```
#Program 1.
x=int(input("Enter a number"))
s=0
y=x
while(y!=0):
  d=y%10
  s=s+d**3
  y=y//10
if(s==x):
  print(x," is an armstrong no")
else:
  print(x," is not an armstrong no")
```

```
Enter a number153
153  is an armstrong no
```

```
#Program 2
def displayMinMax(l):
  if(len(l)==0):
    print("List is empty")
    return
  print("Maximum=",max(l))
  print("Minimum=",min(l))
def displaySecondMax(l):
  if(len(l)==0):
    print("List is empty")
    return
  if(len(l)<=2):
   print("List has too few elements (No of elements should be greater than 3")
   return
  l1=l.copy()
  l1.remove(max(l1))
  print("Second largest element",max(l1))
#Test with different cases here
displayMinMax([3,4,5])
displaySecondMax([3,4,5])
```

```
Maximum= 5
Minimum= 3
Second largest element 4
```

```
#Program 3
def binsearch(l,k):
  if(len(l)==0):
    print("list is empty")
    return
  l=sorted(l)
  print("Sorted array is:",l)
  b=0;e=len(l)-1
  while(b<=e):
    m=(b+e)//2
    if(l[m]==k):
```

```
        print("Element ",k," found at position:",m+1)
        return
      elif(k>l[m]):
        b=m+1
      else:
        e=m-1
    print("Element ",k," not found")
#Test with different cases here
binsearch([5,2,3,1,4],5)
```

```
     Sorted array is: [1, 2, 3, 4, 5]
     Element  5  found at position: 5
```

```
#program 4
#This logic has been developed so that the program can run in colab
#Simply create two files Fibonacci.py and Factorial.py seperately
#Store these files in the same location as the current file
#and then import these files into the current program using import
with open('Fibonacci.py','w') as file1:
  program1='def fibonacci(n):\n\
            f = [0, 1]\n\
            for i in range(2, n+1):\n\
            \tf.append(f[i-1] + f[i-2])\n\
            return f[n]'
  file1.write(program1)
with open('Factorial.py','w') as file2:
  program2='def factorial(n):\n\
            f=1\n\
            for i in range(1,n+1):\n\
            \tf=f*i\n\
            return f'
  file2.write(program2)
import Fibonacci
import Factorial
x=int(input("Enter a no."))
print(x,"th term of fibonacci series is:",Fibonacci.fibonacci(x))
print("Factorial of ",x," is:",Factorial.factorial(x))
```

```
     Enter a no.5
     5 th term of fibonacci series is: 5
     Factorial of  5  is: 120
```

```
#Program 5
def Length(s):
 return len(s)
def String(l):
  if(len(l)==0):
    print("List is empty")
    return
  l.sort(key=Length)
  t = [(i,len(i)) for i in l]
  return t
#Test with different cases here
print(String(['sdfgs','ghjkghjk','fgh']))
```

```
[('fgh', 3), ('sdfgs', 5), ('ghjkghjk', 8)]
```

```
#Program 6
def String(s):
  if(len(s)==0):
    print("Empty string")
    return
  l=[]
  #Find unique characters
  for i in s:
    if(i not in l):
      l.append(i)
  d={}
  for i in l:
    d[i]=s.count(i)
  return d
#Test with different cases here
print(String('asdfasdf'))
```

```
{'a': 2, 's': 2, 'd': 2, 'f': 2}
```

```
#Program 7
class Employee:
    basic=100000
    def addEmployee(self,EmpID,EmpName,EmpDesignation,Experience,Age):
        self.EmpID=EmpID
        self.EmpName=EmpName
        self.EmpDesignation=EmpDesignation
        self.Experience=Experience
        self.Age=Age
        self.salary=0
    def calculateSalary(self):
        salary=0
        if(self.Age<30 and self.Experience>5 and self.Experience<=10):
            salary=Employee.basic*1.5
        elif(self.Age<40 and self.Age>30 and self.Experience>5 and self.Experience<=10):
            salary=Employee.basic*1.75
