1. a)Python Program to Generate all the Divisors of an Integer

# b)Python Program to Find Those Numbers which are Divisible by 7 and Multiple of 5 in a Given Range of Numbers

# a)Python Program to Check if a Number is a Strong Number

# b)Python Program to Check If Two Numbers are Amicable Numbers

# a)Python Program to Find the Sum of Sine Series

# b)Python Program to Find the Sum of Cosine Series

# a)Python Program to Search the Number of Times a Particular Number Occurs in a List

# b)Python Program to Merge Two Lists and Sort it

# a)Python Program to Find the Second Largest Number in a List Using Bubble Sort

# a)Python Program to Find the Union of two Lists

# b)Python Program to Find the Intersection of Two Lists

1. a)Python Program to Create a List of Tuples with the First Element as the Number and Second Element as the Square of the Number

b)Python Program to Detect if Two Strings are Anagrams

# a)Python Program to Form a New String where the First Character and the Last Character have been Exchanged

b)Python Program to Check if a String is a Pangram or Not

1. Python Program to Print All Permutations of a String in Lexicographic Order using Recursion
2. a)Python Program to Sum All the Items in a Dictionary

b)Python Program to Multiply All the Items in a Dictionary

1. a)Python Program to Map Two Lists into a Dictionary

b)Python Program to Count the Number of Vowels Present in a String using Sets

1. Python Program to Find the Area of a Rectangle Using Classes
2. Python Program to Create a Class and Compute the Area and the Perimeter of the Circle
3. Python Program to Create a Class which Performs Basic Calculator Operations
4. Python Program to Create a Class in which One Method Accepts a String from the User and Another Prints it
5. A)Python Program to Generate all the Divisors of an Integer

( a no. that divides into another without a remainder)

n=int(input("Enter an integer:"))

**print**("The divisors of the number are:")

**for** i **in** range(1,n+1):

**if**(n%i==0):

**print**(i)

# 1b)Python Program to Find Those Numbers which are Divisible by 7 and Multiple of 5 in a Given Range of Numbers

**Problem Solution**

1. Take in the upper and lower range and store it in separate variables.  
2. Use a for loop which ranges from the lower range to the upper range.  
3. Then find the numbers which are divisible by both 5 and 7.  
4.Print those numbers  
5. Exit.

1. lower=int(input("Enter the lower range:"))
2. upper=int(input("Enter the upper range:"))
3. **for** i **in** range (lower,upper+1):
4. **if**(i%7==0 **and** i%5==0):
5. **print**(i)

o/p Case 1:

Enter the lower range:1

Enter the upper range:100

35

70

# 2 a)Python Program to Check if a Number is a Strong Number

// is Floor division - division that results into whole number adjusted to the left in the number line

1. x = 15
2. y = 4
3. # Output: x / y = 3.75
4. print('x / y =',x/y)
5. # Output: x // y = 3
6. print('x // y =',x//y)

**Strong Numbers**are the numbers whose sum of factorial of digits is equal to the original number. Given a number, check if it is a Strong Number or not.

**Examples:**

Input : n = 145

Output : Yes

Sum of digit factorials = 1! + 4! + 5!

= 1 + 24 + 120

= 145

Input : n = 534

Output : No

**Problem Description**

The program takes a number and checks if it is a strong number.

**Problem Solution**

1. Take in an integer and store it in a variable.  
2. Using two while loops, find the factorial of each of the digits in the number.  
3. Then sum up all the factorials of the digits.  
4. Check if the sum of the factorials of the digits is equal to the number.  
5. Print the final result.  
6. Exit.

sum1=0

num=int(input("Enter a number:"))

temp=num

**while**(num):

i=1

f=1

r=num%10

**while**(i<=r):

f=f\*i

i=i+1

sum1=sum1+f

num=num//10

**if**(sum1==temp):

**print**("The number is a strong number")

**else**:

**print**("The number is not a strong number")

Case 1:

Enter a number:145

The number is a strong number.

Case 2:

Enter a number:234

The number is not a strong number.

# Python Program to Check If Two Numbers are Amicable Numbers

Amicable Numbers

Two numbers are amicable if the first is equal to the sum of divisors of the second, and if the second number is equal to the sum of divisors of the first.

