Practical N0.11:

Program to demonstrate creating and importing package.

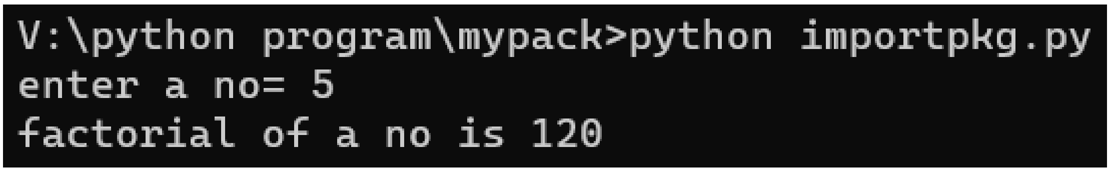
#fact.py

```
def fact(no):
    fact=1
    for i in range(1,no+1):
        fact=fact*i
    print("factorial of a no is",fact)
```

#importpkg.py

```
import mypkg.fact as s
no=int(input("enter a no"))
s.fact(no)
```

OUTPUT:



```
V:\python program\mypack>python importpkg.py
enter a no= 5
factorial of a no is 120
```

Practical No.12 :

Write a program to demonstrate hierarchical inheritance.

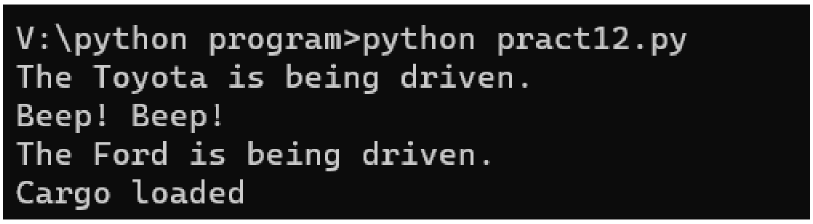
```
class vehicle:
    def __init__(self,brand):
        self.brand = brand


    def drive(self):
        print(f"The {self.brand} is being driven.")


class car(vehicle):
    def honk(self):
        print("Beep! Beep!")


class truck(vehicle):
    def load_cargo(self):
        print("Cargo loaded")
car = car("Toyota")
car.drive()
car.honk()
truck = truck("Ford")
truck.drive()
truck.load_cargo()
```

OUTPUT:



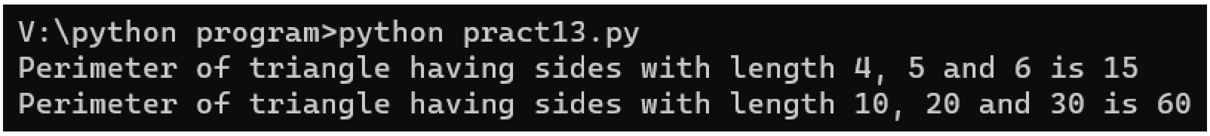
```
V:\python program>python pract12.py
The Toyota is being driven.
Beep! Beep!
The Ford is being driven.
Cargo loaded
```

Practical N0.13:

Program to demonstrate binary operator overloading.

```
class peri_trgl:
    def __init__ (s,a,b,c):
        s.a,s.b,s.c=a,b,c
        def __add__(s,other):
            s.a = s.a + other.a
            s.b = s.b + other.b
            s.c = s.c + c
    def show(s):
                print("Perimeter of triangle having sides with length %d, %d and %d is %d"%(s.a,s.b,s.c,s.a+s.b+s.c))
obj = peri_trgl(4,5,6)
obj2 = peri_trgl(10,20,30)
obj.show()
obj2.show()
```

output:



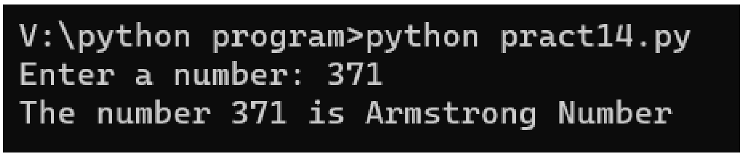
```
V:\python program>python pract13.py
Perimeter of triangle having sides with length 4, 5 and 6 is 15
Perimeter of triangle having sides with length 10, 20 and 30 is 60
```

Practical No.14:

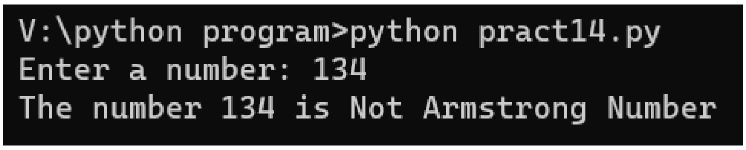
Program to demonstrate use of lambda function with map and reduce functions.

```
import functools
num = int(input("Enter a number: "))
lst = [int(x) for x in str(num)]
lst = list(map(lambda x: x**3, lst))
res = functools.reduce(lambda x, y: x + y, lst)
check = lambda x, y: "Armstrong Number" if x == y else "Not Armstrong Number"
print("The number %d is %s" % (num,check(res,num)))
```

OUTPUT:



```
V:\python program>python pract14.py
Enter a number: 371
The number 371 is Armstrong Number
```



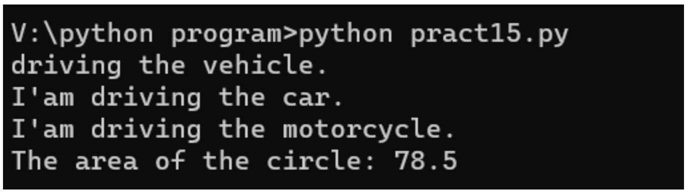
```
V:\python program>python pract14.py
Enter a number: 134
The number 134 is Not Armstrong Number
```

