [5, 5, 5, 5, 5]

Answer:

In [2]:

```
def filter_list(x):

    new_list = []

    for i in range(len(x)):
        new_list.append(i+x[i])

    return new_list

for i in range(3):
    user_input = eval(input("Enter the list which you want to filter:   "))
    print(f'{user_input}  ➞  {filter_list(user_input)}')
```

```
Enter the list which you want to filter:   [0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]  ➞  [0, 1, 2, 3, 4]
Enter the list which you want to filter:   [1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]  ➞  [1, 3, 5, 7, 9]
Enter the list which you want to filter:   [5, 4, 3, 2, 1]
[5, 4, 3, 2, 1]  ➞  [5, 5, 5, 5, 5]
```

Question 3.

Create a function that takes the height and radius of a cone as arguments and returns the volume of the cone rounded to the nearest hundredth. See the resources tab for the formula.

Examples:

cone_volume(3, 2) ➞ 12.57
cone_volume(15, 6) ➞ 565.49
cone_volume(18, 0) ➞ 0

Answer:

In [3]:

```
def cone_volume(h,r):
    import math
    output = round(((1/3)*math.pi*(r**2)*h), ndigits=2)

    return output

for i in range(3):
    user_input_for_height_h = eval(input("Enter the height of the cone:   "))
    user_input_for_radius_r = eval(input("Enter the radius of the cone:   "))

    print(f'cone_volume({user_input_for_height_h},{user_input_for_radius_r})  →  {cone_volume(user_input_for_height_h,user_input_for_radius_r)}')
    print()
```

```
Enter the height of the cone:   3
Enter the radius of the cone:   2
cone_volume(3,2)  →  12.57

Enter the height of the cone:   15
Enter the radius of the cone:   6
cone_volume(15,6)  →  565.49

Enter the height of the cone:   18
Enter the radius of the cone:   0
cone_volume(18,0)  →  0.0
```

Question 4.

This Triangular Number Sequence is generated from a pattern of dots that form a triangle. The first 5 numbers of the sequence, or dots, are:
1, 3, 6, 10, 15
This means that the first triangle has just one dot, the second one has three dots, the third one has 6 dots and so on.

Write a function that gives the number of dots with its corresponding triangle number of the sequence.

Examples:

triangle(1) → 1
triangle(6) → 21
triangle(215) → 23220

Answer:

In [4]:

```
def triangle(x):

    count = 1
    if x == 1:
        count =1
    elif x >= 2:
        count = triangle(x-1) + x
    else:
        return "Invalid Input"

    return count

for i in range(3):
    user_input = eval(input("Enter the height of the cone:   "))
    print(f'triangle({user_input})  →  {triangle(user_input)}')
    print()
```

```
Enter the height of the cone:   1
triangle(1)  →  1
triangle(1)  →  1

Enter the height of the cone:   6
triangle(6)  →  21

Enter the height of the cone:   215
triangle(215)  →  23220
```

Question 5.

Create a function that takes a list of numbers between 1 and 10 (excluding one number) and returns the missing number.

Examples:

missing_num([1, 2, 3, 4, 6, 7, 8, 9, 10]) → 5
missing_num([7, 2, 3, 6, 5, 9, 1, 4, 8]) → 10
missing_num([10, 5, 1, 2, 4, 6, 8, 3, 9]) → 7

Answer:

In [5]:

```
def missing_num(lst):

    for i in range(1,11):
        if i not in lst:
            return i

for i in range(3):
    user_input = eval(input("Enter the list:   "))
    print(f'missing_num({user_input})  →  {missing_num(user_input)}')
    print()
```

```
Enter the list:   [1, 2, 3, 4, 6, 7, 8, 9, 10]
missing_num([1, 2, 3, 4, 6, 7, 8, 9, 10])  →  5

Enter the list:   [7, 2, 3, 6, 5, 9, 1, 4, 8]
missing_num([7, 2, 3, 6, 5, 9, 1, 4, 8])  →  10

Enter the list:   [10, 5, 1, 2, 4, 6, 8, 3, 9]
missing_num([10, 5, 1, 2, 4, 6, 8, 3, 9])  →  7
```

Python Programming Basic Assignment - 21

Question 1.

Write a function that takes a list and a number as arguments. Add the number to the end of the list, then remove the first element of the list. The function should then return the updated list.

Examples:

next_in_line([5, 6, 7, 8, 9], 1) → [6, 7, 8, 9, 1]
next_in_line([7, 6, 3, 23, 17], 10) → [6, 3, 23, 17, 10]
next_in_line([1, 10, 20, 42], 6) → [10, 20, 42, 6]
next_in_line([], 6) → "No list has been selected"

Answer:

In [1]:

```
def next_in_line(List, Element):

    store_room = []

    if len(List) >= 1:
        for i in range(1, len(List)):
            store_room.append(List[i])

        store_room.append(Element)

        return store_room

    else:
        output = "No list has been selected"
        return output

for i in range(4):
    List = eval(input("Enter the list:    "))
    Element = eval(input("Enter the element:   "))
    print(f"next_in_line({List}, {Element}) →  {next_in_line(List, Element)}")
```

```
Enter the list:    [5, 6, 7, 8, 9]
Enter the element:   1
next_in_line([5, 6, 7, 8, 9], 1) →  [6, 7, 8, 9, 1]
Enter the list:    [7, 6, 3, 23, 17]
Enter the element:   10
next_in_line([7, 6, 3, 23, 17], 10) →  [6, 3, 23, 17, 10]
Enter the list:    [1, 10, 20, 42 ]
Enter the element:   6
next_in_line([1, 10, 20, 42], 6) →  [10, 20, 42, 6]
Enter the list:    []
Enter the element:   6
next_in_line([], 6) →  No list has been selected
```

Question2.

Create the function that takes a list of dictionaries and returns the sum of people's budgets.

Examples:

get_budgets([
{ "name": "John", "age": 21, "budget": 23000 },
{ "name": "Steve", "age": 32, "budget": 40000 },
{ "name": "Martin", "age": 16, "budget": 2700 }
]) → 65700

get_budgets([
{ "name": "John", "age": 21, "budget": 29000 },
{ "name": "Steve", "age": 32, "budget": 32000 },
{ "name": "Martin", "age": 16, "budget": 1600 }
]) → 62600

Answer:

In [2]:

```
def get_budgets(List_Of_Dictionaries):

    output = 0
    for i in range(len(List_Of_Dictionaries)):
        output = output + (List_Of_Dictionaries[i]).get("budget")
    return output

get_budgets([
{ "name": "John", "age": 21, "budget": 23000 },
{ "name": "Steve", "age": 32, "budget": 40000 },
{ "name": "Martin", "age": 16, "budget": 2700 }
])
```

Out[2]:

```
65700
```

Question3.

Create a function that takes a string and returns a string with its letters in alphabetical order.

Examples:

alphabet_soup("hello") → "ehllo"
alphabet_soup("edabit") → "abdeit"
alphabet_soup("hacker") → "acehkr"
alphabet_soup("geek") → "eegk"
alphabet_soup("javascript") → "aacijprstv"

Answer:

In [3]:

```
def alphabet_soup(String):

    output = sorted(String)
    print(output)
    return "".join(output)
alphabet_soup("hello")
```

```
['e', 'h', 'l', 'l', 'o']
```

Out[3]:

```
'ehllo'
```

Question4.

