Practical: 1

Name: Poonam Satalkar
Roll No: 27 (B)

Que: WAP to display all types of pyramids of star.

1]

```
rows = int(input("Enter number of rows: "))
for i in range(rows):
    for j in range(i+1):
        print("* ", end="")
    print("\n")
```

Output:

```
Enter number of rows: 6
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

2]

```
for i in range(5):
    for j in range(5):
        print("*", end="")
    print()
```

Output:

```
*****
*****
*****
*****
*****
```

3]

```
for i in range(5):
    for j in range(i):
        print(" ", end="")
    for j in range(i, 5):
        print("* ", end="")
    print()
```

Output:

```
* * * * *
 * * * *
  * * *
   * *
    *
```

4]

```
for i in range(5):
    for j in range(5, i, -1):
        print(" ", end="")
    for k in range(i + 1):
        print("* ", end="")
    print()
```

Output:

```
     *
    * *
   * * *
  * * * *
 * * * * *
```

5]

```
n = 5
# Upper half of the diamond
for i in range(n):
    for j in range(n - 1, i, -1):
        print(" ", end="")
    for k in range(i + 1):
        print("* ", end="")
    print()
# Lower half of the diamond
for i in range(1, n):
    for j in range(i):
        print(" ", end="")
    for k in range(n - 1, i - 1, -1):
        print("* ", end="")
    print()
```

Output:

```
    *
   * *
  * * *
 * * * *
* * * * *
 * * * *
  * * *
   * *
    *
```

Practical 2

Name: Poonam Satalkar
Roll No: 27 (B)

Que : Write program to display multiplication of all numbers from 1 to 10.

```
for i in range(1,11):
    print("\n\nMULTIPLICATION TABLE FOR %d\n" %(i))
    for j in range(1,11):
        print("%-5d X %5d = %5d" % (i, j, i*j))
```

Output:

```
MULTIPLICATION TABLE FOR 1
1       X       1 =       1
1       X       2 =       2
1       X       3 =       3
1       X       4 =       4
1       X       5 =       5
1       X       6 =       6
1       X       7 =       7
1       X       8 =       8
1       X       9 =       9
1       X      10 =      10
MULTIPLICATION TABLE FOR 2
2       X       1 =       2
2       X       2 =       4
2       X       3 =       6
2       X       4 =       8
2       X       5 =      10
2       X       6 =      12
2       X       7 =      14
2       X       8 =      16
2       X       9 =      18
2       X      10 =      20
MULTIPLICATION TABLE FOR 3
3       X       1 =       3
3       X       2 =       6
3       X       3 =       9
3       X       4 =      12
3       X       5 =      15
3       X       6 =      18
3       X       7 =      21
3       X       8 =      24
3       X       9 =      27
3       X      10 =      30
MULTIPLICATION TABLE FOR 4
4       X       1 =       4
4       X       2 =       8
```

4	X	3 =	12
4	X	4 =	16
4	X	5 =	20
4	X	6 =	24
4	X	7 =	28
4	X	8 =	32
4	X	9 =	36
4	X	10 =	40

MULTIPLICATION TABLE FOR

5	X	1 =	5
5	X	2 =	10
5	X	3 =	15
5	X	4 =	20
5	X	5 =	25
5	X	6 =	30
5	X	7 =	35
5	X	8 =	40
5	X	9 =	45
5	X	10 =	50

MULTIPLICATION TABLE FOR 6

6	X	1 =	6
6	X	2 =	12
6	X	3 =	18
6	X	4 =	24
6	X	5 =	30
6	X	6 =	36
6	X	7 =	42
6	X	8 =	48
6	X	9 =	54
6	X	10 =	60

MULTIPLICATION TABLE FOR 7

7	X	1 =	7
7	X	2 =	14
7	X	3 =	21
7	X	4 =	28
7	X	5 =	35
7	X	6 =	42
7	X	7 =	49
7	X	8 =	56
7	X	9 =	63
7	X	10 =	70

MULTIPLICATION TABLE FOR 8

8	X	1 =	8
8	X	2 =	16
8	X	3 =	24
8	X	4 =	32

8	X	5 =	40
8	X	6 =	48
8	X	7 =	56
8	X	8 =	64
8	X	9 =	72
8	X	10 =	80

MULTIPLICATION TABLE FOR 9

9	X	1 =	9
9	X	2 =	18
9	X	3 =	27
9	X	4 =	36
9	X	5 =	45
9	X	6 =	54
9	X	7 =	63
9	X	8 =	72
9	X	9 =	81
9	X	10 =	90

MULTIPLICATION TABLE FOR 10

10	X	1 =	10
10	X	2 =	20
10	X	3 =	30
10	X	4 =	40
10	X	5 =	50
10	X	6 =	60
10	X	7 =	70
10	X	8 =	80
10	X	9 =	90
10	X	10 =	100

Practical 3

Name: Poonam Satalkar
Roll No: 27 (B)

Que : WAP to implement tower of Hanoi

