```python
# 1. Write a Python program to check if a given number is a prime number. (Hint: A prime
number is only divisible by 1 and itself.)
"""
n=int(input("enter no "))
f=1
for i in range(2,n):
    if n%i==0:
        f=0
if f==1:
    print(n,"Prime")
else:
    print(n,"Not Prime")
# output:
# enter no 13
# 13 Prime
# enter no 12
# 12 Not Prime
"""

# 2. Create a program to determine if a number is an Armstrong number. (Hint: An Armstrong
number is equal to the sum of its digits raised to the power of the number of digits.)
'''
n=int(input("Enter no "))
c=0
n1=n2=n
while(n!=0):
    n=n//10
    c+=1
print(c)
sum=0
while(n1!=0):
    a=(n1%10)**c
    sum+=a
    n1=n1//10

if sum==n2:
    print('ArmStrong')
else:
    print("Not Armstrong")
'''
# output:
# Enter no 153
```

```python
# 3
# ArmStrong
# Sum 36
# Not Armstrong


# 3. Develop a Python program to check if a number is a Neon number. (Hint: A Neon
number's square has a digit sum equal to the number itself.)
"""
n=(int(input("enter no ")))
print(n)
n1=n**2
sum=0
while(n1!=0):
   a=n1%10
   sum+=a
   n1=n1//10
if sum==n:
   print('Neon Number ')
else:
   print("Not Neon Number ")
# output:
# enter no 9
# 9
# Neon Number
# enter no 3
# 3
# Not Neon Number
"""


# 4. Write a program to determine if a number is a Harshad number. (Hint: A Harshad number
is divisible by the sum of its digits.)
"""
n=int(input("enter no "))
n1=n
sum=0
while(n!=0):
   sum+=n%10
   n=n//10
if n1%sum==0:
   print("Harshad Number ")
```

```python
else:
    print("Not hardshad number ")
"""
# output
# enter no 23
# Not hardshad number
# enter no 12
# Harshad Number


# 5. Create a Python script to check if three given numbers form a Pythagorean triplet. (Hint:
Check if the square of one number equals the sum of the squares of the other two.)
"""
n1=int(input("enter n1 "))
n2=int(input("enter n2 "))
n3=int(input("enter n3 "))
#5->25
#4->16
#3->9

if (n1**2)==((n2**2)+(n3**2)):
    print("Pythagorean Triplet ")
else:
    print("Not ")
"""
# output:
# enter n1 5
# enter n2 4
# enter n3 3
# Pythagorean Triplet
# enter n1 3
# enter n2 4
# enter n3 5
# Not


# 6. Write a program to find the factorial of a given number. (Hint: The factorial of n is the
product of all positive integers less than or equal to n.)
# n=int(input("Enter no "))
# fact=1
# for i in range(1,n+1):
#     fact*=i
```

```python
# print("Factorial",fact)
# output:
# Enter no 5
# Factorial 120


# 7. Create a Python program to determine if a number is a perfect number. (Hint: A perfect
number equals the sum of its proper divisors, excluding itself.)
"""
n=int(input("Enter no "))
sum=0
for i in range(1,n):
   if n%i==0:
      sum+=i
if sum==n:
   print("Perfect number ")
else:
   print("Not Perfect ")
"""
# output:
# Enter no 12
# Not Perfect
# Enter no 6
# Perfect number


# 8. Write a Python script to generate all Fibonacci numbers up to a given number. (Hint: Each
number is the sum of the previous two numbers, starting with 0 and 1.)
# n=int(input("Enter no "))
# print(0,1,end=" ")
# n1=0
# n2=1

# for i in range(n-2):
#    c=n1+n2
#    print(c,end=" ")
#    n1=n2
#    n2=c
# output:
# 0 1 1 2 3
```

```python
# 9. Develop a Python program to check if a number is a strong number. (Hint: A strong
# number equals the sum of the factorials of its digits.)
# n=int(input("Enter No "))
# n1=n
# result=0
# while(n!=0):
#    a=n%10
#    fact=1
#    for i in range(1,a+1):
#        fact*=i
#    result+=fact
#    n=n//10
# if n1==result:
#    print("Perfect Number ")
# else:
#    print("Not Perfect Number ")
# output:
# Enter No 145
# Perfect Number
# Enter No 12
# Not Perfect Number


# 10. Write a program to check if a number is a palindrome. (Hint: A palindrome reads the
# same forwards and backwards.)
# n=int(input("enter no "))
# n1=n
# r=0
# # 121
# while(n!=0):
#    a=n%10
#    r=a+r*10
#    n=n//10
# print(r)
# if r==n1:
#    print("Palindrome ")
# else:
#    print("Not Palindrome")

# output:
# enter no 321
# Not Palindrome
```

```python
# enter no 121
# Palindrome


# 11. Create a Python program to find the greatest common divisor (GCD) of two numbers.
# (Hint: GCD is the largest number that divides both numbers without leaving a remainder.)
# n1=int(input("enter no1 "))
# n2=int(input("enter no2 "))
# if n1>n2:
#     n=n2
# else:
#     n=n1
# max=0
# for i in range(1,n+1):
#     if n1%i==0 and n2%i==0:
#         max=i
# print(max)
# enter no1 24
# enter no2 18
# 6
# enter no1 28
# enter no2 18
# 2
# enter no1 8
# enter no2 12
# 4


# 12. Write a Python program to check if a number is a perfect square. (Hint: A perfect square
# is the square of an integer.)
# import math
# n=int(input("Enter no "))
# if n%2==0:
#   if ((n//2))**2 == n:
#      print("Perfect Squre ")
#   else:
#      print("Not Perfect Squre")
# else:
#   if (n//2-1)**2==n:
#      print("Perfect Squre")
#   else:
#      print("Not ")
```

```python
# output:
# Enter no 3
# Not
# Enter no 9
# Perfect Squre


# # 13. Develop a Python script to print first 10 prime nos.
# for i in range(1,10):
#     p=1
#     for j in range(2,i):
#         if i%j==0:
#             p=0
#     if p==1:
#         print(i,'Prime')
# output:1 Prime
# 2 Prime
# 3 Prime
# 5 Prime
# 7 Prime


# 14. Create a program to generate all prime numbers within a given range. (Hint: Use a loop
# and check divisibility for each number in the range.)
# start=int(input("Enter starting range"))
# end=int(input("Enter ending range"))
# for i in range(start,end):
#     p=1
#     for j in range(2,i):
#         if i%j==0:
#             p=0
#     if p==1:
#         print(i,"Prime")
# output:
# Enter starting range1
# Enter ending range11
# 1 Prime
# 2 Prime
# 3 Prime
# 5 Prime
# 7 Prime
```

```python
# 15. Write a Python program to find the LCM (least common multiple) of two numbers. (Hint:
# LCM is the smallest positive number divisible by both numbers.)
# n1=int(input("Enter no1 "))
# n2=int(input("Enter no2 "))

# for i in range(2,(n1*n2)+1):
#     if i%n1==0 and i%n2==0:
#         print(i)
#         break
# output:
# Enter no1 15
# Enter no2 12
# 60
# Enter no1 4
# Enter no2 3
# 12
```