1.Write a program to print “Hello World!” to the output console.

```
print(‘Hello,world’)
```

2.Write a program to read something from the input console and write it to the output console.

```
x = input('What is your name? ')
print('Hii',x)
```

3.Write a program to find the sum and difference of two numbers.

```
a = int (input('Enter the first number : '))
b = int (input('Enter the second number : '))
print('The sum of two numbers is ',a+b)
print('The difference of two number is ',a-b)
```

4.Write a program to swap the contents of two variables with a temporary variable.

```
a = input('Enter the first number ')
b = input('Enter the second number ')
num = a
a = b
b = num
print(a)
print(b)
```

5.Write a program to swap the contents of two variables without a temporary variable.

```
a = input('Enter the first number : ')
b = input('Enter the second number : ')
a,b = b,a
print('Swapped ',a)
print(b)
```

6.Write a program to find the largest number among two numbers.

```
a = input('Enter the first number : ')
b = input('Enter the second number : ')
if (a>b) :
print('First is greater then second')
else:
print('Second is greater then
```

7.Write a program to find the largest number among three numbers.

```
a = input('Enter the first number : ')
b = input('Enter the second number : ')
c = input('Enter the third number : ')
if(a>b and b>c):
print('First number is greatest')
elif(b>a and a>c):
print('Second number is greatest')
else :
```

```
print('Third number is the greatest')
```

8.Write a program to find the largest string (size wise & alphabetical wise) among two strings.

```
a = input('Enter the first string : ')
b = input('Enter the second string : ')
len1 = len(a)
len2 = len(b)
if(len1>len2):
print('First String is Large')
else:
print('Second String is Large')
```

9.

Write a program to input marks of a student and to find the percentage & grade as per the SMVDU norms.

```
M = int(input('Enter your full marks : '))
per = M/500*100
if(per>=90):
print('Your Grade is A')
elif(per>=80):
print('Your Grade is B+')
elif(per>=70):
print('Your Grade is B')
elif(per>=60):
print('Your Grade is C+')
elif(per>=50):
print('Your Grade is C')
else:
print('You got Grade D ')
```

10.

Write a program to print the multiplication table of a number up to a range.

```
a = int(input('Enter the number whose table you want : '))
for i in range(1,11):
print(i*a)
```

11.

Write a program to find the area and perimeter of shapes (triangle, rectangle and circle).

```
q = int(input('Enter the choice triangle,rectangle,circle(1/2/3)'))
if(q== 1):
a = int(input('Enter the height of triangle : '))
b = int( input('Enter the base of triangle : '))
c = int( input('Enter the side of triangle : '))
area = a*b/2
perimeter = a+b+c
print('The area of triangle is',area,'and its perimeter is',perimeter)
```

```
elif(q== 2):
l = int(input('Enter the length of rectangle : '))
h = int(input('Enter the height of rectangle : '))
ar = l*h
pr = 2*(l+h)
print('The area of rectangle is',ar,'and its perimeter is',pr)
else:
r = int(input('Enter the radius of circle : '))
are = 3.14 * r*r
peri = 6.28*r
print('The area of circle is',are,'and its perimeter is',peri)
```

12.

Write a program to calculate the net salary of an employee.
(Net salary = BP+TA+DA+HRA, TA = 5% BP, DA = 10% BP, HRA = 15% BP)

```
BP = int(input('Enter your basic pay : '))
TA = 5/100* BP
DA = 10/100* BP
HRA = 15/100*BP
Netsalary = BP+TA+DA+HRA
print('Your total salary',Netsalary)
```

13.Write a program to find the factors of a number.

```
num = int(input("Enter the number :"))
print("Factor of number are :")
for i in range(1,num+1):
if(num%i==0):
print(i)
```

14.

```
# Write a program to print each digits of a number.
num = input('Enter a number')
for i in num :
print(i)
```

15.

```
# Write a program to check whether a given number is prime or not.
num = int(input('Enter a number '))
if(num%2==0):
print('It is Even Number')
else:
print('It is Odd Number')
```

16.

```
# Write a program to check whether a given number is prime or not.
num = int(input('Enter a number : '))
f = False
for i in range(2,num):
if(num%i==0):
```

```
f = True
else:
f=False
if(f==True):
print(num,'is prime number')
else:
print(num,'is not Prime number')
```