```
def TowerOfHanoi(n , source, destination, auxiliary):
    if n==1:
        print ("Move disk 1 from source",source,"to destination",destination)
        return
    TowerOfHanoi(n-1, source, auxiliary, destination)
    print ("Move disk",n,"from source",source,"to destination",destination)
    TowerOfHanoi(n-1, auxiliary, destination, source)
n = 4
TowerOfHanoi(n,'A','B','C')
```

Output:

```
Move disk 1 from source A to destination C
Move disk 2 from source A to destination B
Move disk 1 from source C to destination B
Move disk 3 from source A to destination C
Move disk 1 from source B to destination A
Move disk 2 from source B to destination C
Move disk 1 from source A to destination C
Move disk 4 from source A to destination B
Move disk 1 from source C to destination B
Move disk 2 from source C to destination A
Move disk 1 from source B to destination A
Move disk 3 from source C to destination B
Move disk 1 from source A to destination C
Move disk 2 from source A to destination B
Move disk 1 from source C to destination B
```

Practical 4

Name: Poonam Satalkar
Roll No: 27 (B)

Que : Wap to calculate simple interest using user defined function .accept amount,duration from user .
set interest rate as default parameter.

```
def simpleint(p,n,r):
    si=(p*n*r)/100
    return si
p=float(input("Enter Principle amount : "))
n=float(input("Enter number of years : "))
r=float(input("Enter rate of interest : "))
si=simpleint(p,n,r)
print("simple interest :{}".format(si))
```

Output:

```
Enter Principle amount : 2000
Enter number of years : 5
Enter rate of interest : 0.5
simple interest : 50.0
```

Practical 5

Name: Poonam Satalkar
Roll No: 27 (B)

Que : Wap to print even and odd number in the list

```
x=[10,3,9,6,4,6,8,5]
even_count,odd_count =0,0
for num in x:
    if num % 2 == 0:
        even_count += 1
    else:
        odd_count += 1
print("Even numbers in the list: ", even_count)
print("Odd numbers in the list: ", odd_count)
```

Output:

```
Even numbers in the list:   5
Odd numbers in the list:   3
```

Practical 6

Name: Poonam Satalkar
Roll No: 27 (B)

Que : wap to find out the sum of all numbers,min,max mean median,mode of numbers in a list.

```
ls=list(range(1,21))
print("list is :",ls)
sum=0
for i in ls:
    sum+=i
print("sum of all numbers",sum)
print("Maximum numbers",max(ls))
print("Minimum numbers",min(ls))
```

Output:

```
list is : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
sum of all numbers 210
Maximum numbers 20
Minimum numbers 1
```

```
# Mean
num=[1,2,3,4,5,6,7,7]
n=len(num)
get_sum=sum/n
mean=get_sum/n
print ("mean /avg is :" +str(mean))
#median
num.sort()
if n%2==0:
    median1=num[n//2]
    median2=num[n//2-1]
    median=(median1+median2)/2
else:
    median=num[n//2]
print ("median is :" +str(median))
#mode
from collections import Counter
data=Counter(num)
get_mode=dict(data)
mode=[k for k,v in get_mode.items() if v==max(list(data.values()))]
if len(mode)==n:
    get_mode="No mode found"
else:
    get_mode="mode is :"+','.join(map(str,mode))
print(get_mode)
```

Output:

```
mean /avg is :3.28125
median is :4.5
mode is :7
```

Practical 7

Name: Poonam Satalkar
Roll No: 27 (B)

Que : wap to find out the sum and multiplication of two matrices implemented using list

```
X = [[12,7,3],
     [4 ,5,6],
     [7 ,8,9]]
Y = [[5,8,1],
     [6,7,3],
     [4,5,9]]
result = [[0,0,0],
          [0,0,0],
          [0,0,0]]
print("addition is")
for i in range(len(X)):
    for j in range(len(Y[0])):
        for k in range(len(Y)):
            result[i][j] += X[i][k] + Y[k][j]
for r in result:
    print(r)

print("multiplication is")
for i in range(len(X)):
    for j in range(len(Y[0])):
        for k in range(len(Y)):
            result[i][j] += X[i][k] * Y[k][j]
for r in result:
    print(r)
```

Output:

```
addition is
[37, 42, 35]
[30, 35, 28]
[39, 44, 37]
multiplication is
[151, 202, 95]
[104, 132, 101]
[158, 201, 149]
```

Practical 8

Name: Poonam Satalkar
Roll No: 27 (B)

Que : Wap to store the student roll number and marks using dictionary.
Implements following functions.
Add a records ,delete ,update marks, search a roll number and display marks,
sort the records in assending order and dessending order,display student information with highest marks implement a menu driven program.

