#python

1 .wap inputting your name which will be printed in the reverse order String manipulation

```
  string = input("enter string ")
string2 = ''
for i in string:
 string2 = i+string2
 print("string = ", string)
print("Reversed String=", string2)
```

output-

```
string =  anand
Reversed String= dnana
```

2. Create a Calculator program using four functions

```
def add(x, y):
   return x + y
def subtract(x, y):
   return x - y
def multiply(x, y):
   return x * y
def divide(x, y):
   try:
      return x / y
   except:
      print("invalid input",y)


print("Select operation:")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")


c  = input("Enter choice(1/2/3/4): ")
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))


if c == '1':
   print(num1,"+",num2,"=", add(num1,num2))
```

```
elif c == '2':
   print(num1,"-",num2,"=", subtract(num1,num2))

elif c == '3':
   print(num1,"*",num2,"=", multiply(num1,num2))

elif c == '4':
   print(num1,"/",num2,"=", divide(num1,num2))
else:
   print("Invalid input")
```

3. List Manipulation program identify Even & Odd Number separately

```
def EvenOdd(a):


    n = int(input("Enter number of elements:"))
     for i in range(1, n + 1):
         b = int(input("Enter element:"))
         a.append(b)
    even = []
    odd = []
    dict = {}
    evenadd = 0
    oddadd = 0
    evencount = 0
    oddcount = 0
    for j in a:
        if (j % 2 == 0):
            even.append(j)
            evenadd += j
            evencount += 1
        else:
            odd.append(j)
            oddadd += j
            oddcount = oddcount + 1
    print("The even list", even)
```

```
print("The odd list", odd)

dict["even"] = even

dict["odd"] = odd

dict["evenAddition"] = evenadd

dict["oddAdditoin"] = oddadd

dict["Ecount"] = evencount

dict["Ocount"] = oddcount

print(dict)
```

4. Inheritance Assignment

```
class Employee:
    def __init__(self,name,id,age):
        self.name = name
        self.id = id
        self.age = age



class Course:
    def __init__(self,cid,cname,duration):
        self.cid = cid
        self.coursename = cname
        self.duration = duration#



class Trainee(Employee,Course):
    global d
    d={}
    global l
    l = []
    def __init__(self,tid,course,dict):
        self.tid = tid
        self.course = course
        self.dict = dict
```

```
    def add(self,selfC):
        l.append(selfC.coursename)
        return selfC.coursename
    def update(self,selfc,cname):
        selfc.coursename = cname
        l.append(selfc.coursename)
    def addstatus(self,course,status):
        d[course] = status
    def display(self):
        print(d)


t = Trainee("xyz","python",{'cpp':"complete"})
c = Course(1,"CPP",55)
c2 = Course(2, "Java", 45)
c3 = Course(3, "Electronic", 69)
c4 = Course(4, "DBA", 40)
c5 = Course(5, "Cdsa", 78)
#t.add(c2)
t.update(c2,"python")
t.addstatus(t.add(c4),"incomplete")
t.addstatus(t.add(c5),"complete")
t.display()
```