Suppose that you invest 10,000 dollars for 10 years at an interest rate of 6 percent compounded monthly.What will be the value of your investment at the end of the 10 year period? Create a function that accepts the principal p, the term in years t, the interest rate r, and the number of compounding periods per year n. The function returns the value at the end of term rounded to the nearest cent.

For the example above:

compound_interest(10000, 10, 0.06, 12) → 18193.97

Note that the interest rate is given as a decimal and n=12 because with monthly compounding there are 12 periods per year. Compounding can also be done annually, quarterly, weekly, or daily.

Examples:

compound_interest(100, 1, 0.05, 1) → 105.0
compound_interest(3500, 15, 0.1, 4) → 15399.26
compound_interest(100000, 20, 0.15, 365) → 2007316.26

Answer:

In [4]:

```
def compound_interest(p,t,r,n):

    CI = p*((1+(r/n))**(n*t))
    rounded = round(CI, ndigits = 2)
    return rounded

for i in range(3):
    p = int(input("Enter the principal amount:   "))
    t =  int(input("Enter the time in years:   "))
    r = float(input("Enter the rate of interest:   "))
    n = int(input("Enter the number of compunding years:   "))
    print()
    print(f"compound_interest{(p,t,r,n)} → {compound_interest(p,t,r,n)}")
    print()
    print()
```

```
Enter the principal amount:   100000
Enter the time in years:   20
Enter the rate of interest:   0.15
Enter the number of compunding years:   365

compound_interest(100000, 20, 0.15, 365) → 2007316.26


Enter the principal amount:   100
Enter the time in years:   1
Enter the rate of interest:   0.05
Enter the number of compunding years:   1

compound_interest(100, 1, 0.05, 1) → 105.0


Enter the principal amount:   3500
Enter the time in years:   15
Enter the rate of interest:   0.1
Enter the number of compunding years:   4

compound_interest(3500, 15, 0.1, 4) → 15399.26
```

Question5.

Write a function that takes a list of elements and returns only the integers.

Examples:

return_only_integer([9, 2, "space", "car", "lion", 16]) → [9, 2, 16]
return_only_integer(["hello", 81, "basketball", 123, "fox"]) → [81, 123]
return_only_integer([10, "121", 56, 20, "car", 3, "lion"]) → [10, 56, 20, 3]
return_only_integer(["String", True, 3.3, 1]) → [1]

Answer:

In [5]:

```
def return_only_integer(List):
    store_room = []
    for i in List:
        if type(i)==int:
            store_room.append(i)
    return store_room

for i in range(4):

    List = eval(input("Enter the List:   "))
    print(f"return_only_integer{(List)} ➞ {return_only_integer(List)}")
    print()
    print()
```

```
Enter the List:   [9, 2, "space", "car", "lion", 16]
return_only_integer[9, 2, 'space', 'car', 'lion', 16] ➞ [9, 2, 16]


Enter the List:   ["hello", 81, "basketball", 123, "fox"]
return_only_integer['hello', 81, 'basketball', 123, 'fox'] ➞ [81, 123]


Enter the List:   [10, "121", 56, 20, "car", 3, "lion"]
return_only_integer[10, '121', 56, 20, 'car', 3, 'lion'] ➞ [10, 56, 20, 3]


Enter the List:   ["String", True, 3.3, 1]
return_only_integer['String', True, 3.3, 1] ➞ [1]
```

Python Programming Basic Assignment - 22

Question 1.

Create a function that takes three parameters where:

x is the start of the range (inclusive).
y is the end of the range (inclusive).
n is the divisor to be checked against.

Return an ordered list with numbers in the range that are divisible by the third parameter n. Return an empty list if there are no numbers that are divisible by n.

Examples:

list_operation(1, 10, 3) ➞ [3, 6, 9]
list_operation(7, 9, 2) ➞ [8]
list_operation(15, 20, 7) ➞ []

Answer:

In [1]:

```
def list_operation(s,e,d):

    store_room = []

    for i in range(s,e+1):
        if i%d == 0:
            store_room.append(i)
    return store_room

print(list_operation(1, 10, 3))
print(list_operation(7, 9, 2))
print(list_operation(15, 20, 7))
```

```
[3, 6, 9]
```

```
[8]
[]
```

Question 2.

Create a function that takes in two lists and returns True if the second list follows the first list by one element, and False otherwise. In other words, determine if the second list is the first list shifted to the right by 1.

Examples:

simon_says([1, 2], [5, 1]) → True
simon_says([1, 2], [5, 5]) → False
simon_says([1, 2, 3, 4, 5], [0, 1, 2, 3, 4]) → True
simon_says([1, 2, 3, 4, 5], [5, 5, 1, 2, 3]) → False

Notes:

Both input lists will be of the same length, and will have a minimum length of 2.
The values of the 0-indexed element in the second list and the n-1th indexed element in the first list do not matter.

Answer:

In [2]:

```
def simon_says(l1,l2):

    if l1[0:len(l1)-1] == l2[1:len(l2)]:
        return True
    return False

simon_says([1, 2], [5, 1])
```

Out[2]:

```
True
```

Question 3.

A group of friends have decided to start a secret society. The name will be the first letter of each of their names, sorted in alphabetical order. Create a function that takes in a list of names and returns the name of the secret society.

Examples:

society_name(["Adam", "Sarah", "Malcolm"]) → "AMS"
society_name(["Harry", "Newt", "Luna", "Cho"]) → "CHLN"
society_name(["Phoebe", "Chandler", "Rachel", "Ross", "Monica", "Joey"]) → "CJMPRR"

Answer:

In [3]:

```
def society_name(List):

    store_room = []

    for i in range(len(List)):
        store_room.append(List[i][0])
    output1 = sorted(store_room)
    output2 = "".join(output1)

    return output2

society_name(["Phoebe", "Chandler", "Rachel", "Ross", "Monica", "Joey"])
```

Out[3]:

```
'CJMPRR'
```

Question 4.

An isogram is a word that has no duplicate letters. Create a function that takes a string and returns either True or False depending on whether or not it's an "isogram".

Examples:

is_isogram("Algorism") → True
is_isogram("PasSword") → False
Not case sensitive.
is_isogram("Consecutive") → False

Notes:

Ignore letter case (should not be case sensitive).
All test cases contain valid one word strings.

Answer:

In [4]:

```
def is_isogram(String):

    String = String.upper()
    if len(String) == len(set(String)):
        return True
    return False

is_isogram("Consecutive")
```

Out[4]:

```
False
```

Question 5.

Create a function that takes a string and returns True or False, depending on whether the characters are in order or not.

Examples:

is_in_order("abc") → True
is_in_order("edabit") → False
is_in_order("123") → True
is_in_order("xyzz") → True

Notes:

You don't have to handle empty strings.

Answer:

In [5]:

```
def is_in_order(String):

    if String == "".join(sorted(String)):
        return True
    else:
        return False

is_in_order("123abc")
```

Out[5]:

```
True
```

Python Programming Basic Assignment - 23

Question 1.

Create a function that takes a number as an argument and returns True or False depending on whether the number is symmetrical or not. A number is symmetrical when it is the same as its reverse.

Examples:

is_symmetrical(7227) ➞ True
is_symmetrical(12567) ➞ False
is_symmetrical(44444444) ➞ True
is_symmetrical(9939) ➞ False
is_symmetrical(1112111) ➞ True

Answer:

In [1]:

```
def is_symmetrical(number):

    if str(number) == str(number)[::-1]:
        return True
    else:
        return False

is_symmetrical(1112111)
```

Out[1]:

```
True
```

Question 2.

