1. **Write program to check whether number is Harshad number or Not. .**
2. num = 156;
3. rem = sum = 0;
5. #Make a copy of num and store it in variable n
6. n = num;
8. #Calculates sum of digits
9. **while**(num > 0):
10. rem = num%10;
11. sum = sum + rem;
12. num = num//10;
14. #Checks whether the number is divisible by the sum of digits
15. **if**(n%sum == 0):
16. **print**(str(n) + " is a harshad number");
17. **else**:
18. **print**(str(n) + " is not a harshad number");

#Python program to count vowel or consonant of the given string

str=input("Please enter a string as you wish: ");

vowels=0

consonants=0

for i in str:

if(i == 'a'or i == 'e'or i == 'i'or i == 'o'or i == 'u' or

i == 'A'or i == 'E'or i == 'I'or i == 'O'or i == 'U' ):

vowels=vowels+1;#vowel counter is incremented by 1

else:

consonants=consonants+1;

#consonant counter is incremented by 1

print("The number of vowels:",vowels);

print("\nThe number of consonant:",consonants);

**Write a python program to accept input from user and check whether number is Armstrong or not.**

**#Python program to check if the number is an Armstrong number or not**

# take input from the user

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

# initialize sum

sum = 0

# find the sum of the cube of each digit

temp = num

while temp > 0:

digit = temp % 10

sum += digit \*\* 3

temp //= 10

# display the result

if num == sum:

print(num,"is an Armstrong number")

else:

print(num,"is not an Armstrong number")

**Write a Python program to print factorial of number using Recursion**

1. **def** recur\_factorial(n):
2. **if** n == 1:
3. **return** n
4. **else**:
5. **return** n\*recur\_factorial(n-1)
6. # take input from the user
7. num = int(input("Enter a number: "))
8. # check is the number is negative
9. **if** num < 0:
10. **print**("Sorry, factorial does not exist for negative numbers")
11. **elif** num == 0:
12. **print**("The factorial of 0 is 1")
13. **else**:
14. **print**("The factorial of",num,"is",recur\_factorial(num))
15. **Write a program to print length of String using Recursion. (15 marks)**

def length(str):

if str == "":

return 0

return 1 + length(str[1:])

str = "PrepInsta"

print("length of", str, "is", length(str))

1. **Write a python program to count the occurrence of each word in a given sentence (15 marks)**

# Define a function named word\_count that takes one argument, 'str'.

def word\_count(str):

# Create an empty dictionary named 'counts' to store word frequencies.

counts = dict()

# Split the input string 'str' into a list of words using spaces as separators and store it in the 'words' list.

words = str.split()

# Iterate through each word in the 'words' list.

for word in words:

# Check if the word is already in the 'counts' dictionary.

if word in counts:

# If the word is already in the dictionary, increment its frequency by 1.

counts[word] += 1

else:

# If the word is not in the dictionary, add it to the dictionary with a frequency of 1.

counts[word] = 1

# Return the 'counts' dictionary, which contains word frequencies.

return counts

# Call the word\_count function with an input sentence and print the results.

print( word\_count('the quick brown fox jumps over the lazy dog.'))

1. **Write a python program to print prime number between 1 to 100.**

# range function is not count last number (Ending number)

# only 1 to 100 is counted

for i in range(2,101):

for j in range(2,101):

if i%j == 0:

break

if i == j:

print(i,end=",")

**Write a python program to find the quadrants in which coordinates lies get the value of x and y coordinates as input from the user and check in which quadrants the point lies and print it**

# for initialization of coordinates

x, y = map(int, list(input("Insert the value for variable X and Y : ").split(" ")))

# find true condition of first quadrant

if x > 0 and y > 0:

print("point (", x, ",", y, ") lies in the First quadrant")

# find second quadrant

elif x < 0 and y > 0:

print("point (", x, ",", y, ") lies in the Second quadrant")

# To find third quadrant

elif x < 0 and y < 0:

print("point (", x, ",", y, ") lies in the Third quadrant")

# To find Fourth quadrant

elif x > 0 and y < 0:

print("point (", x, ",", y, ") lies in the Fourth quadrant")

# To find does not lie on origin

elif x == 0 and y == 0:

print("point (", x, ",", y, ") lies at the origin")

# On x-axis

elif y == 0 and x != 0:

print("point (", x, ",", y, ") on x-axis")

# On y-axis

elif x == 0 and y != 0:

print("point (", x, ",", y, ") on at y-axis")