```
d={1:56,4:47,5:88,2:99}
print("Student Info =",d)
def create():
    d1=eval(input("creating new element {}"))
    print("Old record=",d)
    print("New record=",d1)
def add_element():
    k=int(input("enter roll number of student="))
    v=int(input("enter marks of student="))
    d.update({k:v})
    print("after add",d)
def display():
    for k in d:
        print("Roll No:",k,"Marks :",d[k])
def update():
     num=int(input("enter total number student="))
     for i in range(num):
        k=int(input("enter roll number of student="))
        v=int(input("enter marks of student="))
        d.update({k:v})
        print("after update",d)
def delete():
    num=int(input("enter roll number for delete   student info="))
    d.pop(num)
    print("After delete ",d)
    print("Record delete successfully..")
def search():
    r=int(input("enter roll number to search marks="))
    marks=d.get(r,-1)
    if( marks!=-1):
        print("Roll number",r,"marks",marks)
    else:
        print("record is not found..")
def sort_asc():
    print("Sort by ascending= ",sorted(d.items(),key=lambda kv:([1],kv[0])))
def sort_dec():
    print("Sort by descending= ",sorted(d.items(),key=lambda kv:([1],kv[0]),reverse=True))
```

```
def d_max():
    print("Maximum marks=",max(d.items(),key=lambda kv:(kv[1],kv[0])))
def d_min():
    print("Minimum marks=",min(d.items(),key=lambda kv:(kv[1],kv[0])))
def d_sum():
    values=d.values()
    total=sum(values)
    print("Sum of marks=",total)
def d_avg():
    r=[v for v in d.items() if v!=0]
    avg=sum(r)/len(r)
    print("average of =",avg)

choice=''
while choice!='13':
    print("\n 1]Create record")
    print(" 2]Add record")
    print("3]Display student information ")
    print("4]update marks")
    print("5]delete records ")
    print("6]Search roll number and display records")
    print("7]sort by asseding order")
    print("8]sort by desending order ")
    print("9]maximum marks ")
    print("10]minimum marks")
    print("11]sum of marks")
    print("12]average marks")
    print("13]Exit ")

    choice =input("\nSelect Option")
    if choice=='1':
        create()
    elif choice=='2':
        add_element()
    elif choice=='3':
        display()
    elif choice=='4':
        update()
    elif choice=='5':
        delete()
    elif choice=='6':
        search()
    elif choice=='7':
        sort_asc()
    elif choice=='8':
        sort_dec()
    elif choice=='9':
        d_max()
    elif choice=='10':
        d_min()
    elif choice=='11':
        d_sum()
    elif choice=='12':
        d_avg()
    elif choice=='13':
        exit()
    else:
        print("Option not Found")
```

Output:

```
Student Info = {1: 56, 4: 47, 5: 88, 2: 99}

 1]Create record
 2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option2
enter roll number of student=3
enter marks of student=75
after add {1: 56, 4: 47, 5: 88, 2: 99, 3: 75}

 1]Create record
 2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
```

```
12]average marks
13]Exit

Select Option3
Roll No: 1 Marks : 56
Roll No: 4 Marks : 47
Roll No: 5 Marks : 88
Roll No: 2 Marks : 99
Roll No: 3 Marks : 75

  1]Create record
  2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option4
enter total number student=1
enter roll number of student=5
enter marks of student=85
after update {1: 56, 4: 47, 5: 85, 2: 99, 3: 75}

  1]Create record
  2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option5
enter roll number for delete   student info=2
```

```
After delete   {1: 56, 4: 47, 5: 85, 3: 75}
Record delete successfully..

  1]Create record
  2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option6
enter roll number to search marks=3
Roll number 3 marks 75

  1]Create record
  2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option7
Sort by ascending=   [(1, 56), (3, 75), (4, 47), (5, 85)]

  1]Create record
  2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
```

```
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option8
Sort by descending=   [(5, 85), (4, 47), (3, 75), (1, 56)]

  1]Create record
  2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option9
Maximum marks= (5, 85)

  1]Create record
  2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option10
Minimum marks= (4, 47)

  1]Create record
  2]Add record
3]Display student information
```

```
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option11
Sum of marks= 263

  1]Create record
  2]Add record
3]Display student information
4]update marks
5]delete records
6]Search roll number and display records
7]sort by asseding order
8]sort by desending order
9]maximum marks
10]minimum marks
11]sum of marks
12]average marks
13]Exit

Select Option13
```

Practical 9

Name: Poonam Satalkar
Roll No: 27 (B)

Que : wap to implement function decorator to display cube of a number.

```
def deco(fun):
    def inside():
        value=fun()
        return value **3
    return inside
@deco
def num():
    return 7
print("cube of 7 is ",num())
```

Output:

```
cube of 7 is   343
```

Practical 10

Name: Poonam Satalkar
Roll No: 27 (B)

Que : wap to implement generator function to display square of a number from 1 to 10

```
def gensq(x,y):
    while(x<=y):
        yield x*x
        x=x+1
g=gensq(1,10)

for i in g:
    print("\n", i,end='')
```

Output:

```
1
4
9
16
25
36
49
64
81
100
```

Practical 11

Name: Poonam Satalkar
Roll No: 27 (B)

Que : wap program to implement package and module .package- Employeementmgmt
Module salary -function to calculate gross and net salary.
Module empInfo -function to display emp information i.e.
name,designation,dept,qualification,experience.s

