**Code (d):**

#Write a python program to implement function overloading

print("---Default & Standard argument----")

def add(a,b,c=0,d=0):

print (a+b+c+d)

x=add(1,2)

y=add(1,2,3,4)

print('\n')

print("---Keyword argument----")

def add(args1,args2,\*pargs,\*\*kargs):

print (args1,args2,pargs,kargs)

d=add(1,2,3,4,5)

e=add(1,2,3,4,5,a=10,b=20,c=6)

print('\n')

print("---Positional argument----")

def mul(\*pargs):

res=1

for i in pargs:

res\*=i

print(res)

mul()

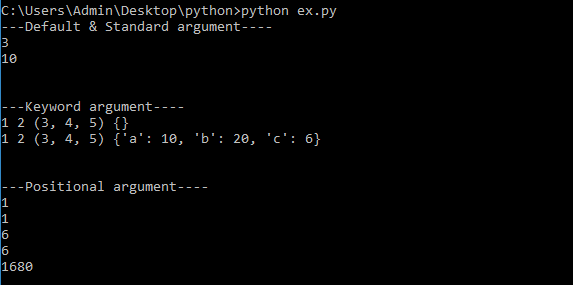
mul(1)

mul(2,3)

mul(2,1,3)

mul(8,7,6,5)

**Output:**

****

**Practical 4**

**Aim:**

1. **Write a program to randomly generate a list with 5 even numbers between 100 and 200 inclusive.**

**b)** **Write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console.**

**c) Write a program to implement binary search. Also print running time of this algorithm.**

**Code (a):**

# Write a program to randomly generate a list with 5 even numbers between 100 and 200 inclusive.

import random

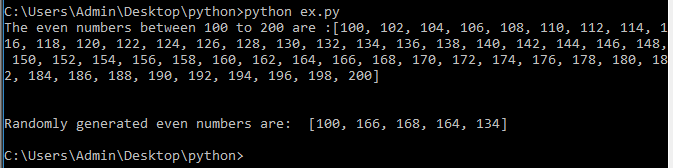
a=(list(filter(lambda x:x%2==0,range (100,201))))

print("The even numbers between 100 to 200 are :"+str(a))

print('\n')

print("Randomly generated even numbers are: ",random.sample(a,5))

**Output:**

****

**Code (b):**

**#**Write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console.

def yrange(n):

i = 1

while i < n:

if i%2==0:

yield i

i+=1

n=int(input("Enter the number: "))

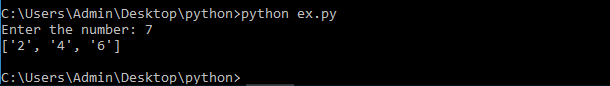
v=[]

for i in yrange(n):

v.append(str(i))

print(v)

**Output:**

****

**Code (c):**

#Write a program to implement binary search. Also print running time of this algorithm.

import time

s=time.time()

def binary\_sort(sorted\_list, length, key):

start = 0

end = length-1

while start <= end:

mid = int((start + end)/2)

if key == sorted\_list[mid]:

print("\nEntered number %d is present at position: %d" % (key, mid))

return -1

elif key < sorted\_list[mid]:

end = mid - 1

elif key > sorted\_list[mid]:

start = mid + 1

print("\nElement not found!")

return -1

lst = []

size = int(input("Enter size of list: \t"))

for n in range(size):

numbers = int(input("Enter any number: \t"))

lst.append(numbers)

lst.sort()

print('The list will be sorted, the sorted list is:', lst)

x = int(input("\nEnter the number to search: "))

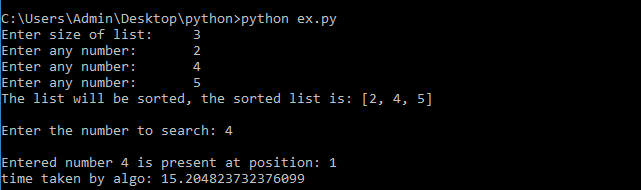
binary\_sort(lst, size, x)

e=time.time()

ex=e-s

print('time taken by algo:',ex)

**Output:**



**Practical 5**

**Aim:**

1. **Implement stack and queue in python.**

**b)** **Create a class Student which contains following members: Name, Enrolment number, Branch, Sem and Mobile number. Write a program to get record of a student and display its information on terminal.**

**Code (a):**

#Implement stack and queue in python.

#stack

print ("Stack")

s=[]

n=int(input("enter size:"))

for i in range(0,n):

n1=int(input("enter element:"))

s.append(n1)

print(s)

for i in range(0,n):

n2=s.pop()

print(str(n2),"poped")

#queue

print ("\nQueue")

l=[]

a=int(input("enter size:"))

for i in range(0,a):

a1=int(input("enter element:"))

l.insert(0,a1)

print(l)

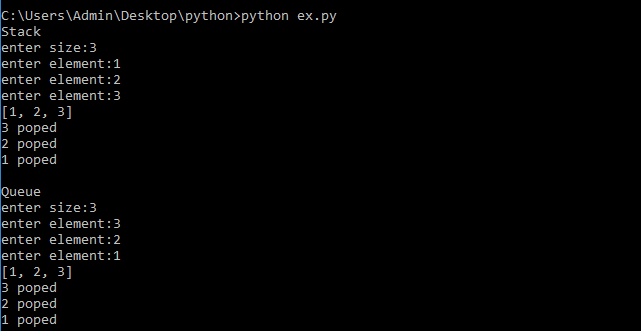
for i in range(0,a):

a2=l.pop()

print(str(a2),"poped")

print ("\n")

**Output:**

****

**Code (b):**

#Create a class Student which contains following members: Name, Enrolment number, Branch, Sem and Mobile number. Write a program to get record of a student and display its information on terminal.

class student :

def \_\_init\_\_(self,name,e\_no,branch,sem,m\_no):

self.name= name

self.e\_no=e\_no

self.branch=branch

self.sem=sem

self.m\_no=m\_no

def putdata(self):

print("Student name ",self.name)

print(" Enrolment number ",self.e\_no)

print(" Branch ",self.branch)

print(" Sem ",self.sem)

print(" Moblie number ",self.m\_no)

name=input("Enter student name:")

e\_no=int(input("Enter student Enrolment number:"))

branch=input("Enter student Branch:")

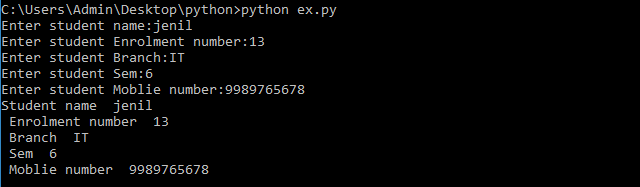
sem=int(input("Enter student Sem:"))

m\_no=int(input("Enter student Moblie number:"))

c = student(name,e\_no,branch,sem,m\_no)

c.putdata()

**Output:**

****