**Problem Solution**

1. Take in both the integers and store it in separate variables.  
2. Find the sum of the proper divisors of both the numbers.  
3. **Check if the sum of the proper divisors is equal to the opposite numbers.**4. If they are equal, they are amicable numbers.  
5. Print the final result.  
6. Exit.

**Program/Source Code**

Here is the source code of the Python Program to check if two numbers are amicable numbers. The program output is also shown below.

x=int(input('Enter number 1: '))

y=int(input('Enter number 2: '))

sum1=0

sum2=0

**for** i **in** range(1,x):

**if** x%i==0:

sum1+=i

**for** j **in** range(1,y):

**if** y%j==0:

sum2+=j

**if**(sum1==y **and** sum2==x):

**print**('Amicable!')

**else**:

**print**('Not Amicable!')

**Runtime Test Cases**

Case 1:

Enter number 1: 220

Enter number 2: 284

Amicable!

Case 2:

Enter number 1: 349

Enter number 2: 234

Not Amicable!

# Python Program to Find the Sum of Sine Series

**Sine and Cosine: Expansions**

**Series:**  
sin(x) = [sum (k=0..inf)] (-1)k x2k+1 / (2k+1)!  
= x - (1/3!)x3 + (1/5!)x5 - (1/7!)x7   
(This can be derived from **Taylor's Theorem**.)

cos(x) = [sum (k=0..inf)] (-1)k x2k / (2k)!  
= 1 - (1/2!)x2 + (1/4!)x4 - (1/6!)x6   
(This can be derived from **Taylor's Theorem**.)

**Program/Source Code**

Here is source code of the Python Program to find the sum of sine series. The program output is also shown below.

**import** math

**def** sin(x,n):

sine = 0

**for** i **in** range(n):

sign = (-1)\*\*i

pi=22/7

y=x\*(pi/180)

sine = sine + ((y\*\*(2.0\*i+1))/math.factorial(2\*i+1))\*sign

**return** sine

x=int(input("Enter the value of x in degrees:"))

n=int(input("Enter the number of terms:"))

**print**(round(sin(x,n),2))

# Python Program to Find the Sum of Cosine Series

**import** math

**def** cosine(x,n):

cosx = 1

sign = -1

**for** i **in** range(2, n, 2):

pi=22/7

y=x\*(pi/180)

cosx = cosx + (sign\*(y\*\*i))/math.factorial(i)

sign = -sign

**return** cosx

x=int(input("Enter the value of x in degrees:"))

n=int(input("Enter the number of terms:"))

**print**(round(cosine(x,n),2))

# Python Program to Search the Number of Times a Particular Number Occurs in a List

a=[]

n=int(input("Enter number of elements:"))

**for** i **in** range(1,n+1):

b=int(input("Enter element:"))

a.append(b)

k=0

num=int(input("Enter the number to be counted:"))

**for** j **in** a:

**if**(j==num):

k=k+1

**print**("Number of times",num,"appears is",k)

# Python Program to Merge Two Lists and Sort it

a=[]

c=[]

n1=int(input("Enter number of elements:"))

**for** i **in** range(1,n1+1):

b=int(input("Enter element:"))

a.append(b)

n2=int(input("Enter number of elements:"))

**for** i **in** range(1,n2+1):

d=int(input("Enter element:"))

c.append(d)

new=a+c

new.sort()

**print**("Sorted list is:",new)

# Python Program to Find the Second Largest Number in a List Using Bubble Sort

a=[]

n=int(input("Enter number of elements:"))

**for** i **in** range(1,n+1):

b=int(input("Enter element:"))

a.append(b)

**for** i **in** range(0,len(a)):

**for** j **in** range(0,len(a)-i-1):

**if**(a[j]>a[j+1]):

temp=a[j]

a[j]=a[j+1]

a[j+1]=temp

**print**('Second largest number is:',a[n-2])

# Python Program to Find the Union of two Lists

a=[]

n= int(input("Enter the number of elements in list:"))

**for** x **in** range(0,n):

element=int(input("Enter element" + str(x+1) + ":"))

a.append(element)

b=[sum(a[0:x+1]) **for** x **in** range(0,len(a))]

**print**("The original list is: ",a)

**print**("The new list is: ",b)

# Python Program to Find the Intersection of Two Lists

**def** intersection(a, b):

**return** list(set(a) & set(b))

**def** main():

alist=[]

blist=[]

n1=int(input("Enter number of elements for list1:"))

n2=int(input("Enter number of elements for list2:"))