```
print("SALARY PROGRAM")
name=str(input("Enter name of employee"))
basic=float(input("Enter basic salary"))
da=float(basic*0.25)
hra=float(basic*0.15)
pf=float((basic+da)*0.12)
ta=float(basic*0.075)
netpay=float(basic+da+hra+ta)
grosspay=float(nettpay-pf)

print("\n\n")
print("SALARY DETAILED BREAKUP")
print("---------------------------------------------------------")
print(" NAME OF EMPLOYEE :", name)
print("Desiganation:, Database Engineer")
print("Qualifications : M.Tech Computer science ")
print("Experience :,3 Years")
print("BASIC SALARY",basic)
print("DEARNESS ALLOW",da)
print("HOUSE RENT ALLOW",hra)
print("TRAVEL ALLOW",ta)
print("---------------------------------------------------------")
print("NET SALARY PAY",netpay)
print("PROVIDENT FUND",pf)
print("---------------------------------------------------------")
print("GROSS PAYMENT",grosspay)
print("---------------------------------------------------------")
```

Output:

```
SALARY PROGRAM
Enter name of employeexyz
Enter basic salary3452

SALARY DETAILED BREAKUP
---------------------------------------------------------
 NAME OF EMPLOYEE : xyz
Desiganation:, Database Engineer
```

```
Qualifications : M.Tech Computer science
Experience :,3 Years
BASIC SALARY 3452.0
DEARNESS ALLOW 863.0
HOUSE RENT ALLOW 517.8
TRAVEL ALLOW 258.9
---------------------------------------------------------
NET SALARY PAY 5091.7
PROVIDENT FUND 517.8
---------------------------------------------------------
GROSS PAYMENT 4573.9
---------------------------------------------------------
```

Practical 12

Name: Poonam Satalkar
Roll No: 27 (B)

Que : Wap to implement a class to store student information as id,name,marks.
implements all class, instance, public .private attributes.
implements instance,class ,constructor,destructor,getter and setter methods.

```
class Student:
    counter=0
    classname="MCA1"
    def __init__(self,r,n):
        self.rollno=r
        self.name=n
        Student.counter+=1
    def display(self):
        print("ROll number :",self.rollno)
        print("Name :",self.name)

    #setter method
    def set_name(self,name):
        self.name=name

    #getter method
    def get_name(self):
        return self.name
    @classmethod
    def displaytotal(cls):
        print("Total Student :",Student.counter)

    @staticmethod
    def dispclass():
        print("Student class name is ",Student.classname)

s1=Student(1,"poonam")
s1.display()
Student.displaytotal()

s2=Student(1,"janvi")
s2.display()
Student.displaytotal()

s3=Student(1,"purva")
s3.display()
Student.displaytotal()
```

Output:

```
ROll number : 1
Name : poonam
Total Student : 1
ROll number : 1
Name : janvi
Total Student : 2
ROll number : 1
Name : purva
Total Student : 3
```

Practical 13

Name: Poonam Satalkar
Roll No: 27 (B)

Que : wap to validate email id,password ,url and and mobile number using regular expression.

for email

```
import re
email_1 = re.compile(r'([A-Za-z0-9]+[.-_])*[A-Za-z0-9]+@[A-Za-z0-9-]+(\.[A-Z|a-z]{2,})+')
def emailValid(email):
    if re.fullmatch(email_1, email):
        print("The given mail is valid")
    else:
        print("The given mail is invalid")
emailValid("sachin.sharma@gmail.com")
emailValid("johnsnow123@yahoo.co.uk")
emailValid("mathew123@...uk")
emailValid("...@domain.us")
```

Output:

```
The given mail is valid
The given mail is valid
The given mail is invalid
The given mail is invalid
```

for password

```
import re
def main():
    passwd =input("enter password")
    reg = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!#%*?&]{6,20}$"

    pat = re.compile(reg)
    mat = re.search(pat, passwd)
    if mat:
        print("Password is valid.")
    else:
        print("Password invalid !!")

if __name__ == '__main__':
    main()
```

Output:

```
enter password   ass@123
Password invalid !!
```

Practical 14

Name: Poonam Satalkar
Roll No: 27 (B)

Que :Write a program to built any five Exception.

```
#Arithmetic Exception
try:
    a=10/0
    print(a)
except ArithmeticError:
    print("this statement is raising an arithmetic exception")
else:
    print("Success.")
```

Output:

```
this statement is raising an arithmetic exception
```

```
#Index out of bound error
try:
    a=[1,2,3,4,5]
    print(a[5])
except LookupError:
    print(" Index out of bound error")
else:
    print("Success.")
```

Output:

```
Index out of bound error
```

```
#ZeroDivisionError
a=int(input("Enter a :"))
b=int(input("Enter b :"))
c=a/b
print("a/b=%d"%c)
print("output",c)
```

Output:

```
Enter a :10
Enter b :0
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_11528\2929506835.py in <module>
      2 a=int(input("Enter a :"))
      3 b=int(input("Enter b :"))
----> 4 c=a/b
      5 print("a/b=%d"%c)
      6 print("output",c)
```

```
ZeroDivisionError: division by zero
```

```
#Syntax Error Exception
try:
    print(eval("welcome to nashik"))
except SyntaxError,err:
    print("Syntax error %s(%s-%s):%s"%\
          (err.filename,err.lineno,err.offset,err.text))
    print(err)
```

Output:

```
File "C:\Users\Admin\AppData\Local\Temp\ipykernel_16396\2241455757.py", line 5
    except SyntaxError,err:
                      ^
SyntaxError: invalid syntax
```

```
#Key Error Exception
array={'a':1,'b':1}
print(array['c'])
```

Output:

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_11528\1404083845.py in <module>
      1 #Key Error Exception
      2 array={'a':1,'b':1}
----> 3 print(array['c'])

KeyError: 'c'
```

Practical 15

Name: Poonam Satalkar
Roll No: 27 (B)

Que : wap to implement user defined exception to display message of account balance is below 1000 while withdrawing amount.

```
class Error(Exception):
    pass
class ValueTooSmallError(Error):
    pass
class ValueTooLargeError(Error):
    pass
print("your balance=1000")

while True:
    try:
        amt=int(input("Enter amt="))
        if amt<1000:
            raise ValueTooSmallError
        elif amt>1000:
            raise ValueTooLargeError
        break
    except ValueTooSmallError:
        print("you withdraw!!")
        print()

    except ValueTooLargeError:
        print("Insufficient Balance!!,try again")
        print()
```

Output:

```
your balance=1000
Enter amt=300
you withdraw!!

Enter amt=2000
Insufficient Balance!!,try again

Enter amt=1000
```

Practical 16

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a module to implement following arithmetic functions: add,substract,multiply & division,log,pow,sqrt,sin,cos,tan.
write a menu driven program to use these functions.

```
from math import *
pi=3.14
print("-------Menu-------")
print("1.Addition Function")
print("2.Subtraction Function")
print("3.Multiplication Function")
print("4.Division Function")
print("5.log Function")
print("6.pow Function")
print("7.sqrt Function")
print("8.sin Function")
print("9.cost Function")
print("10.tan Function")
print("11.Exit")

num1=int(input("Enter first number :"))
num2=int(input("Enter second number :"))

print("Enter which operation would you like to perform:")
ch=input("Select Option:")

result=0

if ch=='1':
    result=num1+num2
elif ch=='2':
    result=num1-num2
elif ch=='3':
    result=num1*num2
elif ch=='4':
    result=num1/num2
elif ch=='5':
    result=log(num1*e**num2)
elif ch=='6':
    result=pow(num1,num2)
elif ch=='7':
    result=sqrt(num1**2+num2**2)
elif ch=='8':
    result=sin(pi/num1)
```

```
elif ch=='9':
    result=cos(pi/num1)
elif ch=='10':
    result=tan(pi/num2)
elif ch=='11':
    exit()
else:
    print("Input Arithmetic function is not recognized")

print("Answer :",result)
```

Output:

```
-------Menu-------
1.Addition Function
2.Subtraction Function
3.Multiplication Function
4.Division Function
5.log Function
6.pow Function
7.sqrt Function
8.sin Function
9.cost Function
10.tan Function
11.Exit
Enter first number :12
Enter second number :4
Enter which operation would you like to perform:
Select Option:7
Answer : 12.649110640673518
```

Practical No .17

Name: Poonam Satalkar
Roll No: 27 (B)

Que : Write a program to display date in following format "Friday,23 April 2017" .

```
import datetime
x=datetime.datetime(2024,3,24)
print("\n Day And Date Format:")
print(x.strftime("%A,%d %B %Y"))
```

Output:

```
Day And Date Format:
Sunday,24 April 2024
```

Practical No .18

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a program to display number of days remaining up to 31 st Dec 2020

```
from datetime import datetime
date1=datetime(2021,12,31)
date2=datetime(2021,9,17)

diff=date1-date2
print("Date Remaining: ",diff)
```

Output:

```
Date Remaining:   105 days, 0:00:00
```

Practical No .19

Name: Poonam Satalkar
Roll No: 27 (B)

Que : Implement stack Functions,Implement Queue functions(using list)

```
stack=[]
stack.append('x')
stack.append('y')
stack.append('z')
print(stack)
print("Element pop from stack:")
print(stack.pop())
print(stack.pop())
print(stack.pop())
print(" stack after Element are poped:")
print(stack)
```

Output:

```
['x', 'y', 'z']
Element pop from stack:
z
y
x
 stack after Element are poped:
[]
```

# Implementation of Queue

```
queue=[]
queue.append(10)
queue.append(20)
queue.append(30)
queue.append(40)
print("Initial Queue is:",queue)
print("\n Elements dequeued from queue:")
print(queue.pop(0))
print(queue.pop(0))
print(queue.pop(0))
print(" queue after removing   Element:",queue)
```

Output:

```
Initial Queue is: [10, 20, 30, 40]

 Elements dequeued from queue:
10
20
30
 queue after removing   Element: [40]
```

Practical No .20

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a program for addition ,subtraction ,multiplication of two matrices using numpy

```
import numpy as np
a=np.array([[7,6],[5,4]])
b=np.array([[4,5],[3,2]])

print("Elements of matrix a :")
print(a)
print("Elements of matrix b :")
print(b)
print("Addition of two matrix is:")
print(np.add(a,b))
print("Subtraction of two matrix is:")
print(np.subtract(a,b))
print("Multiplication of two matrix is:")
print(np.multiply(a,b))
```