17.

```
# Write a program to check whether a given number is armstrong or not.
n = int(input('Enter a number : '))
t = n
r = 0
while(n>0):
d = n%10
r = r*10+d
n=n//10
if(t==r):
print('Yes,it is a Palindrome Number')
else:
print('No it is not a Palindrome Number')
```

18.

```
# Write a program to check whether a given number is armstrong or not.
n = int(input('Enter the number : '))
l = n
s = 0
while(n>0):
d = n%10
n = n/10
s = s+d*d*d
if(s==l):
print('It is an armstrong number')
else:
print('It is not an armstrong number')
```

19.

```
# Write a program to check whether a given year is leap year or not.
year = int(input('Enter the year '))
x = year%100
r = x%4
if(r==0):
print('It is a leap year')
else:
print('It is not a leap year')
```

20.

```
# Write a program to find the factorial of a number.
num = int(input('Enter a number : '))
factorial = 1
```

```
if(num<0):
print('Factorial does not exist for negative number')
elif(num==0):
print('Factorial is 1')
else:
for i in range(1,num+1):
factorial = factorial*i
print('The factorial of these number is ',factorial)
```

21.

```
# Write a program to generate the Fibonacci series up to a range.
n = 5
a = 0
b = 1
sum = 0
count = 1
print("Fibonacci Series: ", end = " ")
while(count <= n):
print(sum, end = " ")
count += 1
a = b
b = sum
sum = a + b
```

22.

```
# Write a program to generate the first “n” Fibonacci numbers.
n = int(input("Enter the value of 'n': "))
a = 0
b = 1
sum = 0
count = 1
print("Fibonacci Series: ", end = " ")
while(count <= n):
print(sum, end = " ")
count += 1
a = b
b = sum
sum = a + b
```

23.

```
# Write a program to print following pattern.
'''*
**
***'''
for i in range(1,5):
for j in range(1,i):
print('*',end = '')
print('\n')
```

24.

```
#Write a program to print the following pattern.
'''....
***
**
*'''
for i in range(4,0,-1):
for j in range(1,i):
print('*',end='')
print('\n')
```

25.

```
#Write a program to print the following pattern.
'''*
**
***
**
*'''
for i in range(1,5):
for j in range(1,i):
print('*',end = '')
print('\n')
for i in range(4,0,-1):
for j in range(1,i):
print('*',end='')
print('\n')
```

26.

```
#Write a program to print the diamond pattern.
''' *
* *
* * *
* *
*'''
n=int(input("enter the number of rows"))

for i in range(n):

for j in range(1,int((n/2))-i+3):

print(sep=" ",end=" ")

for k in range(1,i+2):

print("*", end=" ")



print()

for i in range(n):
```

```
for j in range(1,5-(int((n/2))-i+3)+2):

print(sep=" ",end=" ")

for k in range(1,5-i):

print("*", end=" ")

print()
```

27.

```
#Write a program to print following pattern.
''' 1
12
123'''
def a(n):
num=1
for i in range (0,n):
num=1
for j in range (0,i+1):
print(num ,end=" ")
num=num+1
print("\r")
n=5
a(n)
```

28.

```
#Write a program to print the following pattern.
'''.....
321
32
3'''
def a(n):
num=1
for i in range (4,n,-1):
num=3
for j in range (0,i-1):
print(num ,end=" ")
num=num-1
print("\r")
n=0
a(n)
```

29.

```
# Write a program to print the following pattern.
''' 1
2 3
4 5 6'''
```

```
currentNumber = 1

stop = 2

rows = 3

for i in range(rows):

for column in range(1, stop):

print(currentNumber, end= "")

currentNumber += 1

print("")

stop += 2
```

30.

```
#Write a program to print the following pattern.
''' 1
2 3
4 5 6
7 8'''
def halfDiamondStar(N):
for i in range(N):
for j in range(1, i + 1):
print(j, end = "")
print()
for i in range(1, N):
for j in range(i, N):
print(j, end = "")
print()
N = 5;
halfDiamondStar(N);
```

31.

```
#Write a program to find the sum of first “n” natural numbers using recursion.
def recur_sum(n):
if n <= 1:
return n
else:
return n + recur_sum(n-1)


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

if num < 0:
print("Enter a positive number")
else:
print("The sum is",recur_sum(num))
```