**print**("For list1:")

**for** x **in** range(0,n1):

element=int(input("Enter element" + str(x+1) + ":"))

alist.append(element)

**print**("For list2:")

**for** x **in** range(0,n2):

element=int(input("Enter element" + str(x+1) + ":"))

blist.append(element)

**print**("The intersection is :")

**print**(intersection(alist, blist))

main()

1. Python Program to Create a List of Tuples with the First Element as the Number and Second Element as the Square of the Number

l\_range=int(input("Enter the lower range:"))

u\_range=int(input("Enter the upper range:"))

a=[(x,x\*\*2) **for** x **in** range(l\_range,u\_range+1)]

**print**(a)

**Program Explanation**

1. User must enter the upper and lower range for the numbers.  
2. List comprehension must be used to create a list of tuples where the first number is the number itself from the given range and the second element is a square of the first number.  
3. The list of tuples which is created is printed.

**Runtime Test Cases**

Case 1:

Enter the lower range:1

Enter the upper range:4

[(1, 1), (2, 4), (3, 9), (4, 16)]

Case 2:

Enter the lower range:45

Enter the upper range:49

[(45, 2025), (46, 2116), (47, 2209), (48, 2304), (49, 2401)]

1. Python Program to Detect if Two Strings are Anagrams

**Problem Solution**

1. Take two strings from the user and store them in separate variables.  
2. Then use sorted() to sort both the strings into lists.  
3. Compare the sorted lists and check if they are equal.  
4. Print the final result.  
5. Exit.

**Program/Source Code**

Here is source code of the Python Program to detect if two strings are anagrams. The program output is also shown below.

s1=raw\_input("Enter first string:")

s2=raw\_input("Enter second string:")

**if**(sorted(s1)==sorted(s2)):

**print**("The strings are anagrams.")

**else**:

**print**("The strings aren't anagrams.")

# Python Program to Form a New String where the First Character and the Last Character have been Exchanged

**def** change(string):

**return** string[-1:] + string[1:-1] + string[:1]

string=raw\_input("Enter string:")

**print**("Modified string:")

**print**(change(string))

1. Python Program to Check if a String is a Pangram or Not

**Problem Solution**

1. Take a string from the user and store it in a variable.  
2. Pass the string as an argument to a function.  
3. In the function, form two sets- one of all lower case letters and one of the letters in the string.  
4. Subtract these both sets and check if it is equal to an empty set.  
5. Print the final result.  
6. Exit.

**Program/Source Code**

Here is source code of the Python Program to check if a string is a pangram or not. The program output is also shown below.

**from** string **import** ascii\_lowercase **as** asc\_lower

**def** check(s):

**return** set(asc\_lower) - set(s.lower()) == set([])

strng=raw\_input("Enter string:")

**if**(check(strng)==True):

**print**("The string is a pangram")

**else**:

**print**("The string isn't a pangram")

1. Python Program to Print All Permutations of a String in Lexicographic Order using Recursion

**from** math **import** factorial

**def** print\_permutations\_lexicographic\_order(s):

"""Print all permutations of string s in lexicographic order."""

seq = list(s)

**for** \_ **in** range(factorial(len(seq))):

**print**(''.join(seq))

nxt = get\_next\_permutation(seq)

*# if seq is the highest permutation*

**if** nxt **is** None:

*# then reverse it*

seq.reverse()

**else**:

seq = nxt

**def** get\_next\_permutation(seq):

"""Return next greater lexicographic permutation. Return None if cannot.

This will return the next greater permutation of seq in lexicographic

order. If seq is the highest permutation then this will return None.

seq is a list.

"""

**if** len(seq) == 0:

**return** None

nxt = get\_next\_permutation(seq[1:])

*# if seq[1:] is the highest permutation*

**if** nxt **is** None:

*# reverse seq[1:], so that seq[1:] now is in ascending order*

seq[1:] = reversed(seq[1:])

*# find q such that seq[q] is the smallest element in seq[1:] such that*

*# seq[q] > seq[0]*

q = 1

**while** q < len(seq) **and** seq[0] > seq[q]:

q += 1

*# if cannot find q, then seq is the highest permutation*

**if** q == len(seq):

**return** None

*# swap seq[0] and seq[q]*

seq[0], seq[q] = seq[q], seq[0]

**return** seq

**else**:

**return** [seq[0]] + nxt

s = input('Enter the string: ')

print\_permutations\_lexicographic\_order(s)

