Q6) ARMSTRONG

def isarmstrong(num):

order=len(str(num))

sum=0

temp=num

while temp>0:

digit=temp%10

sum+=digit\*\*order

temp//=10

return num==sum

for num in range(1,1001):

if isarmstrong(num):

print(num)

Q7) PRIME

def prime(num):

if num<2:

return False

else:

for i in range(2,101):

if(num%i==0):

return False

else:

return True

for num in range(1,101):

if prime(num):

print(num)

Q10) LINEAR

import random

def linear(arr,x):

for i in range(len(arr)):

if arr[i]==x:

return i

return -1

arr=random.sample(range(1,100),20)

print("array:", arr)

x=int(input("enter no.:"))

result=linear(arr,x)

if result!=-1:

print("elemeny found at:",result)

else:

print("not found")

Q11) BINARY

def binary(arr,x):

low=0

high=len(arr)-1

while low<=high:

mid=(low+high)//2

if arr[mid]==x:

return mid

elif arr[mid]<x:

low=mid+1

else:

high=mid-1

return -1

arr=sorted(["banana","cherry","date","apple"])

print("sorted array:",arr)

x=input("enter element:")

result=binary(arr,x)

if result!= -1:

print("found at:",result)

else:

print("not found")

Q12). RANDOM MODULE

\*random\* :

random

randint

randrange

sample

choice

choices

seed

uniform

shuffle

gauss

Q13) MATH MODULE

\*math\* :

sqrt

cbrt

pow

ceil

floor

isclose

sin

cos

tan

factorial

log( base 10 and natural)

Q14) TYPES PF ARGUMENT

#positional argument

def greet(name, age):

print(f"Hello, {name}! You are {age} years old.")

# Calling the function with positional arguments

greet("Alice", 25)

#default argument

def greet(name, age=18): # 'age' has a default value of 18

print(f"Hello, {name}! You are {age} years old.")

# Calling the function with and without the age argument

greet("Alice")

greet("Bob", 25)

#keyword argument

def greet(name, age):

print(f"Hello, {name}! You are {age} years old.")

# Calling the function with keyword arguments

greet(age=25, name="Alice")

#arbitrary positional argument

def add(\*args):

total = sum(args)

print(f"Sum of values: {total}")

# Calling the function with a variable number of arguments

add(1, 2, 3) # Output: Sum of values: 6

add(4, 5, 6, 7, 8) #

#arbitrary keyword argument

def display\_info(\*\*kwargs):

for key, value in kwargs.items():

print(f"{key}: {value}")

# Calling the function with multiple keyword arguments

display\_info(name="Alice", age=25, city="New York")