In [1]:

```
#sum of all digits in a number
num=int(input("Enter a four digit number"))
x0=num//1000
print(x0)
x1=(num-x0*1000)//100
print(x1)
x2=(num-x0*1000-x1*100)//10
print(x2)
x3=(num-x0*1000-x1*100-x2*10)
print(x3)
print("the sum of the digits is ",x0+x1+x2+x3)
```

```
Enter a four digit number1234
1
2
3
4
the sum of the digits is  10
```

In [2]:

```
#to calculate bonus and comission rate and the salary
basic_salary=2500
bonus_rate=200
comission_rate=0.12
n=int(input("enter the number of cameras:"))
price=float(input("enter the price of the camera"))
bonus=bonus_rate*n
print("bonus",bonus)
comission=n*comission_rate*price
print("the comission is",comission)
salary=bonus+basic_salary+comission
print("salary is:",salary)
```

```
enter the number of cameras:5
enter the price of the camera1000
bonus 1000
the comission is 600.0
salary is: 4100.0
```

In [3]:

```
#height in cms
print("enter your height")
h_ft=int(input("feet"))
h_inch=int(input("inches"))
h_inch+=h_ft*12
print(h_inch)
h_cm=round(h_inch*2.54,1)
print("your height is ",h_cm,'cm')
```

```
enter your height
feet5
inches2
62
your height is  157.5 cm
```

In [4]:

```
#area and perimeter of a circle
r=float(input("enter radius"))
area=round(3.14*r*r,2)
print("area of the circle",area)
perimeter=round(2*3.14*r,2)
print("perimeter of the circle is",perimeter)
```

```
enter radius4
area of the circle 50.24
perimeter of the circle is 25.12
```

In [6]:

```
# conversion of letters to consecutive letter
string=input("enter the string of four characters")
ch1=chr(ord(string[0])+1)
ch2=chr(ord(string[1])+1)
ch3=chr(ord(string[2])+1)
ch4=chr(ord(string[3])+1)
print("After Conversion:",ch1,ch2,ch3,ch4,sep="")
```

```
enter the string of four charactersabcd
After Conversion:bcde
```

In [7]:

```
# to find the roots of an equation
import math
a=float(input("enter the first coefficient"))
b=float(input("enter the second coefficient"))
c=float(input("enter the third coefficient"))
d=b*b-4*a*c
if(d>0):
    print("real and distict root")
    root1=(-b+math.sqrt(d)/(2*a))
    root2=(-b+math.sqrt(d)/(2*a))
    print("root1",root1,"root2",root2)
elif(d==0):
    print("real and equal roots")
    root1=root2=-b/(2*a)
    print("roots",root1,root2)
else:
    print("imaginary roots")
    real=-b/(2*a)
    img=math.sqrt(-d)/(2*a)
    print("root1",real)
```

```
enter the first coefficient1
enter the second coefficient-6
enter the third coefficient9
real and equal roots
roots 3.0 3.0
```

In [8]:

```
# to find the factors of a number and determine whether it is prime or not
n=int(input("enter a number "))
i=1
count=0
print("factor of ",n,":",end=" ")
while i<=n:
    if n%i==0:
        print(i,end=' ')
        count+=1
    i+=1
print()
if count==2:
    print("prime")
else:
    print("not prime")
```

```
enter a number 53
factor of  53 : 1 53
prime
```

In [9]:

```
#pyramid pattern 1
n=int(input("enter the desired size:"))
a=65
i=0
while i<n:
    j=0
    while j<n-i:

        print(chr(a+i+j),end=" ")
        j+=1
    i=i+1
    print()
```

```
enter the desired size:8
A B C D E F G H
B C D E F G H
C D E F G H
D E F G H
E F G H
F G H
G H
H
```

In [10]:

```
#pyramid pattern 2
n=int(input("Enter n:"))
i=1
while i<=n:
    j=1
    while j<=i:
        print(j,end=" ")
        j=j+1
    j=j-2
    while j>=i:
        print(j,end=" ")
        j=j-1
    i=i+1
    print()
```

```
Enter n:5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

In [11]:

```
# to determine if a number is co-prime
a=int(input("enter the first input:"))
b=int(input("enter the second input"))
import math
if math.gcd(a,b)==1:
    print(a,b,"are co-prime")
else:
    print(a,b,"are not co prime ")
```

