

1A hundred years old:

```
name=input("Enter your name:")
age=int(input("Enter your Age:"))
sum=(2023-age)+100
print(name,"you will be 100 years old in year:", sum)
```

1b even odd:

```
num=int(input("Enter a number:"))
if num%2==0:
    print("The number is even")
else:
    print("The number is odd")
```

1c fibonacci series

```
v=int(input("Enter the number of series:"))
a=0
b=1
for n in range(0,v):
    if n<=1:
        c=n
    else:
        c=a+b
        a=b
        b=c
    print(c)
```

1d reverse number

```
def revnum(num):
    sum=0
    while num!=0:
        rem=num%10
        sum=rem+sum*10
```

```
    num//=10
    print("reverse number is",sum)
num=int(input("enter number:"))
revnum(num)
```

1e Armstrong and palindrome

```
def armnum(num):
    sum=0
    temp=num
    while temp>0:
        digit=temp%10
        sum+=digit ** 3
        temp//=10
    if num==sum:
        print(num,"is an armstrong number:")
    else:
        print(num,"is not an armstrong number:")

def palnum(num):
    sum=0
    temp=num
    while num!=0:
        rem=num%10
        sum=rem+sum*10
        num//=10
    if temp==sum:
        print(temp,"is a palindrome number:")
    else:
        print(temp,"is not a palindrome number:")

num=int(input("Enter a num:"))
```

```
armnum(num)
```

```
palnum(num)
```

1f factorial with recursive

```
def fact(x):
```

```
    if x==1:
```

```
        return 1
```

```
    else:
```

```
        return x*fact(x-1)
```

```
x=int(input("Enter a number:"))
```

```
print("The factorial of", x , "is", fact(x))
```

2a vowel or consonant:

```
def vchk(ch):
```

```
    if(ch=='a' or ch=='A' or ch=='e' or ch=='E' or ch=='i' or ch=='I' or ch=='o' or ch=='O' or  
ch=='u' or ch=='U'):
```

```
        print(ch,"is vowel")
```

```
    else:
```

```
        print(ch,"is consonant")
```

```
ch=input("Enter a single character(a-z/A-Z) only:")
```

```
vchk(ch)
```

2b count length list or string

```
def calen(n):
```

```
    count=0
```

```
    for i in n:
```

```
        count=count+1
```

```
    return count
```

```
print(calen("Student"))
```

2c histogram:

```
def histogram(items):
```

```
    for n in items:
```

```
output=' '  
times=n  
while(times>0):  
    output+='*'  
    times=times-1  
print(output)  
histogram([4,9,7])
```

3a check to sentence it is program

```
import string, sys  
def ispangram(str1, alphabet=string.ascii_lowercase):  
    alphaset=set(alphabet)  
    return alphaset<=set(str1.lower())  
print(ispangram("The quick brown fox jumps over a lazy dog"))
```

3b print list elemets less than 5

```
a=[1,1,2,3,5,8,13,21,34,55,89]  
for i in a:  
    if i<5:  
        print(i)
```

4a take 2 list return true if anyone common

```
def find_common(st1,st2):  
    res=False  
    for x in st1:  
        for y in st2:  
            if x==y:  
                res=True  
    return res
```

```
print(find_common([4,6,7],[4,3,13]))
print(find_common([10,9,8],[5,6,14]))
```

4b list after removing specified elem pract 4b

```
name=['yashu','vedu','somu','pari','sonu','monu']
name=[x for (i,x) in enumerate(name) if i not in(0,2,4,5)]
print(name)
```

4c copy list

```
L1=[8,1,2,3,5]
L2=list(L1)
print("L1: ",L1)
print("L2: ",L2)
```

5a dictionary ascending and descending

```
import operator
d={1:22, 3:13, 4:8, 2:11, 0:27}
print(d)
t=sorted(d.items(), key=operator.itemgetter(0))
print("In ascending order by value:", t)
t=t=sorted(d.items(), key=operator.itemgetter(0), reverse=True)
print("In descending order by value:", t)
```

5b concatenate dictionary

```
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50, 6:60}
print(dic1)
dic1.update(dic2)
```

```
dic1.update(dic3)
print(dic1)
```

5c sum of all dic items prac 5c

```
d={'t1':10, 't2':20, 't3':30}
print(d)
print("sum:",sum(d.values()))
```

6a read entire text file prac 6a

```
f=open('abc.txt','r')
t=f.read()
print(t)
f.close()
```

6b append text file

```
f=open('abc.txt','a+')
f.write('\n Easy to learn')
f=open('abc.txt','r')
t=f.read()
print(t)
f.close()
```

6c read last n lines of a file

```
f=open('abc.txt','r')
t=f.readlines()
print(t[-1])
f.close
```

7a class stores the info of students and display

```
class student:
    def __init__(self,name,address,mobile,email):
        self.name=name
        self.address=address
        self.mobile=mobile
        self.email=email
    def display(self):
        print("Name:",name)
        print("Address:",address)
        print("Mobile:",mobile)
        print("Email:",email)
```

```
print("Enter your details:")
name=input("Enter your name:")
address=input("Enter your Address:")
mobile=input("Enter your Mobile:")
email=input("Enter your Email:")
s1=student(name,address,mobile,email)
s1.display()
```

7b inheritance using python

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

```
def getInfo(self):
```

```
return (self.name+" "+str(self.age))
```

```
def isEmployee(person):
```

```
    return False
```

```
class Employee(person):
```

```
    def isEmployee(self):
```

```
        return True
```

```
emp=person("Piyush",10)
```

```
print(emp.getInfo(), emp.isEmployee())
```

```
emp=Employee("Pallavi", 20)
```

```
print(emp.getInfo(), emp.isEmployee())
```

7c MULTIPLIER

```
class numbers:
```

```
    MULTIPLIER=3.5
```

```
    def __init__(self,x,y):
```

```
        self.x=x
```

```
        self.y=y
```

```
    def add(self):
```

```
        return self.x + self.y
```

```
    @classmethod
```

```
    def multiply(cls,a):
```

```
        return cls.MULTIPLIER*a
```

```
@staticmethod
def subtract(b,c):
    return b-c
```

```
@property
def value(self):
    return(self.x, self.y)
```

```
#setter
def set_value(self,x,y):
    self.x=x
    self.y=y
```

```
#deleter
def del_value(self):
    del self.x
    del self.y
```

```
obj1=numbers(10,20)
print("add() output=", obj1.add())
print("multiply() output=", numbers.multiply(10))
print("subtract() output=", numbers.subtract(10,5))
print("property output=", obj1.value())
print(obj1.set_value(100,200))
print("property output=", obj1.value())
print(obj1.del_value())
print("property output=", obj1.value())
```

8a geometry square & circle

```
import math

def square_area(x):
    return(x*x)

def circle_area(y):
    return (math.pi*r*r)

import geometry

def pointyShapeVolume(x,y,squareBase):
    if squareBase==True:
        print("square area=",geometry.square_area(x))
    else:
        print("circle area=",geometry.circle_area(x))

print(dir(geometry))

pointyShapeVolume(2,3,True)
pointyShapeVolume(2,4,False)
```

8b exception handling

```
try:
    number=int(input("Enter a number between 1-10 ="))
    r=100/number
except(ValueError):
    print("Please enter number only:")
except(ZeroDivisionError):
    print("Please enter number greater than 0")
else:
    print("Result:",r)
finally:
    print("You are in finally block:")
```