        elif(self.Age<40 and self.Age>30 and self.Experience>10 and self.Experience<=20):
            salary=Employee.basic*2
        elif(self.Age<50 and self.Age>40 and self.Experience>20 and self.Experience<=25):
            salary=Employee.basic*2.25
        elif(self.Age<50 and self.Age>40 and self.Experience>25 and self.Experience<=30):
            salary=Employee.basic*2.5
        elif(self.Age<58 and self.Age>50 and self.Experience>30):
            salary=Employee.basic*3
        self.salary=salary
    def displayDetails(self):
        print("Employee Name=",self.EmpName)
        print("Employee ID=",self.EmpID)
        print("Designation=",self.EmpDesignation)
        print("Experience=",self.Experience)
        print("Age=",self.Age)
        print("Salary=",self.salary)
```

```
#Test with different cases here
o=Employee()
o.addEmployee('dsgf12','Ravi','Manager',6,25)
o.calculateSalary()
o.displayDetails()
```

```
 Employee Name= Ravi
 Employee ID= dsgf12
 Designation= Manager
 Experience= 6
 Age= 25
 Salary= 150000.0
```

```
#Program 8
import re
def Find(string):
    regex = r"(?i)\b((?:https?://|www\d{0,3}[.]|[a-z0-9.\-]+[.][a-z]{2,4}/)(?:[^\s()<>]+|\
    url = re.findall(regex,string)
    return [x[0] for x in url]
#Test with different cases here
string = 'My Profile: https://auth.geeksforgeeks.org/user/Chinmoy%20Lenka/articles in the
print("Urls: ", Find(string))
```

```
 Urls:  ['https://auth.geeksforgeeks.org/user/Chinmoy%20Lenka/articles', 'https://www
```

```
#Program 9.
import re
import sys
print("Enter 5 lines of text")
with open('Student Details.txt','w') as file:
    for i in range(5):
        file.write(input()+"\n")
with open('Student Details.txt','r') as file:
    s=file.read()
    l=re.split('[\n\s]',s)
    for i in range(l.count('')):
        l.remove('')
    min=sys.maxsize;maxi=0;mini=0
    max=-sys.maxsize
    for i in l:
        if(len(i)<min):
            min=len(i)
            mini=l.index(i)
        if(len(i)>max):
            max = len(i)
            maxi=l.index(i)
    print("Shortest word=",l[mini],".Length=",min)
    print("Longest word=", l[maxi],".Length=",max)
```

```
 Enter 5 lines of text
 hello bro
 my name
```

```
is asgfgsfsdfgsdfg
and i study at
afhgsdghdfgh
Shortest word= i .Length= 1
Longest word= asgfgsfsdfgsdfg .Length= 15
```

```
#Program 10
class BankAccount:
    def __init__(self,CustName,AccountNumber,Balance,TypeOfAccount,Address):
        self.CustName=CustName
        self.AccountNumber=AccountNumber
        self.Balance=Balance
        self.TypeOfAccount=TypeOfAccount
        self.Address=Address
    def withdraw(self,money):
        if(self.Balance-money<0):
            raise Exception('Insufficient balance.Transaction failed')
        self.Balance=self.Balance-money
    def deposit(self,money):
        if(money<0):
          raise Exception("Negative amount cannot be deposited")
        else:
          self.Balance+=money
    def displayDetails(self):
        print("Customer Name=",self.CustName)
        print("Account Number=",self.AccountNumber)
        print("Balance=",self.Balance)
        print("Type Of Account=",self.TypeOfAccount)
        print("Address=",self.Address)

#Test case with no errors. Use withdraw or deposit error to invoke exception
obj1=BankAccount("DKS",'BF33',100000000,'Deposit','Bangalore')
obj1.withdraw(10000000)
obj1.deposit(1000)
obj1.displayDetails()
```

```
Customer Name= DKS
Account Number= BF33
Balance= 90001000
Type Of Account= Deposit
Address= Bangalore
```

✓ 0s completed at 12:18 PM