Output:

```
Elements of matrix a :
[[7 6]
 [5 4]]
Elements of matrix b :
[[4 5]
 [3 2]]
Addition of two matrix is:
[[11 11]
 [ 8   6]]
Subtraction of two matrix is:
[[3 1]
 [2 2]]
Multiplication of two matrix is:
[[28 30]
 [15   8]]
```

Practical No .21

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a program to accept a number in range 1 to 10 from the user .If the number is matches with the randomly generated number user wins else user looses.

```
import random
target_num,guess_num=random.randint(1,10),0
while target_num!=guess_num:
      guess_num=int(input("Guess the number between the range 1 to 10 :"))
print("\n User Wins")
```

Output:

```
Guess the number between the range 1 to 10 :4
Guess the number between the range 1 to 10 :5

 User Wins
```

Practical No .22

Name: Poonam Satalkar
Roll No: 27 (B)

Que : Write a python program for following functions: add,update,delete,reverse element in a tuple.find repeated elements.

```
def main():
    tupleObj=(12,34,45,55,22,33,55,32,55)
    print("\n----Add element at specific index in tuple----")
    print("Original Tuple :",tupleObj)
    n=2
    tupleObj=tupleObj[:n]+(19,)+tupleObj[n:]
    print("Modified Tuple :",tupleObj)
    print("\n----update element at specific index in tuple----")
    print("Original Tuple :",tupleObj)
    n=2
    tupleObj=tupleObj[:n]+('test',)+tupleObj[n+1:]
    print("Modified Tuple :",tupleObj)
if __name__=='__main__':
    main()
tupleObj=(12,34,45,55,22,33,55,32,55)
y=reversed(tupleObj)
print("---Reversed the elements in the tuple---")
print(tuple(y))
print("\n")
print("---Repeated elements in tuple---")
tupleObj=12,34,45,55,22,33,55,32,55
print(tupleObj)
count=tupleObj.count(55)
print(count)
print("\n The repeated element is 55 occurs 3 times")
```

Output:

```
----Add element at specific index in tuple----
Original Tuple : (12, 34, 45, 55, 22, 33, 55, 32, 55)
Modified Tuple : (12, 34, 19, 45, 55, 22, 33, 55, 32, 55)
----update element at specific index in tuple----
Original Tuple : (12, 34, 19, 45, 55, 22, 33, 55, 32, 55)
Modified Tuple : (12, 34, 'test', 45, 55, 22, 33, 55, 32, 55)
---Reversed the elements in the tuple---
(55, 32, 55, 33, 22, 55, 45, 34, 12)
---Repeated elements in tuple---
(12, 34, 45, 55, 22, 33, 55, 32, 55)
3
The repeated element is 55 occurs 3 times
```

Practical No .23

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a program to implement functions add, update,delete,search,display all students in a file.
store roll no, name ,total marks of a student in file.

```
class student:
    def __init__(self,name,rollno,m1,m2):
        self.name=name
        self.rollno=rollno
        self.m1=m1
        self.m2=m2
    def accept(self,Name,Rollno,marks1,marks2):
        ob=student(Name,Rollno,marks1,marks2)
        ls.append(ob)
    def display(self,ob):
        print("Name:",ob.name)
        print("RollNo:",ob.rollno)
        print("Marks1:",ob.m1)
        print("Marks2:",ob.m2)
        print("\n")
    def search(self,n):
        for i in range(ls.__len__()):
            if(ls[i].rollno==n):
                return i
    def delete(self,n):
        i=obj.search(n)
        del ls[i]
    def update(self,n,No):
        i=obj.search(n)
        roll=No
        ls[i].rollno=roll;

ls=[]
obj=student("",0,0,0)
print("\n Operation used :")
print("\n 1.Accept Student Details\n2. Display Student Details\n3 search Student Details\n4.
Delete Student Details\n 5.update Student Details\n 6 Exit")
obj.accept("A",1,100,100)
obj.accept("B",2,90,90)
obj.accept("C",3,80,80)
print("\n")
print("\n List of Students:")
for i in range(ls.__len__()):
    obj.display(ls[i])
print("\n student found")
s=obj.search(2)
obj.display(ls[i])
obj.delete(2)
print(ls.__len__())
print("List after deletion")
for i in range(ls.__len__()):
    obj.display(ls[i])
obj.update(3,2)
print(ls.__len__())
print("List after updation")
for i in range(ls.__len__()):
    obj.display(ls[i])
```

Output:

```
Operation used :

 1.Accept Student Details
2. Display Student Details
3 search Student Details
4. Delete Student Details
 5.update Student Details
 6 Exit



 List of Students:
Name: A
RollNo: 1
Marks1: 100
Marks2: 100


Name: B
RollNo: 2
Marks1: 90
Marks2: 90


Name: C
RollNo: 3
Marks1: 80
Marks2: 80
```

```
 student found
Name: C
RollNo: 3
Marks1: 80
Marks2: 80


2
List after deletion
Name: A
RollNo: 1
Marks1: 100
Marks2: 100


Name: C
RollNo: 3
Marks1: 80
Marks2: 80


2
List after updation
Name: A
RollNo: 1
Marks1: 100
Marks2: 100


Name: C
RollNo: 2
Marks1: 80
Marks2: 80
```