```
enter the first input:5
enter the second input6
5 6 are co-prime
```

In [13]:

```
# program to determine the working of decorators
def my_decorator(fn):
    symbol='$'
    def wrapper(n):
        print("My today's earnings is"+symbol,end='')
        fn(n)
    return wrapper

@my_decorator

def day_earnings(number):
    print(number)
day_earnings(200)
day_earnings(3000)
day_earnings(80000)
```

```
My today's earnings is$200
My today's earnings is$3000
My today's earnings is$80000
```

In [16]:

```
# to sort a list of tuples.
s={("890","ram",(95,78,99)),("123","kishan",(90,98,89)),("567","arjun",(59,68,100))}


def get_totalmarks(t):
    return t[2][0]+t[2][1]+t[2][2]
name=sorted(s,key=get_totalmarks)
print("Student list is sorted because on total marks:\n",name)
```

```
Student list is sorted because on total marks:
 [('567', 'arjun', (59, 68, 100)), ('890', 'ram', (95, 78, 99)), ('123',
'kishan', (90, 98, 89))]
```

In [1]:

```
# geometric progression
a=int(input("starting number "))
r=int(input("enter the common ratio"))
m=int(input("enter the n th time the format"))
sum=0
for i in range(0,m):
    curr_term=a*pow(r,i)
    print(curr_term,end=' ')
    sum=sum+curr_term
print("sum",sum)
```

```
starting number 2
enter the common ratio3
enter the n th time the format5
2 6 18 54 162 sum 242
```

In [2]:

```
str_places = '''karnataka bangalore lalbagh
tamilnadu kanyakumari vivekananda_rock
kerala thiruvananthapuram padmanabha_temple
kerala idukki munnar
karnataka mysore brindavan_gardens
karnataka mysore mysore_palace
karnataka hassan shravanabelagola
tamilnadu chennai egmore_museum
tamilnadu kanyakumari kaamaakshmi_temple
karnataka bangalore cubbon_park
karnataka hampi maharnavami_dibba'''
```

In [4]:

```
# to count the number of states and their names(unique)
stateset = set()
for line in str_places.split('\n'):
    stateset.add(line.split()[0])
print("Number of states : ", len(stateset))
print("States : ", stateset)
```

```
Number of states :  3
States :  {'karnataka', 'tamilnadu', 'kerala'}
```

In [12]:

```
#Find and print the number of cities and their names for each state and find the number 

state_cities = {} # create an empty dict
state_cities_count = {} # create an empty dict
number_lines = str_places.split('\n')
for line in number_lines:
    state = line.split()[0]
    if state not in state_cities_count:
        state_cities_count[state] = 0
    state_cities_count[state] += 1
    city = line.split()[1]
    if state not in state_cities:
            state_cities[state] = set()
    state_cities[state].add(city)
#print(state_cities)
for state in state_cities:
    print(state, " => ", state_cities[state])
print()
#print(state_cities_count)
for state in state_cities_count:
    print(state, " => ", state_cities_count[state])
```

```
karnataka  =>  {'mysore', 'bangalore', 'hassan', 'hampi'}
tamilnadu  =>  {'kanyakumari', 'chennai'}
kerala  =>  {'thiruvananthapuram', 'idukki'}

karnataka  =>  6
tamilnadu  =>  3
kerala  =>  2
```

In [14]:

```
#Given a dict where values are not unique, write a function to
#create a new dict where the key is the value and the value is
#concatenated keys of the original dict and return it.
#original dictionary:{'apple':'fruit', 'cat':'mammal', 'beans':'veg',
#'dog':'mammal', 'mango':'fruit', 'carrot':'veg'}
#new dictionary:{'fruit': ['apple', 'mango'], 'mammal': ['cat', 'dog'], 'veg': ['beans',

def highest_scorer(dict2):
    new_dict={}
    for key in dict2:
        value=dict2[key]
        if value not in new_dict:
            new_dict[value] = []
        new_dict[value].append(key)
    return new_dict

dict1 = {'apple':'fruit', 'cat':'mammal', 'beans':'veg',
'dog':'mammal', 'mango':'fruit', 'carrot':'veg'}
print(highest_scorer(dict1))
```

```
{'fruit': ['apple', 'mango'], 'mammal': ['cat', 'dog'], 'veg': ['beans',
'carrot']}
```

In [ ]:

```

```