32.

```
#Write a program to find the factorial of a number using recursion.

def recur_factorial(n):
if n == 1:
return n
else:
return n*recur_factorial(n-1)

num = int(input('Enter a number : '))


if num < 0:
print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
print("The factorial of 0 is 1")
else:
print("The factorial of", num, "is", recur_factorial(num))
```

33.

```
#Write a program to generate the fibonacci series up to a range using recursion.


def recur_fibo(n):
if n <= 1:
return n
else:
return(recur_fibo(n-1) + recur_fibo(n-2))
nterms = int(input("How many terms? "))
if nterms <= 0:
print("Plese enter a positive integer")
else:
print("Fibonacci sequence:")
for i in range(nterms):
print(recur_fibo(i))
```

34.

```
#Write a program to find out the doublet and triplet of a number using lambda functions.

from itertools import combinations

lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(lambda(lst, key): lambda(val): sum(val) == key ;list(filter(valid, list(combinations(lst,
3)))))
```

35.

```
#Write a program to implement the use of arguments in to a function and return values
from a function.
def greet(name, msg):
```

```
print("Hello", name + ', ' + msg)

greet("Gos", "Good morning!")
def sum():
return(65)
print(sum())
```

36.

```
#Write a program to demonstrate the use of local and global variables.
x = "global "

def foo():
global x
y = "local"
x = x * 2
print(x)
print(y)

foo()
```