Given a string of numbers separated by a comma and space, return the product of the numbers.

Examples:

multiply_nums("2, 3") ➞ 6
multiply_nums("1, 2, 3, 4") ➞ 24
multiply_nums("54, 75, 453, 0") ➞ 0
multiply_nums("10, -2") ➞ -20

Answer:

In [2]:

```
def multiply_nums(String):

    List_ofElements_in_string = String.split(",")
    product = 1
    for i in List_ofElements_in_string:
        product = product * int(i)
    return product
multiply_nums("1, 2, 3, 4")
```

Out[2]:

```
24
```

Question 3.

Create a function that squares every digit of a number.

Examples:

square_digits(9119) ➞ 811181
square_digits(2483) ➞ 416649
square_digits(3212) ➞ 9414

Notes:
The function receives an integer and must return an integer.

Answer:

In [3]:

```
def square_digits(number):

    String = ""
    for i in str(number):
        String = String + str(int(str(i))**2)
    return int(String)

square_digits(2483)
```

Out[3]:

```
416649
```

Question 4.

Create a function that sorts a list and removes all duplicate items from it.

Examples:

setify([1, 3, 3, 5, 5]) ➞ [1, 3, 5]
setify([4, 4, 4, 4]) ➞ [4]
setify([5, 7, 8, 9, 10, 15]) ➞ [5, 7, 8, 9, 10, 15]
setify([3, 3, 3, 2, 1]) ➞ [1, 2, 3]

Answer:

Answer:

In [4]:

```
def setify(List):

    return list(set(sorted(List)))

setify([1, 3, 3, 5, 5])
```

Out[4]:

```
[1, 3, 5]
```

Question 5.

Create a function that returns the mean of all digits.

Examples:

mean(42) → 3
mean(12345) → 3
mean(666) → 6

Notes:
The mean of all digits is the sum of digits / how many digits there are
(e.g. mean of digits in 512 is (5+1+2)/3(number of digits) = 8/3=2).

The mean will always be an integer.

Answer:

In [5]:

```
def mean(number):
    SUM = 0
    for i in str(number):
        SUM = SUM + int(str(i))
    return SUM/len(str(number))

mean(123456)
```

Out[5]:

```
3.5
```

Python Programming Basic Assignment - 24

Question 1.

Create a function that takes an integer and returns a list from 1 to the given number, where:

1. **If the number can be divided evenly by 4, amplify it by 10 (i.e. return 10 times the number).**
2. **If the number cannot be divided evenly by 4, simply return the number.**

Examples:

amplify(4) → [1, 2, 3, 40]
amplify(3) → [1, 2, 3]
amplify(25) → [1, 2, 3, 40, 5, 6, 7, 80, 9, 10, 11, 120, 13, 14, 15, 160, 17, 18, 19, 200, 21, 22, 23, 240, 25]

Notes:

The given integer will always be equal to or greater than 1.
Include the number (see example above).
To perform this problem with its intended purpose, try doing it with list comprehensions. If that's too difficult, just solve the challenge any way you can.

Answer:

In [1]:

```
def amplify(number):

    store_room = []
    for i in range(1,number+1):
        if i%4 == 0:
            store_room.append(10*i)
        else:
            store_room.append(i)

    return store_room

amplify(15)
```

Out[1]:

```
[1, 2, 3, 40, 5, 6, 7, 80, 9, 10, 11, 120, 13, 14, 15]
```

Question 2.

Create a function that takes a list of numbers and return the number that's unique.

Examples:

unique([3, 3, 3, 7, 3, 3]) → 7
unique([0, 0, 0.77, 0, 0]) → 0.77
unique([0, 1, 1, 1, 1, 1, 1, 1]) → 0

Notes:
Test cases will always have exactly one unique number while all others are the same.

Answer:

In [2]:

```
def unique(List):

    for i in List:
        if List.count(i) == 1:
            element = i
    return element

unique([3, 3, 3, 7, 3, 3])
```

Out[2]:

```
7
```

Question 3.

Your task is to create a Circle constructor that creates a circle with a radius provided by an argument. The circles constructed must have two getters getArea() (PIr^2) and getPerimeter() (2PI*r) which give both respective areas and perimeter (circumference).

For help with this class, I have provided you with a Rectangle constructor which you can use as a base example.

Examples:

**circy = Circle(11)
circy.getArea()
Should return 380.132711084365**

**circy = Circle(4.44)
circy.getPerimeter()
Should return 27.897342763877365**

**Notes:
Round results up to the nearest integer.**

Answer:

In [3]:

```
import math

class Circle:

    def __init__(self, radius):
        self.radius = radius

    def getArea(self):
        area = (math.pi)*(self.radius**2)
        return area

    def getPerimeter(self):
        perimeter = 2*(math.pi)*(self.radius)
        return perimeter

circy = Circle(11)
print(circy.getArea())
print()
circy = Circle(4.44)
print(circy.getPerimeter())
```

```
380.132711084365

27.897342763877365
```

Question4

Create a function that takes a list of strings and return a list, sorted from shortest to longest.

Examples:

**sort_by_length(["Google", "Apple", "Microsoft"]) ➞ ["Apple", "Google", "Microsoft"]
sort_by_length(["Leonardo", "Michelangelo", "Raphael", "Donatello"]) ➞ ["Raphael", "Leonardo", "Donatello", "Michelangelo"]
sort_by_length(["Turing", "Einstein", "Jung"]) ➞ ["Jung", "Turing", "Einstein"]**

**Notes:
All test cases contain lists with strings of different lengths, so you won't have to deal with multiple strings of the same length.**

Answer:

In [4]:

```
def sort_by_length(List):

    sorted_list = sorted(List, key = len)

    return sorted_list

sort_by_length(["Leonardo", "Michelangelo", "Raphael", "Donatello"])
```

Out[4]:

```
['Raphael', 'Leonardo', 'Donatello', 'Michelangelo']
```

Question 5.

Create a function that validates whether three given integers form a Pythagorean triplet. The sum of the squares of the two smallest integers must equal the square of the largest number to be validated.

Examples:

is_triplet(3, 4, 5) → True
$3^2 + 4^2 = 25$
$5^2 = 25$

is_triplet(13, 5, 12) → True
$5^2 + 12^2 = 169$
$13^2 = 169$

is_triplet(1, 2, 3) → False
$1^2 + 2^2 = 5$
$3^2 = 9$

Notes:
Numbers may not be given in a sorted order.

Answer:

In [5]:

```
def is_triplet(x,y,z):

    List = []
    List.extend((x,y,z))
    Sorted_List = sorted(List)


    if Sorted_List[0]**2 + Sorted_List[1]**2 == Sorted_List[2]**2:
        return True

    else:
        return False

is_triplet(13, 6, 12)
```

Out[5]:

```
False
```

Python Programming Basic Assignment - 25

Question 1.

Create a function that takes three integer arguments (a, b, c) and returns the amount of integers which are of equal value.

Examples:

equal(3, 4, 3) ➞ 2
equal(1, 1, 1) ➞ 3
equal(3, 4, 1) ➞ 0

Notes:
Your function must return 0, 2 or 3.