1. Python Program to Sum All the Items in a Dictionary

d={'A':100,'B':540,'C':239}

**print**("Total sum of values in the dictionary:")

**print**(sum(d.values()))

1. Python Program to Multiply All the Items in a Dictionary

d={'A':10,'B':10,'C':239}

tot=1

**for** i **in** d:

tot=tot\*d[i]

**print**(tot)

1. Python Program to Map Two Lists into a Dictionary

keys=[]

values=[]

n=int(input("Enter number of elements for dictionary:"))

**print**("For keys:")

**for** x **in** range(0,n):

element=int(input("Enter element" + str(x+1) + ":"))

keys.append(element)

**print**("For values:")

**for** x **in** range(0,n):

element=int(input("Enter element" + str(x+1) + ":"))

values.append(element)

d=dict(zip(keys,values))

**print**("The dictionary is:")

**print**(d)

1. Python Program to Count the Number of Vowels Present in a String using Sets

s=raw\_input("Enter string:")

count = 0

vowels = set("aeiou")

**for** letter **in** s:

**if** letter **in** vowels:

count += 1

**print**("Count of the vowels is:")

**print**(count)

1. Python Program that Displays which Letters are in the First String but not in the Second

s1=raw\_input("Enter first string:")

s2=raw\_input("Enter second string:")

a=list(set(s1)-set(s2))

**print**("The letters are:")

**for** i **in** a:

**print**(i)

# 12. Python Program to Find the Area of a Rectangle Using Classes

**Program/Source Code**

Here is the source code of the Python Program to take the length and breadth from the user and find the area of the rectangle. The program output is also shown below.

**class** rectangle():

**def** \_\_init\_\_(self,breadth,length):

self.breadth=breadth

self.length=length

**def** area(self):

**return** self.breadth\*self.length

a=int(input("Enter length of rectangle: "))

b=int(input("Enter breadth of rectangle: "))

obj=rectangle(a,b)

**print**("Area of rectangle:",obj.area())

**print**()

# 13. Python Program to Create a Class and Compute the Area and the Perimeter of the Circle

**Program/Source Code**

Here is the source code of the Python Program to take the radius from the user and find the area of the circle using classes. The program output is also shown below.

**import** math

**class** circle():

**def** \_\_init\_\_(self,radius):

self.radius=radius

**def** area(self):

**return** math.pi\*(self.radius\*\*2)

**def** perimeter(self):

**return** 2\*math.pi\*self.radius

r=int(input("Enter radius of circle: "))

obj=circle(r)

**print**("Area of circle:",round(obj.area(),2))

**print**("Perimeter of circle:",round(obj.perimeter(),2))

14.Python Program to Create a Class which Performs Basic Calculator Operations

**Program/Source Code**

Here is the source code of the Python Program to take the length and breadth from the user and find the area of the rectangle. The program output is also shown below.

**class** cal():

**def** \_\_init\_\_(self,a,b):

self.a=a

self.b=b

**def** add(self):

**return** self.a+self.b

**def** mul(self):

**return** self.a\*self.b

**def** div(self):

**return** self.a/self.b

**def** sub(self):

**return** self.a-self.b

a=int(input("Enter first number: "))

b=int(input("Enter second number: "))

obj=cal(a,b)

choice=1

**while** choice!=0:

**print**("0. Exit")

**print**("1. Add")

**print**("2. Subtraction")

**print**("3. Multiplication")

**print**("4. Division")

choice=int(input("Enter choice: "))

**if** choice==1:

**print**("Result: ",obj.add())

**elif** choice==2:

**print**("Result: ",obj.sub())

**elif** choice==3:

**print**("Result: ",obj.mul())

**elif** choice==4:

**print**("Result: ",round(obj.div(),2))

**elif** choice==0:

**print**("Exiting!")

**else**:

**print**("Invalid choice!!")

**print**()

15.Python Program to Create a Class in which One Method Accepts a String from the User and Another Prints it

**Program/Source Code**

Here is the source code of the Python Program that takes the string from the user and prints the string using classes. The program output is also shown below.

**class** print1():

**def** \_\_init\_\_(self):

self.string=""

**def** get(self):

self.string=input("Enter string: ")

**def** put(self):

**print**("String is:")

**print**(self.string)

obj=print1()

obj.get()

obj.put()