37.

```
# Write a program to implement the usage of some built-in library modules (math, matplotlib, turtle, numpy, random etc.)math
import math
print(math.sqrt(4))
print(math.pi)
print(math.e)
print(math.radians(30))
print(math.degrees(math.pi/6))
print(math.sin(90))
print(math.cos(90))
print(math.tan(90))
print(math.log(8))

#!/usr/bin/env python
# coding: utf-8

# # Hello welcome

# In[1]:


import numpy as np


# In[2]:


myarr = np.array([3,4,5,6])
```

```
# In[3]:

myarr

# In[4]:

print(myarr)

# In[5]:

mua = np.array([3,4,5,6],np.int32)

# In[6]:

mua

# In[7]:

mua[0]

# In[8]:

mj=np.array([[1,2,3],[3,4,5]])

# In[9]:

mj

# In[10]:

print(mj)

# In[11]:
```

```
mj[1,2]
```

```
# In[12]:
```

```
mua.shape
```

```
# In[13]:
```

```
mj.shape
```

```
# In[14]:
```

```
mua.dtype
```

```
# In[15]:
```

```
mj.dtype
```

```
# In[16]:
```

```
mj[0,1]=90
```

```
# In[17]:
```

```
mj
```

```
# In[18]:
```

```
mua[0]=60
```

```
# In[19]:
```

```
mua
```

```
# In[20]:
```

```
mua.size
```

```
# In[21]:
```

```
li = np.array([[1,2,3],[4,5,6],[7,8,9]])
```

```
# In[22]:
```

```
li.size
```

```
# In[23]:
```

```
mj.size
```

```
# In[24]:
```

```
k=np.array({1,2,3})
```

```
# In[25]:
```

```
k.dtype
```

```
# In[26]:
```

```
zer = np.zeros((2,5))
```

```
# In[27]:
```

```
zer
```

```
# In[28]:
```

```
zer.dtype
```

```
# In[29]:
```

```
np.range(15)
```

```
# In[ ]:
```

```
sp = np.linspace(1,5,5)
```

```
# In[ ]:
```

```
sp
```

```
# In[ ]:
```

```
emp = np.empty(4,6)
```

```
# emp_like=np.empty.like(moj)
```

```
# In[31]:
```

```
arr = np.identity(45)
```

```
# In[32]:
```

```
arr
```

```
# In[34]:
```

```
arr = np.arrange(99)
```

```
# In[37]:
```

```
mj = mj.reshape(3,2)
```

```
# In[38]:
```

```
mj
```

```
# In[39]:
```

```
mj.ravel()
```

```
# In[40]:
```

```
li
```

```
# In[41]:
```

```
kp = np.array(li)
```

```
# In[42]:
```

```
kp.sum(axis=0)
```

```
# In[43]:
```

```
kp.sum(axis=1)
```

```
# In[44]:
```

```
kp.T
```

```
# In[46]:
```

```
kp.ndim
```

```
# In[47]:
```

```
kp.flat
```

```
# In[48]:
```

```
for i in kp.flat:
print(i)
```

```
# In[49]:
```

```
kp.size
```

```
# In[50]:
```

```
kp.nbytes
```

```
# In[51]:
```

```
lol = np.array([4,7,8,9,6])
```

```
# In[52]:
```

```
lol.argmax()
```

```
# In[53]:
```

```
lol.argmin()
```

```
# In[54]:
```

```
lol.min()
```

```
# In[55]:
```

```
lol.max()
```

```
# In[56]:
```

```
lol.argsort()
```

```
# In[58]:
```

```
mj.argmax()
```

```
# In[59]:
```

```
mj.argmin()
```

```
# In[61]:
```

```
mj.argsort()
```

```
# In[62]:
```

```
mj.max()
```

```
# In[63]:
```

```
mj.min()
```

```
# In[64]:
```

```
mj.min(axis=0
)
```

```
# In[65]:
```

```
mj.max(axis=1)
```

```
# In[66]:
```

```
mj.argsort(axis=0)
```

```
# In[67]:
```

```
mj.argmax(axis=0.)
```

```
# In[68]:
```

```
mj.savel()
```

```
# In[69]:
```

```
mj
```

```
# In[70]:
```

```
mj.reshape(6,)
```

```
# In[71]:
```

```
ar1 = np.array([[4,7,8],[7,9,8],[5,3,8]])
ar2 = np.array([[2,5,4],[8,9,4],[7,2,3]])
prod = ar1*ar2
sum = ar1+ar2
```

```
# In[72]:
```

```
prod
```

```
# In[73]:
```

```
sum
```

```
# In[74]:
```

```
np.sqrt(ar2)
```

```
# In[75]:
```

```
np.where(ar2>5)
```

```
# In[77]:
```

```
np.count_nonzero(ar2)
```

```
# In[78]:
```

```
np.nonzero(ar2)
```

```
# In[79]:
```

```
import sys
```

```
# In[80]:
```

```
ch = [0,1,2,3,4]
sys.getsizeof(1)*len(ch)
```

```
# In[83]:
```

```
l = np.array([0,1,2,3,4])
sys.getsizeof(1)*len(l)
```

```
# In[ ]:
```

```
#!/usr/bin/env python
# coding: utf-8
```

```
# In[1]:
```

```
import numpy as np
import pandas as pd
```

```
# In[2]:
```

```
dict = {
"name":['hero','king','love','chandu','kappu','raju'],
"marks":[90,86,97,56,86,99],
"city":['kolapur','sonpur','bokaro','agra','bihar','delhi']
}
```

```
# In[3]:
```

```
df = pd.DataFrame(dict)
```

```
# In[4]:
```

```
df
```

```
# In[5]:
```

```
df.to_csv('marks.csv')
```

```
# In[6]:
```

```
df.to_csv('marks_false.csv',index = False)
```

```
# In[7]:
```

```
df.head()
```

```
# In[8]:
```

```
df.head(2)
```

```
# In[9]:
```

```
df.tail(2)
```

```
# In[10]:
```

```
df.describe()
```

```
#!/usr/bin/env python
# coding: utf-8
```

```
# In[2]:
```

```
from matplotlib import pyplot as plt
plt.plot([1,2,3],[4,5,1])
plt.show()
```

```
# In[4]:
```

```
x = [5,6,7]
y = [7,9,6]
plt.plot(x,y)
plt.title('Hur')
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.show()
```

```
# In[6]:
```

```
from matplotlib import pyplot as plt
from matplotlib import style
```

```
# In[9]:
```

```
style.use('ggplot')
x = [5,6,7]
y = [7,9,6]
plt.plot(x,y,'c',label='line',linewidth = 5)
plt.title('Hur')
plt.ylabel('Y axis')
plt.xlabel('X axis')
```

```
plt.grid(True,color = 'k')
plt.show()


# In[5]:


from matplotlib import pyplot as plt
plt.bar([1,3,4,5],[5,2,7,8],label = 'Example One')
plt.bar([2,6,7,4],[4,6,3,9],label = 'Example Two')
plt.legend()
plt.title('Hang')
plt.xlabel('Sell')
plt.ylabel('Taxes')
plt.show()


from turtle import *
import turtle
color('red')
begin_fill()
forward(150)
left(90)
forward(200)
left(90)
forward(25)
left(90)
forward(175)
right(90)
forward(100)
right(90)
forward(175)
left(90)
forward(25)
left(90)
forward(200)
end_fill()
turtle.hideturtle()
done()
```