Answer:

```
In [1]:
```

```
def equal(a,b,c):
    count = 0
    if a == b and a == c :
        count = 3
    elif a == b or a == c :
        count = 2
    else:
        count = 0
    return count

equal(3, 3, 3)
```

```
Out[1]:

3
```

Question 2.

Write a function that converts a dictionary into a list of keys-values tuples.

Examples:

dict_to_list({
"D": 1,
"B": 2,
"C": 3
}) ➞ [("B", 2), ("C", 3), ("D", 1)]

dict_to_list({
"likes": 2,
"dislikes": 3,
"followers": 10
}) ➞ [("dislikes", 3), ("followers", 10), ("likes", 2)]

Notes:
Return the elements in the list in alphabetical order.

Answer:

```
In [2]:
```

```
def dict_to_list(Dictionary):
    return [(k,v) for k,v in sorted(Dictionary.items())]

dict_to_list({
    'D': 1,
    'B': 2,
    'C': 3
    })
```

Out[2]:

```
[('B', 2), ('C', 3), ('D', 1)]
```

Question 3.

Write a function that creates a dictionary with each (key, value) pair being the (lower case, upper case) versions of a letter, respectively.

Examples:

mapping(["p", "s"]) → { "p": "P", "s": "S" }
mapping(["a", "b", "c"]) → { "a": "A", "b": "B", "c": "C" }
mapping(["a", "v", "y", "z"]) → { "a": "A", "v": "V", "y": "Y", "z": "Z" }

Notes:
All of the letters in the input list will always be lowercase.

Answer:

In [3]:

```
def mapping(List):

    return {k.lower(): k.upper() for k in List}

mapping(["a", "v", "y", "z"])
```

Out[3]:

```
{'a': 'A', 'v': 'V', 'y': 'Y', 'z': 'Z'}
```

Question 4.

Write a function, that replaces all vowels in a string with a specified vowel.

Examples:

vow_replace("apples and bananas", "u") → "upplus und bununus"
vow_replace("cheese casserole", "o") → "chooso cossorolo"
vow_replace("stuffed jalapeno poppers", "e") →"steffed jelepene peppers"

Notes:
All words will be lowercase. Y is not considered a vowel.

Answer:

In [4]:

```
def vow_replace(String, specified_vowel):
    vowel = ["a","e","i","o","u"]
    for i in String:
        if i.lower() in vowel:
            String = String.replace(i, specified_vowel)
    return String

vow_replace("apples and bananas", "u")
```

Out[4]:

```
'upplus und bununus'
```

Question 5.

Create a function that takes a string as input and capitalizes a letter if its ASCII code is even and returns its lower case version if its ASCII code is odd.

Examples:

ascii_capitalize("to be or not to be!") → "To Be oR NoT To Be!"
ascii_capitalize("THE LITTLE MERMAID") → "THe LiTTLe meRmaiD"
ascii_capitalize("Oh what a beautiful morning.") → "oH wHaT a BeauTiFuL moRNiNg."

Answer:

In [5]:

```
def ascii_capitalize(String):

    for i in String:
        if ord(i) % 2 == 0:
            String = String.replace(i, i.upper())
        else:
            String = String.replace(i, i.lower())
    return String

ascii_capitalize("Oh what a beautiful morning.")
```

Out[5]:

```
'oH wHaT a BeauTiFuL moRNiNg.'
```

Python Programming Basic Assignment - 3

Question1.

Write a Python Program to Check if a Number is Positive, Negative or Zero?

Answer:

In [1]:

```
n = float(input("Enter a number to be checked whether it is Positive, Negative or Zero:
"))

if n<0:
    print("The entered number is Negative")

elif n>0:
    print("The entered number is Positive")
```

```
else:
    print("The entered number is Zero")
```

```
Enter a number to be checked whether it is Positive, Negative or Zero:  -3424
The entered number is Negative
```

Question2.

Write a Python Program to Check if a Number is Odd or Even?

Answer:

In [2]:

```
n = int(input("Enter the number to check if it is Odd or Even:      "))

if n%2 == 0:
    print(n, " is an Even number")
else:
    print(n, " is an Odd number")
```

```
Enter the number to check if it is Odd or Even:      345
345  is an Odd number
```

Question3.

Write a Python Program to Check Leap Year?

Answer:

In [3]:

```
y = int(input("Enter the year to be checked wheter it is a leap year year or not:    "))

if (y%400) == 0:
    print(y, " is a leap year")
elif (y%4) == 0 and (y%100) !=0 :
    print(y, " is a leap year")
else:
    print(y, " is not a leap year")
```

```
Enter the year to be checked wheter it is a leap year year or not:    2022
2022  is not a leap year
```

Question4.

Write a Python Program to Check Prime Number?

Answer:

In [4]:

```
n = int(input("Enter the number to be checked wheter it is Prime Number or not:    "))

if n > 1:

    if n == 2 or n == 3:
        print(n, " is a Prime Number")
    elif (n+1)%6 == 0 or (n-1)%6 == 0:
        print(n, " is a Prime Number")

    else:
        print(n, " is not a Prime Number")

else:
    print(n, " is not a Prime Number")
```

```
Enter the number to be checked wheter it is Prime Number or not:    71
71  is a Prime Number
```

Question5.

Write a Python Program to Print all Prime Numbers in an Interval of 1-10000?

Answer:

In [5]:

```
print("Prime numbers between", 1, "and", 10000, "are: ")

for number in range(1, 10001):
    if number > 1:
        for i in range(2, number):
            if (number % i) == 0:
                break
        else:
            print(number)
```

```
Prime numbers between 1 and 10000 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
101
103
107
109
113
127
131
137
139
149
151
```

```
157
163
167
173
179
181
191
193
197
199
211
223
227
229
233
239
241
251
257
263
269
271
277
281
283
293
307
311
313
317
331
337
347
349
353
359
367
373
379
383
389
397
401
409
419
421
431
433
439
443
449
457
461
463
467
479
487
491
499
503
509
521
523
541
547
557
563
569
571
577
587
593
```

```
599
601
607
613
617
619
631
641
643
647
653
659
661
673
677
683
691
701
709
719
727
733
739
743
751
757
761
769
773
787
797
809
811
821
823
827
829
839
853
857
859
863
877
881
883
887
907
911
919
929
937
941
947
953
967
971
977
983
991
997
1009
1013
1019
1021
1031
1033
1039
1049
1051
1061
1063
1069
```

```
1087
1091
1093
1097
1103
1109
1117
1123
1129
1151
1153
1163
1171
1181
1187
1193
1201
1213
1217
1223
1229
1231
1237
1249
1259
1277
1279
1283
1289
1291
1297
1301
1303
1307
1319
1321
1327
1361
1367
1373
1381
1399
1409
1423
1427
1429
1433
1439
1447
1451
1453
1459
1471
1481
1483
1487
1489
1493
1499
1511
1523
1531
1543
1549
1553
1559
1567
1571
1579
1583
1597
1601
```

```
1607
1609
1613
1619
1621
1627
1637
1657
1663
1667
1669
1693
1697
1699
1709
1721
1723
1733
1741
1747
1753
1759
1777
1783
1787
1789
1801
1811
1823
1831
1847
1861
1867
1871
1873
1877
1879
1889
1901
1907
1913
1931
1933
1949
1951
1973
1979
1987
1993
1997
1999
2003
2011
2017
2027
2029
2039
2053
2063
2069
2081
2083
2087
2089
2099
2111
2113
2129
2131
2137
2141
2143
```

```
2153
2161
2179
2203
2207
2213
2221
2237
2239
2243
2251
2267
2269
2273
2281
2287
2293
2297
2309
2311
2333
2339
2341
2347
2351
2357
2371
2377
2381
2383
2389
2393
2399
2411
2417
2423
2437
2441
2447
2459
2467
2473
2477
2503
2521
2531
2539
2543
2549
2551
2557
2579
2591
2593
2609
2617
2621
2633
2647
2657
2659
2663
2671
2677
2683
2687
2689
2693
2699
2707
2711
2713
```

```
2719
2729
2731
2741
2749
2753
2767
2777
2789
2791
2797
2801
2803
2819
2833
2837
2843
2851
2857
2861
2879
2887
2897
2903
2909
2917
2927
2939
2953
2957
2963
2969
2971
2999
3001
3011
3019
3023
3037
3041
3049
3061
3067
3079
3083
3089
3109
3119
3121
3137
3163
3167
3169
3181
3187
3191
3203
3209
3217
3221
3229
3251
3253
3257
3259
3271
3299
3301
3307
3313
3319
3323
```

```
3329
3331
3343
3347
3359
3361
3371
3373
3389
3391
3407
3413
3433
3449
3457
3461
3463
3467
3469
3491
3499
3511
3517
3527
3529
3533
3539
3541
3547
3557
3559
3571
3581
3583
3593
3607
3613
3617
3623
3631
3637
3643
3659
3671
3673
3677
3691
3697
3701
3709
3719
3727
3733
3739
3761
3767
3769
3779
3793
3797
3803
3821
3823
3833
3847
3851
3853
3863
3877
3881
3889
3907
```

```
3911
3917
3919
3923
3929
3931
3943
3947
3967
3989
4001
4003
4007
4013
4019
4021
4027
4049
4051
4057
4073
4079
4091
4093
4099
4111
4127
4129
4133
4139
4153
4157
4159
4177
4201
4211
4217
4219
4229
4231
4241
4243
4253
4259
4261
4271
4273
4283
4289
4297
4327
4337
4339
4349
4357
4363
4373
4391
4397
4409
4421
4423
4441
4447
4451
4457
4463
4481
4483
4493
4507
4513
```

```
4517
4519
4523
4547
4549
4561
4567
4583
4591
4597
4603
4621
4637
4639
4643
4649
4651
4657
4663
4673
4679
4691
4703
4721
4723
4729
4733
4751
4759
4783
4787
4789
4793
4799
4801
4813
4817
4831
4861
4871
4877
4889
4903
4909
4919
4931
4933
4937
4943
4951
4957
4967
4969
4973
4987
4993
4999
5003
5009
5011
5021
5023
5039
5051
5059
5077
5081
5087
5099
5101
5107
5113
```

```
5119
5147
5153
5167
5171
5179
5189
5197
5209
5227
5231
5233
5237
5261
5273
5279
5281
5297
5303
5309
5323
5333
5347
5351
5381
5387
5393
5399
5407
5413
5417
5419
5431
5437
5441
5443
5449
5471
5477
5479
5483
5501
5503
5507
5519
5521
5527
5531
5557
5563
5569
5573
5581
5591
5623
5639
5641
5647
5651
5653
5657
5659
5669
5683
5689
5693
5701
5711
5717
5737
5741
5743
```

```
5749
5779
5783
5791
5801
5807
5813
5821
5827
5839
5843
5849
5851
5857
5861
5867
5869
5879
5881
5897
5903
5923
5927
5939
5953
5981
5987
6007
6011
6029
6037
6043
6047
6053
6067
6073
6079
6089
6091
6101
6113
6121
6131
6133
6143
6151
6163
6173
6197
6199
6203
6211
6217
6221
6229
6247
6257
6263
6269
6271
6277
6287
6299
6301
6311
6317
6323
6329
6337
6343
6353
6359
```

```
6361
6367
6373
6379
6389
6397
6421
6427
6449
6451
6469
6473
6481
6491
6521
6529
6547
6551
6553
6563
6569
6571
6577
6581
6599
6607
6619
6637
6653
6659
6661
6673
6679
6689
6691
6701
6703
6709
6719
6733
6737
6761
6763
6779
6781
6791
6793
6803
6823
6827
6829
6833
6841
6857
6863
6869
6871
6883
6899
6907
6911
6917
6947
6949
6959
6961
6967
6971
6977
6983
6991
6997
```

```
7001
7013
7019
7027
7039
7043
7057
7069
7079
7103
7109
7121
7127
7129
7151
7159
7177
7187
7193
7207
7211
7213
7219
7229
7237
7243
7247
7253
7283
7297
7307
7309
7321
7331
7333
7349
7351
7369
7393
7411
7417
7433
7451
7457
7459
7477
7481
7487
7489
7499
7507
7517
7523
7529
7537
7541
7547
7549
7559
7561
7573
7577
7583
7589
7591
7603
7607
7621
7639
7643
7649
7669
```

```
7673
7681
7687
7691
7699
7703
7717
7723
7727
7741
7753
7757
7759
7789
7793
7817
7823
7829
7841
7853
7867
7873
7877
7879
7883
7901
7907
7919
7927
7933
7937
7949
7951
7963
7993
8009
8011
8017
8039
8053
8059
8069
8081
8087
8089
8093
8101
8111
8117
8123
8147
8161
8167
8171
8179
8191
8209
8219
8221
8231
8233
8237
8243
8263
8269
8273
8287
8291
8293
8297
8311
8317
```

```
8329
8353
8363
8369
8377
8387
8389
8419
8423
8429
8431
8443
8447
8461
8467
8501
8513
8521
8527
8537
8539
8543
8563
8573
8581
8597
8599
8609
8623
8627
8629
8641
8647
8663
8669
8677
8681
8689
8693
8699
8707
8713
8719
8731
8737
8741
8747
8753
8761
8779
8783
8803
8807
8819
8821
8831
8837
8839
8849
8861
8863
8867
8887
8893
8923
8929
8933
8941
8951
8963
8969
8971
```

```
8999
9001
9007
9011
9013
9029
9041
9043
9049
9059
9067
9091
9103
9109
9127
9133
9137
9151
9157
9161
9173
9181
9187
9199
9203
9209
9221
9227
9239
9241
9257
9277
9281
9283
9293
9311
9319
9323
9337
9341
9343
9349
9371
9377
9391
9397
9403
9413
9419
9421
9431
9433
9437
9439
9461
9463
9467
9473
9479
9491
9497
9511
9521
9533
9539
9547
9551
9587
9601
9613
9619
9623
```

```
9629
9631
9643
9649
9661
9677
9679
9689
9697
9719
9721
9733
9739
9743
9749
9767
9769
9781
9787
9791
9803
9811
9817
9829
9833
9839
9851
9857
9859
9871
9883
9887
9901
9907
9923
9929
9931
9941
9949
9967
9973
```

Python Programming Basic Assignment - 4

Question1.

Write a Python Program to Find the Factorial of a Number?

Answer:

In [1]:

```
n = int(input("Enter the  number for which you have to find its factorial:   "))

if n < 0:
    print("Warning !!! factorial does not exist for negative numbers")

elif n == 0:
    print("The factorial of 0 is 1")

else:
    Factorial = 1

    for i in range(1, (n + 1)):
        Factorial = Factorial*i
    print("The factorial of ", n, "is ", Factorial)
```

```
Enter the  number for which you have to find its factorial:   7
The factorial of  7 is  5040
```

Question2.