Practical NO.15:

Write a program to demonstrate method overriding.

```
class vehicle:
    def drive(self):
        print("driving the vehicle.")
 class car(vehicle):
    def drive(self):
        print("I'am driving the car.")
 class motorcycle(vehicle):
    def drive(self):
        print("I'am driving the motorcycle.")
 class shape:
    def calculate_area(self):
        print("")
 class circle(shape):
    def __init__(self, radius):
        self.radius = radius
    def calculate_area(self):
        area = 3.14 * self.radius ** 2
        print("The area of the circle:", area)
 vehicle = vehicle()
car = car()
motorcycle = motorcycle()
circle = circle()
vehicle.drive()
car.drive()
motorcycle.drive()
circle.calculate_area()
```

OUTPUT:



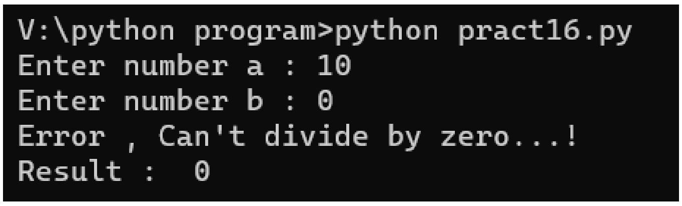
```
V:\python program>python pract15.py
driving the vehicle.
I'am driving the car.
I'am driving the motorcycle.
The area of the circle: 78.5
```

Practical No.16

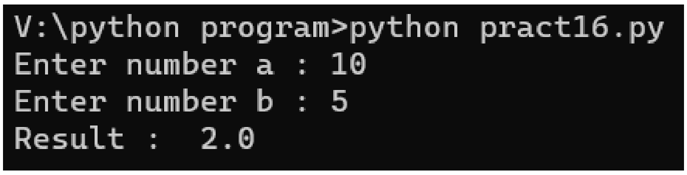
Write a program to demonstrate exception handling using raise, try, except and finally statements.

```
a = int(input("Enter number a : "))
b = int(input("Enter number b : "))
c = 0
try:
    c = a/b
    if b == 0:
        raise ZeroDivisionError
except ZeroDivisionError:
        print("Error , Can't divide by zero...!")
finally:
    print("Result : ",c)
```

OUTPUT:



```
V:\python program>python pract16.py
Enter number a : 10
Enter number b : 0
Error , Can't divide by zero...!
Result :  0
```



```
V:\python program>python pract16.py
Enter number a : 10
Enter number b : 5
Result :  2.0
```

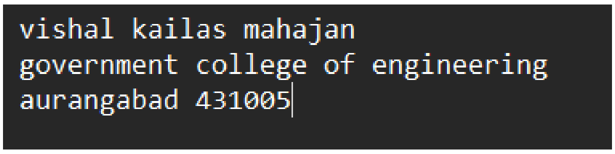
Practical No.17

Write a function in Python to capitalize first letter of every word in a text file

```
def capitalize():
    try:
        fi=open("sample.txt","r")
        fo=open("output.txt","w")
        for i in fi:
            strng=""
            for st in i.split():
                st=st.capitalize()
                strng=strng + st + " "
            fo.write(strng)
            fo.write("\n")
        fo.close()
    except FileNotFoundError:
        print("File does not exist!!!")
    except IOError:
        print("Unable to read a file")
capitalize()
```

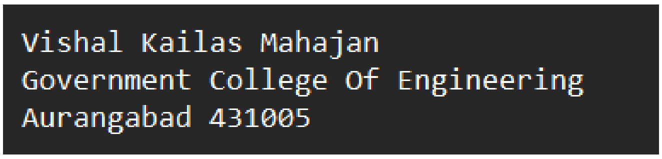
output:

sample.txt



```
vishal kailas mahajan
government college of engineering
aurangabad 431005
```

Output.txt



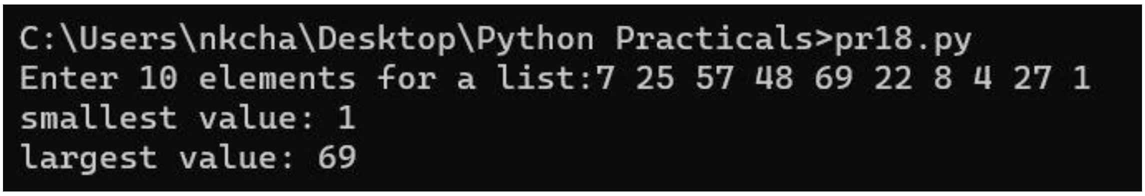
```
Vishal Kailas Mahajan
Government College Of Engineering
Aurangabad 431005
```

Practical No.18:

Title : Create a list of 10 elements and write a program to write this list in binary file and then read it back to find out the smallest and largest value

```
import pickle
string=input("Enter 10 elements for a list:").split()
lst=[]
for i in string:
   lst.append(int(i))
with open("binary.dat","wb")as s:
   pickle.dump(lst,s)
with open("binary.dat","rb")as fo:
   loaded_list=pickle.load(fo)
print("smallest value:",min(loaded_list))
print("largest value:",max(loaded_list))
```

OUTPUT:



```
C:\Users\nkcha\Desktop\Python Practicals>pr18.py
Enter 10 elements for a list:7 25 57 48 69 22 8 4 27 1
smallest value: 1
largest value: 69
```