38.

```
#Write a program to implement the use of user defined library modules.
from calc import add
print(add(10, 2))
```

39.

```
#Write a program to print the sine and cosine values for the degrees 0, 30, 45, 60 and 90.

import math
gun =[0,30,45,60,90]
for i in gun :
```

```
print(math.sin(i))
print(math.cos(i))
```

40.

```
"""Write a program to make a list in Python and perform following operations on List:
a) length using len() function
b) print element at index 0
c) adding an elements of a list to another list using + operator
d) appending an element to the list
e) negative indexing in list
f) remove the first occurrence of element a from list
g) reverse the list
h) sort list
"""
friend = ["Karan","Love","Kiya","Kelly","Alia"]
print(friend[0])
# printing any element from list
print(friend[1:3:-1])
# printing element upto any num from list
print(friend)
# printing whole list
friend.append("Creed")
print(friend)
# adding new element at last in list
friend.insert(2,"Me")
print(friend)
# adding new element in certain place
friend = ["Karan","Love","Me","Kiya","Kelly","Alia"]
friend.remove("Me")
print(friend)
# removing any element from list
friend.clear()
print(friend)
# clear all element from list
friend = ["Karan","Love","Me","Kiya","Kelly","Alia"]
friend.pop()
print(friend)
# to remove last element from list
print(friend.index("Me"))
# to get its index
friend = ["Karan","Love","Me","Kiya","Kelly","Alia","Me","Me"]
print(friend.count("Me"))
# to get numbers of time element it in the list
luck_num = ["89","78","45","56","73","39"]
luck_num.sort()
print(luck_num)
# to arrange in ascending order
luck_num.reverse()
print(luck_num)
# to reverse the list
```

41.

```
#Write a program to demonstrate list creation, copy the entire list to another list.
luck_num = ["89","78","45","56","73","39"]
frnd = luck_num.copy()
print(frnd)
# to copy one list to other
```

42.

```
#Write a program to demonstrate slicing operations on the list.
my_list = [1, 2, 3, 4, 5]

print(my_list[:])
print(my_list[2:])
print(my_list[:2])
print(my_list[2:4])
print(my_list[::2])
print(my_list[::-2])
print(my_list[1:4:2])
```

43.

```
#Write a program to perform spilt and join operation on a list.
result1 = slice(3)
print(result1)
result2 = slice(1, 5, 2)
print(slice(1, 5, 2))
py_string = 'AlexSir'
slice_object = slice(3)
print(py_string[slice_object])
slice_object = slice(1, 6, 2)
print(py_string[slice_object])
pystring = 'Python'
slice_object = slice(-1, -4, -1)
print(pystring[slice_object])
mary = 'Mary had a little lamb'
mary.split()
mwords = mary.split()
print(mwords)
print(mary.join(mwords))
```

44.

```
#Write a program to search an element in a list, find the number of occurrences and the index of its first occurrence.
friend = ["Karan","Love","Me","Kiya","Kelly","Alia","Me","Me"]
print(friend.count("Me"))

print(friend.index("Me"))
if 'love' in friend:
print('Yes')
else:
print('No')
```

45.

```
#Write a program to create a 2D list and a 3D list.
a_list = [[2,3,4],[5,6,7]]
print(a_list[0][0])
l = a_list[0][0]*a_list[1][0]
print(l)
for i in a_list:
print(i)
b_list = [[2,3,4],[5,6,7],[8,4,1]]
print(int(b_list[0][0][0]))
l = b_list[0][0][0]*b_list[1][0][1]
print(l)
for j in b_list:
print(j)
```

46.

```
#Write a program to iterate over a 2D list in different ways.
rows =3
columns= 2
mylist = [[0 for x in range(columns)] for x in range(rows)]
for i in range(rows):
for j in range(columns):
mylist[i][j] = '%s,%s'%(i,j)
print(mylist)
```