Write a Python Program to Display the multiplication Table?

Answer:

In [2]:

```
n = int(input("Enter the  number for which you want multiplication table to be dispalyed:
"))

for i in range(1, 11):
    Product = n*i
    print(n, " X ", i, " = ", Product)
```

```
Enter the  number for which you want multiplication table to be dispalyed:   9
9  X  1  =  9
9  X  2  =  18
9  X  3  =  27
9  X  4  =  36
9  X  5  =  45
9  X  6  =  54
9  X  7  =  63
9  X  8  =  72
9  X  9  =  81
9  X  10  =  90
```

Question3.

Write a Python Program to Print the Fibonacci sequence?

Answer:

In [3]:

```
n1 = 0
n2 = 1

n = int(input("Enter number of terms for which you want to print Fibonacci sequence:  "))

count = 1

if n <= 0:
    print("Enter positive number of terms")

elif n == 1:
    print("Fibonacci sequence for ", n, " term is: ")
    print(n1)

else:
    print("Fibonacci sequence for the first ", n, " terms is: ")
    while count < (n+1):
        print(n1)
        Last_Term = n1 + n2
        n1 = n2
        n2 = Last_Term
        count = count + 1
```

```
Enter number of terms for which you want to print Fibonacci sequence:  8
Fibonacci sequence for the first  8  terms is:
0
1
1
2
3
5
8
13
```

Question4.

Write a Python Program to Check Armstrong Number?

Answer:

In [4]:

```
n = int(input("Enter the number to check if it is an Armstrong number or not:   "))

power = len(str(n))

sum = 0
for i in str(n):
    sum = sum + (int(i))**power
if sum == n:
    print (n, " is an Armstrong number")
else:
    print (n, " is not an Armstrong number")
```

```
Enter the number to check if it is an Armstrong number or not:   6786
6786  is not an Armstrong number
```

Question5.

Write a Python Program to Find Armstrong Number in an Interval?

Answer:

In [5]:

```
l = int(input("Enter the lower range of the interval in which you want to find Armstrong
number:   "))
u = int(input("Enter the upper range of the interval in which you want to find Armstrong
number:   "))

print("Following are the Armstrong numbers in the interval, (", l, ",", u, ")")


for i in range(l, u+1):
    power = len(str(i))

    sum = 0
    for j in str(i):
        sum = sum + (int(j))**power
    if sum == i:
        print (i)
```

```
Enter the lower range of the interval in which you want to find Armstrong number:   123
Enter the upper range of the interval in which you want to find Armstrong number:   56875
Following are the Armstrong numbers in the interval, ( 123 , 56875 )
153
```

```
370
371
407
1634
8208
9474
54748
```

Question6.

Write a Python Program to Find the Sum of Natural Numbers?

Answer:

In [6]:

```python
n = int(input("Enter the number upto which you want the sum of all natural numbers:  "))

sum = 0

for i in range(1, n+1):
    sum = sum + i

print("The sum of first ",n, "natural numbers is ", sum)
```

```
Enter the number upto which you want the sum of all natural numbers:  99
The sum of first  99 natural numbers is  4950
```

Python Programming Basic Assignment - 5

Question1.

Write a Python Program to Find LCM?

Answer:

In [1]:

```python
n1 = int(input("Enter the first number: "))
n2 = int(input("Enter the second number: "))


if n1 > n2:
    Greater = n1
else:
    Greater = n2


while True:
    if Greater% n1 == 0 and Greater%n2 == 0:
        LCM = Greater
        break
    Greater = Greater + 1
print("The LCM of ", n1, " and ", n2, " is ", LCM)
```

```
Enter the first number: 12
Enter the second number: 18
The LCM of  12  and  18  is  36
```

Question2.

Write a Python Program to Find HCF?

Answer:

In [2]:

```
n1 = int(input("Enter the first number: "))
n2 = int(input("Enter the second number: "))


if n1 < n2:
    Lesser = n1
else:
    Lesser = n2


while True:
    if n1% Lesser == 0 and n2%Lesser == 0:
        HCF = Lesser
        break
    Lesser = Lesser - 1
print("The HCF of ", n1, " and ", n2, " is ", HCF)
```

```
Enter the first number: 12
Enter the second number: 18
The HCF of  12  and  18  is  6
```

Question3.

Write a Python Program to convert Decimal to Binary, Octal and Hexadecimal?

Answer:

In [3]:

```
i = int(input("Enter an integer in decimal system: "))

print()
b = bin(i)
o = oct(i)
h = hex(i)

print("The decimal value of", i, "is: ", b[2::], " in Binary system")
print("The decimal value of", i, "is: ", o[2::], " in Octal system")
print("The decimal value of", i, "is: ", h[2::], " in Hexadecimal system")
```

```
Enter an integer in decimal system: 251

The decimal value of 251 is:  11111011  in Binary system
The decimal value of 251 is:  373  in Octal system
The decimal value of 251 is:  fb  in Hexadecimal system
```

Question4.

Write a Python Program To Find ASCII value of a character?

Answer:

In [4]:

```
Character = input("Enter the character for which you want ASCII value: ")

print ("The ASCII value of '" + Character + "' is", ord(Character))
```

```
Enter the character for which you want ASCII value: %
The ASCII value of '%' is 37
```

Question5.

Write a Python Program to Make a Simple Calculator with 4 basic mathematical operations?

Answer:

In [5]:

```
def Addition(x, y):
    return x + y

def Subtraction(x, y):
    return x - y


def Multiplication(x, y):
    return x * y

def Division(x, y):
    return x / y


print("Select the type of operation you want to perform with the help of the calcualtor from the options given below:")
print("1.  Addition")
print("2.  Subtraction")
print("3.  Multiplication")
print("4.  Division")

while True:
    Operation_Type = input("Enter the type of operation(1 or 2 or 3 or 4): ")

    if Operation_Type in ('1', '2', '3', '4'):
        n1 = float(input("Enter the first number: "))
        n2 = float(input("Enter the second number: "))

        if Operation_Type == '1':
            print(n1, "+", n2, "=", Addition(n1, n2))

        elif Operation_Type == '2':
            print(n1, "-", n2, "=", Subtraction(n1, n2))

        elif Operation_Type == '3':
            print(n1, "*", n2, "=", Multiplication(n1, n2))

        elif Operation_Type == '4':
            print(n1, "/", n2, "=", Division(n1, n2))


        Next_Calculation = input("Do you want to perform next calculation? (yes or no): ")

        if Next_Calculation == "no":
            print("Have a nice day!    SEE YOU AGAIN SOON")
            break

    else:
        print("Enter a valid operation type")
```

```
Select the type of operation you want to perform with the help of the calcualtor from the
options given below:
1.  Addition
```

```
2.  Subtraction
3.  Multiplication
4.  Division
Enter the type of operation(1 or 2 or 3 or 4): 5
Enter a valid operation type
Enter the type of operation(1 or 2 or 3 or 4): 4
Enter the first number: 12
Enter the second number: 43534
12.0 / 43534.0 = 0.00027564662103183717
Do you want to perform next calculation? (yes or no): yes
Enter the type of operation(1 or 2 or 3 or 4): 3
Enter the first number: 123
Enter the second number: 23423
123.0 * 23423.0 = 2881029.0
Do you want to perform next calculation? (yes or no): no
Have a nice day!    SEE YOU AGAIN SOON
```

Python Programming Basic Assignment - 6

Question1.