Practical No .24

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a python program to check that a string contains only a certain set of characters(in case a-z,A-Zand 0-9).

```
import re
def CharString(string):
    charRe=re.compile(r'[^a-zA-Z0-9.]')
    string=charRe.search(string)
    return not bool(string)
print(CharString("AGDADDFjnghsdvjc877853"))
print(CharString("@#$%^&*}{"))
```

Output:

```
True
False
```

Practical No .25

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a python program that matches a string that has an 'a' followed by two to three 'b'.

```
import re
def text_match(text):
    patterns='ab{2,3}'
    if re.search(patterns,text):
        return "Found a match"
    else:
        return "Not match"
print(text_match("ab"))
print(text_match("aabbbbbbcc"))
```

Output:

```
Not match
Found a match
```

Practical No .26

Name: Poonam Satalkar
Roll No: 27 (B)
Que : write a program to display last 5 lines in a file.

```
def LastNlines(fname,N):
    with open(fname) as file:
        for line in (file.readlines()[-N:]):
            print(line,end='')
if __name__=='__main__':
    fname='file1.txt'
    N=5
    try:
        LastNlines(fname,N)
    except:
        print("file not found")
```

Output:

```
2 khushi
3 dhanashri
4 ashwini
5 pallavi
6 sayali
```

Practical No .27

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a program using numpy : create a matrix with values ranging from 12 to 38, display matrix elements, transpose the matrix and display transpose matrix.

```
import numpy as np
array=np.arange(12,38)
array

matrix=np.array([[4,5,7],[8,2,3],[1,8,6]])
print("Matrix Elements")
print(matrix)

arr1=np.array([[4,5,7],[8,2,3]])
print(f'Original Matrix:\n{arr1}')

arr1_transpose=arr1.transpose()
print(f'Transpose Matrix:\n{arr1_transpose}')
```

Output:

```
Matrix Elements
[[4 5 7]
 [8 2 3]
 [1 8 6]]
Original Matrix:
[[4 5 7]
 [8 2 3]]
Transpose Matrix:
[[4 8]
 [5 2]
 [7 3]]
```

Practical No .28

Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a program using pandas for following:
read an excel file and import data in a data frame,perform data cleaning operation, display information about data.

```
import pandas as pd
data=pd.read_csv("student_por.csv")
data
```

Output:

	school	sex	age	address	famsize	pstatus	medu	fedu	mjob	fjob	...	famrel	freetime	goout	dalc	walc	health	absences	g1	g2	g3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	4	0	11	11
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	2	9	11	11
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	6	12	13	12
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	0	14	14	14
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	0	11	13	13

school	sex	age	address	famsize	pstatus	medu	fedu	mjob	fjob	...	famrel	freetime	goout	dalc	walc	health	absences	g1	g2	g3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
644	MS	F	19	R	GT3	T	2	3	services	other	...	5	4	2	1	2	5	4	10	11	10
645	MS	F	18	U	LE3	T	3	1	teacher	services	...	4	3	4	1	1	1	4	15	15	16
646	MS	F	18	U	GT3	T	1	1	other	other	...	1	1	1	1	1	5	6	11	12	9
647	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	6	10	10	10
648	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	4	10	11	11

649 rows × 33 columns

#for printing first 5 rows
data.head()
Output:

	school	sex	age	address	famsize	pstatus	medu	fedu	mjob	fjob	...	famrel	freetime	goout	dalc	walc	health	absences	g1	g2	g3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	4	0	11	11
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	2	9	11	11
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	6	12	13	12
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	0	14	14	14
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	0	11	13	13

5 rows × 33 columns

```
# data cleaning using isnull() function
data.isnull()
```

Output:

school	sex	age	address	famsize	pstatus	medu	fedu	mjob	fjob	...	famrel	freetime	goout	dalc	walc	health	absences	g1	g2	g3	
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
644	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

school	sex	age	address	famsize	pstatus	medu	fedu	mjob	fjob	...	famrel	freetime	goout	dalc	walc	health	absences	g1	g2	g3	
645	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
646	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
647	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
648	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

649 rows × 33 columns

In [9]:

#using isna() function
data.isna()
Output:

school	sex	age	address	famsize	pstatus	medu	fedu	mjob	fjob	...	famrel	freetime	goout	dalc	walc	health	absences	g1	g2	g3	
0	Fal	Fal	False	False	False	Fal	Fal	Fa	Fal	Fal	...	False	Fal	Fal	Fal	Fal	False	Fal	Fal	Fal	Fal

	school	sex	age	address	famsize	pstatus	medu	fedu	mjob	fjob	...	famrel	freetime	goout	dalc	walc	health	absences	g1	g2	g3
	se	se				se	se	lse	se	se			se	se	se	se		se	se	se	se
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
644	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
645	Fal	Fal	False	False	False	False	Fal	False	Fal	Fal	...	False	False	Fal	Fal	False	False	Fal	Fal	Fal	Fal

school	sex	age	address	famsize	pstatus	medu	fedu	mjob	fjob	...	famrel	freetime	goout	dalc	walc	health	absences	g1	g2	g3	
	se	se					se		se	se				se	se			se	se	se	se
646	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
647	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
648	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