47.

```
#Write a program to demonstrate the use of numpy library for creating arrays in Python.
import numpy as np
myarr = np.array([3,4,5,6])
print(myarr)
mua = np.array([3,4,5,6],np.int32)
print(mua)
print(mua[0])
mj=np.array([[1,2,3],[3,4,5]])
print(mj)
print(mj[1,2])
zer = np.zeros((2,5))
print(zer)
print(sp = np.linspace(1,5,5))
print(emp = np.empty(4,6))
print(arr = np.identity(45))
print(arr = np.arrange(99))
print(mj = mj.reshape(3,2))
```

48.

```
#Write a program to find the sum and difference of two matrices.
import numpy as np
ar1 = np.array([[4,7,8],[7,9,8],[5,3,8]])
ar2 = np.array([[2,5,4],[8,9,4],[7,2,3]])

add = ar1+ar2
differenc = ar2-ar2
```

```
print(add)
print(differenc)
```

49.

```
#Write a program to find the product of two matrices.
import numpy as np
ar1 = np.array([[4,7,8],[7,9,8],[5,3,8]])
ar2 = np.array([[2,5,4],[8,9,4],[7,2,3]])
prod = ar1*ar2
print(prod)
```

50.

```
#Write a program to demonstrate use of tuple in Python with their inbuilt functions.
my_tuple = ('Max','28','Boston')
print(type(my_tuple))
print(my_tuple[0])
for i in my_tuple:
print(i)
if'max'in my_tuple:
print('Yes')
else:
print('No')
print(len(my_tuple))
print(my_tuple.count('P'))
print(my_tuple.index('o'))
my_list = list(my_tuple)
print(type(my_list))
```

51.

```
myset ={1,2,3}
print(myset)
myset.add(2)
print(myset)
myset.discard(3)
print(myset)
myset1 ={1,2,3,4,5}
myset1.clear()
print(myset1)
print(myset.pop())
myset2 ={1,2,3,4,5}
u = myset2.union(myset)
print(u)
diff = myset1.difference(myset2)
```

52.

```
#Write a program to demonstrate use of dictionary in Python with their inbuilt functions.
mydict ={
'name':'Alex','age':30,'University': 'SMVDU'
}
mydict2 ={
```

```
'name' :'Sir','age':34,"University" : 'SMVDU'
}
value = mydict['name']
print(value)
mydict['email']='alex@smvdu.ac.in'
del mydict['name']
print(mydict)
print(mydict.pop('age'))
print(mydict.popitem())
mydict_cpy = dict(mydict)
print(mydict_cpy)
mydict.update(mydict2)
print(mydict)
```

53.

```
#Write a program to implement the usage of iterators on collections.
from collections import namedtuple
from itertools import product
point = namedtuple('Point',['x','y'])
pt = point(1,-4)
pt2 = point(4,6)
prod = product(pt,pt2)
print(list(prod))
```

54.

```
#Write a program to implement the usage of iterators on strings.
iterable_value = 'SMVDU'
iterable_obj = iter(iterable_value)
while True:
try:
item = next(iterable_obj)
print(item)
except StopIteration:
break
```

55.

```
#Write a program to implement the usage of look up table as a dictionary for finding out the country name from code of 25 countries.
country ={
1:'Afghanistan',
2:'Albania',
3:'Algeria',
4:'Andorra',
5:'Angola',
6:'Antigua',
7:'Argentina',
8:'Armenia',
9:'Australia',
10:'Austria',
11:'Azerbaijan',
12:'Bahamas',
```

```
13:'Bahrain',
14:'Bangladesh',
15:'Barbados',
16:'Belarus',
17:'Belgium',
18:'Belize',
19:'Benin',
20:'Bhutan',
21:'Bolivia',
22:'Bosnia',
23:'Botswana',
24:'Brazil',
25:'Brunei',
}
print(country[1])
```

56.

```
#Write a program to create a look up table using dictionary for finding the factorial of numbers up to 25.
factorial = {
1:1,
2:2,
3:6,
4:24,
5:120,
6:720,
7:504,
8:40320,
9:362880
,10:3628800
,11:39916800
,12:479001600
,13:6227020800
,14:87178291200
,15:1307674368000
,16:20922789888000
,17:355687428096000
,18:6402373705728000
,19:121645100408832000
,20:2432902008176640000
,21:51090942171709440000
,22:1124000727777607680000
,23:25852016738884976640000
,24:620448401733239439360000
,25:15511210043330985984000000
}
print(factorial[4])
```