Write a Python Program to Display Fibonacci Sequence Using Recursion?

Answer:

In [1]:

```python
def FSUR(n):
    
    """
    FSUR is a function to display Fibonacci Sequence Using Recursion.
    It takes number of terms as an integer as input.
    It gives a sequence of integers as an output.
    """
    if n <= 1:
        return n
    else:
        return(FSUR(n-1) + FSUR(n-2))

Number_Of_Terms = int(input("Enter number of terms for which you want to print Fibonacci sequence:  "))

if Number_Of_Terms <= 0:
    print("Enter positive number of terms")

else:
    print("Fibonacci sequence for the first ", Number_Of_Terms, " terms is: ")

for i in range(Number_Of_Terms):
    print(FSUR(i))
```

```
Enter number of terms for which you want to print Fibonacci sequence:  10
Fibonacci sequence for the first  10  terms is: 
0
1
1
2
3
5
8
13
21
34
```

Question2.

Write a Python Program to Find Factorial of Number Using Recursion?

Answer:

In [2]:

```
def FNUR(n):
    
    """
    FSUR is a function to find Factorial of a Number Using Recursion.
    It take integer as an input.
    It gives integer as an output.
    """
    if n == 0:
        return 1
    else:
        return((FNUR(n-1)) * n)

Number_Of_Terms = int(input("Enter the  number for which you have to find its factorial: "))

if Number_Of_Terms < 0:
    print("Warning !!! factorial does not exist for negative numbers")

else:
    print("The factorial of ", Number_Of_Terms, "is ", FNUR(Number_Of_Terms))
```

```
Enter the  number for which you have to find its factorial:   5
The factorial of  5 is  120
```

Question3.

Write a Python Program to calculate your Body Mass Index?

Answer:

In [3]:

```
def BMI(Weight, Height):
    
    """
    It is function to calculate Body Mass Index.
    It takes two inputs, first Weight in kilograms, second Height in metres, both as floating point number or integer.
    It gives BMI in the form of a floating point number as an output
    """
    
    bmi = Weight/(Height**2)
    print("The Body Mass Index for the given Weight and Height is ", bmi)

Weight = float(input("Enter weight in kilograms:    "))
Height = float(input("Enter height in metres:    "))

BMI(Weight, Height)
```

```
Enter weight in kilograms:    60
Enter height in metres:    1.7
The Body Mass Index for the given Weight and Height is  20.761245674740486
```

Question4.

Write a Python Program to calculate the natural logarithm of any number?

Answer:

In [4]:

```
def NL(Number):

    """
    It is a function to calculate the natural logarith of any number.
    Note:- The natural logarithm of any number is defined only for positive numbers.
    It takes integer or floating point number as an input.
    It gives output in the form of a floating point number.
    """

    import math

    if Number > 0:
        print("The natural logarithm of ",Number, " is ", math.log(Number))

    else:
        print("The natural logarithm for any number less than or equal to zero is undefin
ed.")

Number = float(input("Enter the number for which you want its natural logarithm:  "))

NL(Number)
```

```
Enter the number for which you want its natural logarithm:  2
The natural logarithm of  2.0  is  0.6931471805599453
```

Question5.

Write a Python Program for cube sum of first n natural numbers?

Answer:

In [5]:

```
def CUBE(x):

    """
    It is a function to find the sum of cube of first n natural numbers.
    It takes Number_Of_Terms as n as input in the form of an integer.
    It gies output in the form of an integer.
    """

    if x == 1:
        return 1
    else:
        return (CUBE(x-1) + (x**3))


Number_Of_Terms = int(input("Enter the natural number upto which you want to find sum of
cubes:  "))

if Number_Of_Terms < 1:
        print("Sorry! given input is not a natural number")

elif Number_Of_Terms == 1:
    print("If you want to find sum, give input any natural number > 1")

else:
    print("The sum of cube of first",Number_Of_Terms, "natural numbers is ", CUBE(Number_
Of_Terms))
```

```
Enter the natural number upto which you want to find sum of cubes:  4
The sum of cube of first 4 natural numbers is  100
```

Python Programming Basic Assignment - 7

Question 1.

Write a Python Program to find sum of array?

Answer:

In [1]:

```
def SumOfArray():
    input_array = eval(input("Enter the Array for which you want to find the sum of its e
lements: "))
    print()
    print(f'The sum of the elements in the array {input_array} is {sum(input_array)}')

SumOfArray()
```

```
Enter the Array for which you want to find the sum of its elements: [1, 2, 3, 4, 5, 6, 7,
8, 9, 10]

The sum of the elements in the array [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] is 55
```

Question 2.

Write a Python Program to find largest element in an array?

Answer:

In [2]:

```
def Largest_Element():
    input_array = eval(input("Enter the Array for which you want to find the largest of i
ts elements: "))
    print()
    print(f'The largest of the elements in the array {input_array} is {max(input_array)}'
)

Largest_Element()
```

```
Enter the Array for which you want to find the largest of its elements: [1, 2, 3, 4, 5, 6,
7, 8, 9, 10]

The largest of the elements in the array [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] is 10
```

Question 3.

Write a Python Program for array rotation?

Answer:

In [3]:

```
def array_Rotation():
    input_array = eval(input("Enter the Array you want to rotate: "))
    print()
    print(f'The array {input_array} after a rotaion is {(input_array)[::-1]}')

array_Rotation()
```

```
Enter the Array you want to rotate: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

The array [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] after a rotaion is [10, 9, 8, 7, 6, 5, 4, 3, 2,
1]
```

Question 4.

Write a Python Program to Split the array and add the first part to the end?

Answer:

In [4]:

```
def array_split_merger():
    input_Array = eval(input("Enter the array you want to split:  "))
    print()
    index_no = int(input("Enter the number of elements you want to split from the array \
nand merge it to the end of the original array: "))
    print()
    output = input_Array[index_no:len(input_Array)]+ input_Array[:index_no]
    print(f"The array {input_Array} after splitting and then merging is {output}")

array_split_merger()
```

```
Enter the array you want to split:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Enter the number of elements you want to split from the array
and merge it to the end of the original array: 5

The array [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] after splitting and then merging is [6, 7, 8, 9,
10, 1, 2, 3, 4, 5]
```

Question 5.

Write a Python Program to check if given array is Monotonic?

Answer:

In [5]:

```
def monotonic_Checker():
    input_Array = eval(input("Enter the Array you want to check if it is monotonic or not
: "))
    print()
    if(all(input_Array[i] <= input_Array[i+1] for i in range(len(input_Array)-1)) or all
(input_Array[i] >= input_Array[i+1] for i in range(len(input_Array)-1))):
        print(f'Array {input_Array} is Monotonic')
    else:
        print(f'Array {input_Array} is not Monotonic')

monotonic_Checker()
monotonic_Checker()
monotonic_Checker()
```

```
Enter the Array you want to check if it is monotonic or not: [1, 2, 3, 4, 5, 6, 7, 8, 9, 1
0]

Array [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] is Monotonic
Enter the Array you want to check if it is monotonic or not: [10, 9, 8, 7, 6, 5, 4, 3, 2,
1]

Array [10, 9, 8, 7, 6, 5, 4, 3, 2, 1] is Monotonic
Enter the Array you want to check if it is monotonic or not: [1, 2, 3, 4, 5, 6, 7, 8, 8888
89, 10]

Array [1, 2, 3, 4, 5, 6, 7, 8, 888889, 10] is not Monotonic
```

Python Programming Basic Assignment - 8

Question 1.