649 rows × 33 columns

```
#using isna().any() function
data.isna().any()
```

Output:

```
school          False
sex             False
age             False
address         False
famsize         False
pstatus         False
medu            False
fedu            False
mjob            False
fjob            False
reason          False
guardian        False
```

```
traveltime      False
studytime        False
failures         False
schoolsup        False
famsup            False
paid               False
activities      False
nursery          False
higher           False
internet        False
romantic         False
famrel           False
freetime         False
goout             False
dalc              False
walc               False
health           False
absences         False
g1               False
g2               False
g3               False
dtype: bool
```

#using isna().any.sum()

```
data.isna().any().sum()
```

Output:

```
0
```

# using de-duplicated()

```
data.duplicated()
```

Output:

```
0         False
1         False
2         False
3         False
4         False
          ...
644       False
645       False
646       False
647       False
648       False
Length: 649, dtype: bool
```

#using of Info()

```
data.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 649 entries, 0 to 648
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      649 non-null    object
 1   sex         649 non-null    object
 2   age         649 non-null    int64
 3   address     649 non-null    object
 4   famsize     649 non-null    object
 5   pstatus     649 non-null    object
 6   medu        649 non-null    int64
 7   fedu        649 non-null    int64
 8   mjob        649 non-null    object
 9   fjob        649 non-null    object
 10  reason      649 non-null    object
 11  guardian    649 non-null    object
 12  traveltime  649 non-null    int64
 13  studytime   649 non-null    int64
 14  failures    649 non-null    int64
 15  schoolsup   649 non-null    bool
 16  famsup      649 non-null    bool
 17  paid        649 non-null    bool
 18  activities  649 non-null    bool
 19  nursery     649 non-null    bool
 20  higher      649 non-null    bool
 21  internet    649 non-null    bool
 22  romantic    649 non-null    bool
 23  famrel      649 non-null    int64
 24  freetime    649 non-null    int64
 25  goout       649 non-null    int64
 26  dalc        649 non-null    int64
 27  walc        649 non-null    int64
 28  health      649 non-null    int64
 29  absences    649 non-null    int64
 30  g1          649 non-null    int64
 31  g2          649 non-null    int64
 32  g3          649 non-null    int64
dtypes: bool(8), int64(16), object(9)
memory usage: 132.0+ KB
```

Practical No .29

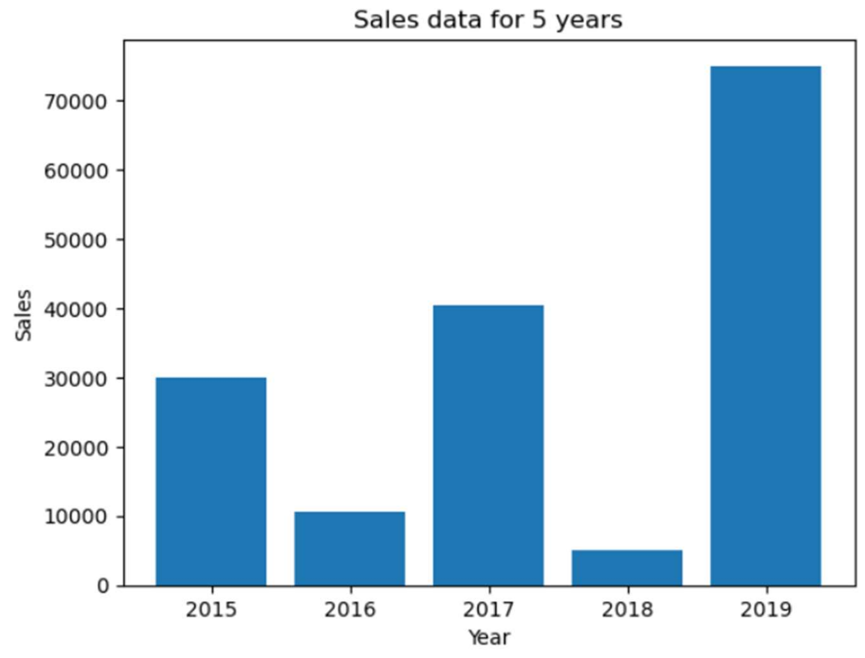
Name: Poonam Satalkar
Roll No: 27 (B)

Que : write a program using matplotlib to display bar chart for sales data for 5 years.Assume suitable data.

```
import matplotlib.pyplot as plt
x=['2015','2016','2017','2018','2019']
sales=[30000,10500,40500,5000,75000]
x_pos=[i for i, _ in enumerate(x)]

plt.bar(x_pos,sales)
plt.xlabel("Year")
plt.ylabel("Sales")
plt.title("Sales data for 5 years")
plt.xticks(x_pos,x)
plt.grid(which='minor',linestyle=':',linewidth='0.5',color='black')
plt.show()
```

Output:


Sales data for 5 years
Sales
Year