57.

```
#Write a program to implement the concept of class and object creation.
class Person:
```

```
age = 10

def greet(self):
print('Hello')



print(Person.age)
print(Person.greet)
print(Person.__doc__)


class Person:
age = 10

def greet(self):
print('Hello')


harry = Person()
print(Person.greet)
print(harry.greet)

harry.greet()
```

58.

```
#Write a program to store marks of “n” students in a class for “m” subjects using OOP.
class Marks () :
dict1={}
n=int(input ("Enter the number of students: "))
m=int(input ("Enter the number of subjects: "))
for i in range(1,n+1) :
marks=[]
print("Enter the marks of",i," student: ", end="")
for j in range(1,m+1) :
x=int(input())
marks.append (x)
dict1[i]=marks
obj=Marks()
print (obj.dict1)
```

59.

#Write a program to create a class for animals and to create sub-classes for birds, dogs and humans.

```
class Animals:
c1= 'hair'
c2= 'heart'
c3= 'blood'
class Birds (Animals) :
c4= 'feathers'
c5= 'beak'
c6= 'two legs'
```

```
class Dogs (Animals) :
c7= 'four legs'
c8= 'canines'
c9= 'Tails'
class Humans (Animals) :
c10= 'Two hands and Two legs'
c11= 'Most developed'
c12= 'variety of organs'

obj1=Animals ()
obj2=Birds ()
obj3=Dogs ()
obj4=Humans ()

print (obj1.c1)
print (obj2.c2)
print (obj2.c4)
print (obj3.c3)
print (obj4.c3)
```

60.

#Write a program to create a class for polygons and to create sub-classes for triangles, rectangles, squares, pentagons etc.

```
class Shape (object):
sides = None
def _init_(self, sides) :
self.sides = sides
def perimeter (self) :
perimeter = 0
for side in self.sides :
perimeter += side
return perimeter
class Triangle(Shape):
def _init_(self,side1,side2,side3) :
self.sides = [side1, side2, side3]
class Rectangle(Shape):
def _init_(self,length,width) :
self.sides = [length, width, length, width]
class Square(Shape) :
def _init_(self, side) :
self.sides = [side, side, side, side]

triangle = Triangle(3, 4, 5)
print("Triangle sides: ", triangle.sides)
print("Perimeter of triangle: ", triangle.perimeter ())
rectangle = Rectangle(4, 2)
print("Rectangle sides: ", rectangle.sides)
print ("Perimeter of rectangle: ", rectangle.perimeter ())
square = Square(2)
print ("Square sides: ", square.sides)
print("Perimeter of square: ", square. perimeter ())
```

61.
#Write a program to create a class for trees and to create sub-classes like fruits, dry fruits, juices etc. (multi-level inheritance)

```
class tree :
def _init_ (self, n, c) :
self.name = n
self.colour = c
class fruits (tree) :
def _init_ (self, n, c) :
tree. _init_(self, n, c)
def p(self) :
print ("The fruit is:", self.name)
print("The colour is:", self.colour)
class veg (tree) :
def _init_ (self, n, c) :
tree. _init_ (self, n, c)
def p(self) :
print ("The vegetable is:", self.name)
print ("The colour is:", self.colour)
class juice (tree) :
def _init_ (self, n, f) :
tree. _init_(self, n, f)
def p(self) :
print("The juice is of fruit:", self.colour)
class dry (tree) :
def _init_ (self, n, c) :
tree. _init_(self, n, c)
def p(self) :
print("The dry fruit is:", self.name)
print("The colour is:", self.colour)
n = input ("Enter the type of food (Fruit/Vegetable/Juice/Dry Fruit) :")
if n == "Fruit":
a=input("Enter the name: ")
b=input("Enter the colour: ")
f1 =fruits (a, b)
f1.p()
elif n == "Vegetable" :
a=input("Enter the name: ")
b=input("Enter the colour: ")
v1 =veg (a, b)
v1.p()
elif n == "Juice" :
a=input("Enter the fruit it is made of:")
j1 = juice ("Juice",a)
j1.p()
elif n == "Dry Fruite" :
a=input("Enter the name: ")
b=input("Enter the colour: ")
d1 =dry (a, b)
d1.p()
```

```
else :
print ("Invalid Input. ")
```