Write a Python Program to Add Two Matrices?

Answer:

In [1]:

```
import numpy as np

matrix1 = np.array([[1,2,3],
        [4,5,6],
        [7,8,9]])
matrix2 = np.array([[9,8,7],
        [6,5,4],
        [3,2,1]])

matrix_sum = np.add(matrix1,matrix2)
print("The Sum of the matrices \n\n{}  and \n\n{}  is \n\n{} ".format(matrix1,matrix2,mat
rix_sum))
```

```
The Sum of the matrices

[[1 2 3]
 [4 5 6]
 [7 8 9]]  and

[[9 8 7]
 [6 5 4]
 [3 2 1]]  is

[[10 10 10]
 [10 10 10]
 [10 10 10]]
```

Question 2.

Write a Python Program to Multiply Two Matrices?

Answer:

In [2]:

```
import numpy as np

matrix1 = np.array([[1,2,3],
        [4,5,6],
        [7,8,9]])
matrix2 = np.array([[9,8,7],
        [6,5,4],
        [3,2,1]])

matrix_product = np.dot(matrix1,matrix2)
print("The Product of the matrices \n\n{}  and \n\n{}  is \n\n{} ".format(matrix1,matrix2
,matrix_product))
```

```
The Product of the matrices

[[1 2 3]
 [4 5 6]
 [7 8 9]]  and

[[9 8 7]
 [6 5 4]
 [3 2 1]]  is

[[ 30  24  18]
 [ 84  69  54]
 [138 114  90]]
```

Question 3.

Write a Python Program to Transpose a Matrix?

Answer:

In [3]:

```
import numpy as np

matrix1 = np.array([[1,2,3],
        [4,5,6],
        [7,8,9]])


matrix_transpose = np.transpose(matrix1)
print("The Transpose of the matrix \n\n{} is \n\n{} ".format(matrix1,matrix_transpose))
```

```
The Transpose of the matrix

[[1 2 3]
 [4 5 6]
 [7 8 9]] is

[[1 4 7]
 [2 5 8]
 [3 6 9]]
```

Question 4.

Write a Python Program to Sort Words in Alphabetic Order?

Answer:

In [4]:

```
def words_sorter():

    user_input = input("Enter a string of words to sort them in alphabetical order: ")
    print()
    sorted_list = sorted(user_input.lower().split())
    print(' '.join(sorted_list))

words_sorter()
```

```
Enter a string of words to sort them in alphabetical order:  Write a Python Program to So
rt Words in Alphabetic Order

a alphabetic in order program python sort to words write
```

Question 5.

Write a Python Program to Remove Punctuation From a String?

Answer:

In [5]:

```
def punctuation_remover():
    import string
    user_input = input("Enter the string from which you want to remove the punctuation: "
)
    output = ''.join([i for i in user_input if i not in string.punctuation])
    return output

punctuation_remover()
```

```
Enter the string from which you want to remove the punctuation: Write a Python Program to
Remove Punctuation From a String?~!@#$%^&*()_+|
```

Out[5]:

```
'Write a Python Program to Remove Punctuation From a String'
```

Python Programming Basic Assignment - 9

Question1.

Write a Python program to check if the given number is a Disarium Number?

Answer:

In [1]:

```
def DisNum(x):

    """
    It is a function to check if a number is a Disarium Number or not.

    NOTE: A number is said to be the Disarium number when the sum of its digit raised to
          the power of their respective positions becomes equal to the number itself.

    It takes positive integer as an input.
    """
    SUM = 0
    count = 1

    for i in str(x):
        SUM = SUM + ((int(i))**count)
        count = count + 1

    if SUM == x:
        print(x, " is a Disarium Number")

    else:
        print(x, " is not a Disarium Number")

x = int(input("Enter the number you want to check if it is a Disarium Number or not:  "))
DisNum(x)
```

```
Enter the number you want to check if it is a Disarium Number or not:  175
175  is a Disarium Number
```

Question2.

Write a Python program to print all disarium numbers between 1 to 100?

Answer:

In [2]:

```
def DisNum(x):

    """
    It is a function to print all Disarium numbers between 1 to 100.
    """

    SUM = 0
    count = 1

    for i in str(x):
        SUM = SUM + ((int(i))**count)
        count = count + 1

    if SUM == x:
        return SUM

CHECK = 0
for i in range(1, 101):
    CHECK =DisNum(i)
    if CHECK == i:
        print(i)
```

```
1
2
3
4
5
6
7
8
9
89
```

Question3.

Write a Python program to check if the given number is Happy Number?

Answer:

In [3]:

```python
def HapNum(x):

    """
    It is a function to check if the given number is Happy Number or not.

    NOTE:
    A number is said to be happy if it yields 1 when replaced by the sum of squares of it
s digits repeatedly.
    If this process results in an endless cycle of numbers containing 4, then the number
will be an unhappy number.

    It takes poditive integer as an input.
    """

    SUM = 0
    for i in str(x):
        SUM = SUM + ((int(i))**2)
    return SUM

v = int(input("Enter the number you want to check if it is a Happy Number or not:  "))
CHECK = v

while CHECK != 1 and CHECK != 4:
    CHECK = HapNum(CHECK)

if CHECK == 1:
    print(v, " is a Happy Number")

elif CHECK == 4:
    print(v," is not a Nappy Number")
```

```
Enter the number you want to check if it is a Happy Number or not:  31
31  is a Happy Number
```

Question4.

Write a Python program to print all happy numbers between 1 and 100?

Answer:

In [4]:

```python
def HapNum(x):

    """
    It is a function to print Happy Numbers between 1 and 100.
    """

    SUM = 0
    for i in str(x):
        SUM = SUM + ((int(i))**2)
    return SUM

print("List of happy numbers between 1 and 100: ")

for i in range(1, 101):
    CHECK = i

    while CHECK != 1 and CHECK != 4:
        CHECK = HapNum(CHECK)

    if CHECK == 1:
        print(i)
```

```
List of happy numbers between 1 and 100:
1
7
10
13
19
23
28
31
32
44
49
68
70
79
82
86
91
94
97
100
```

Question5.

Write a Python program to determine whether the given number is a Harshad Number?

Answer:

In [5]:

```
def HarNum(x):

    """
    It is a function to determine whether the given number is a Harshad Number, also called as Niven Number, or not.
    It takes positive integer as an input
    """

    SUM = 0
    for i in str(x):
        SUM = SUM + int(i)
    if x % SUM == 0:
        print(x, " is a Harshad Number")
    else:
        print(x, " is not a Harshad Number")

x = int(input("Enter the number you want to verify if it is a Harshad Number or not:  "))
HarNum(x)
```

```
Enter the number you want to verify if it is a Harshad Number or not:  156
156  is a Harshad Number
```

Question6.

Write a Python program to print all pronic numbers between 1 and 100?

Answer:

In [6]:

```
def PronicPrint(x):
    """
    It is a function to print all pronic numbers between 1 and 100.
    """

    for i in range(1, 101):
        if type(i*(i+1)) == int and (i*(i+1)) < 100:
            print(i*(i+1))

PronicPrint(x)
```

```
2
6
12
20
30
42
56
72
90
```