62.
#

Write a program to create a class for students and this class should inherit the properties of all departments. (multiple inheritance)

```
class CSE:
def_init _(self, a, b) :
self.sub1 = a
self.sub2 = b
def subCSE (self) :
print ("The first subject is: ", self.sub1)
print("The second subject is: ", self.sub2)
class ECE:
def _init _(self, a) :
self.sub3 = a
def subECE (self) :
print("The third subject is: ", self.sub3)
class MATH:
def _init_(self, a):
self.sub4 = a
def subMATH (self) :
print ("The fourth subject in: ", self.sub4)
class PSC:
def _init_(self, a) :
self.sub5 = a
def subPSC (self) :
print("The fifth subject is: ", self.sub5)
class Student (CSE, ECE, MATH, PSC) :
def _init_(self, n, s1, s2, s3, s4, s5) :
self.name = n
print("\nThe student is of CSE Dept.\nHis name is: ", self.name)
CSE._init_(self, s1, s2)
ECE._init_(self, s3)
MATH._init_(self, s4)
PSC._init_(self, s5)

a = "Data Structure using C"
b = "Programming using Python"
c = "Digital Electronics"
d = "Engineering Mathematics"
e = "Discourse on Human Virtues"
n = input ("Enter the student name: ")
s = Student (n, a, b, c, d, e)
s.subCSE ()
s.subECE ()
```

63.

```
#Write a program to create all shapes in a canvas using tkinter library.
from tkinter import *
root = Tk()
root.geometry('300x300')

c = Canvas(root,height=250,width=300,bg='blue')
l = c.create_line(5,5,200,200,width=5)
o = c.create_oval(20,20,100,100,fill='red')
a = c.create_rectangle(50,50,100,200,fill='red')
#k = c.create_rectangle(100,100,100,100,fill='red')

c.pack()


root.mainloop()
```

64.

```
#Write a program to create a table of 3*3 square cells in a canvas and to colour each cell
differently using tkinter library.
from tkinter import *
root = Tk()
root.geometry('300x300')

c = Canvas(root,height=250,width=300,bg='red')
r1 = c.create_rectangle(20,20,50,50,fill='blue')
r2 = c.create_rectangle(70,20,50,50,fill='red')
r3 = c.create_rectangle(20,70,50,50,fill='green')
r4 = c.create_rectangle(120,20,50,50,fill='snow')
r5 = c.create_rectangle(20,120,50,50,fill='white')
r6 = c.create_rectangle(120,70,50,50,fill='red')
r7 = c.create_rectangle(70,120,50,50,fill='black')
r8 = c.create_rectangle(20,20,50,50,fill='deep sky blue')
r9 = c.create_rectangle(20,20,50,50,fill='yellow')



c.pack()


root.mainloop()
```

65.

```
#Write a program to prepare a simple registration form using tkinter library.
from tkinter import *

root = Tk()
root.geometry('500x500')
root.title("Registration Form")

label_0 = Label(root, text="Registration form",width=20,font=("bold", 20))
label_0.place(x=90,y=53)
```

```
label_1 = Label(root, text="FullName",width=20,font=("bold", 10))
label_1.place(x=80,y=130)

entry_1 = Entry(root)
entry_1.place(x=240,y=130)

label_2 = Label(root, text="Email",width=20,font=("bold", 10))
label_2.place(x=68,y=180)

entry_2 = Entry(root)
entry_2.place(x=240,y=180)

label_3 = Label(root, text="Gender",width=20,font=("bold", 10))
label_3.place(x=70,y=230)
var = IntVar()
Radiobutton(root, text="Male",padx = 5, variable=var, value=1).place(x=235,y=230)
Radiobutton(root, text="Female",padx = 20, variable=var, value=2).place(x=290,y=230)

label_4 = Label(root, text="country",width=20,font=("bold", 10))
label_4.place(x=70,y=280)

list1 = ['Canada','India','UK','Nepal','Iceland','South Africa'];
c=StringVar()
droplist=OptionMenu(root,c, *list1)
droplist.config(width=15)
c.set('select your country')
droplist.place(x=240,y=280)

label_4 = Label(root, text="Programming",width=20,font=("bold", 10))
label_4.place(x=85,y=330)
var1 = IntVar()
Checkbutton(root, text="java", variable=var1).place(x=235,y=330)
var2 = IntVar()
Checkbutton(root, text="python", variable=var2).place(x=290,y=330)

Button(root, text='Submit',width=20,bg='brown',fg='white').place(x=180,y=380)

root.mainloop()
```