```
#-------------Ficha 10-----------------#SORT ALGORITHMS SORTED
FUNCTION E BINARY SEARCH
from collections import Counter
import csv
import math
#1
def openFile(x):
        with open(x,'r') as fileOp:
                text = fileOp.read()
        return text

"""x = input('File Name - ')
a = openFile(x).replace("\n","").split(" ")
print(sorted(a, key=Counter(a).get))"""

#3
def medi(l):
        l = sorted(l)
        b = len(l)
        if b%2 != 0:
                mediana = l[b//2]
        else:
                mediana = (l[int(b/2)]+l[int(b/2)-1])/2
        return mediana

"""ana = [1,2,3,4,5,6,7,8]
print(medi(ana))"""

#Application of the binary search
def binSearch(lst,x):
        first = 0
        last = len(lst)
        while first < last:
                mid = (first+last)//2
                if x <= lst[mid]:
                        last = mid
                else:
                        first = mid+1
        return first

def calcValue(x,yk,xk):
        a = binSearch(xk,x)
        print(a)
        if xk[a] != x:
                b = a-1
                print(b)
                result = yk[a]-yk[b]*xk[a]-xk[b]*(x-xk[b])+ yk[b]
        else:
                result = yk[a]
        return result

"""xk = [0,  1,   2,   3,   4,   6,   8,  10,  13,  16,   20]
yk = [0, 78, 156, 233, 309, 454, 588, 707, 853, 951, 1000]
x = float(input("X = "))
print('Y = ',calcValue(x,yk,xk))"""
```

```
#7
def function(x):
        y = x + math.sin(10*x)
        return y

def rootFinder(a,b,f,p=0.0001): #This function is pretty much a
deviation of the Binary Search
        #p = precision (p=0 would search for a value close to
infinity)
        mid = a+(b-a)/2
        while abs(f(mid)) > p:
                if (f(mid) > 0 and f(a)>0) or (f(mid)<0 and f(a)<0):
#situations in which a is equal to mid
                        a=mid
                else: #situations in which b is equal to mid
                        b=mid
                mid = a+(b-a)/2
        return z

#print('Zeros between 0.25 and 0.5 = ',rootFinder(0.25,0.5,function, p
= 0.0001))

############################################################3
#ALGORITHMS

#This is the binary search algorithm(finds medium value):
def binSearch(alist):
        first = 0
        last = len(lst)
        while first < last:
                mid = (first+last)//2
                if x <= lst[mid]:
                        last = mid
                else:
                        first = mid+1
        return first

"""alist = [54,26,93,17,77,31,44,55,20]
x = int(input())
binSearch(alist)
print(alist)"""


#############################################################
#ALTERNATIVES TO USING NORMAL PYTHON SORTED

#This is a BUBBLE SORT. It compares each value with the one infront
#to check if its bigger(if it is, then they swap places)
def bubbleSort(alist):
    for passnum in range(len(alist)-1,0,-1):
        for i in range(passnum):
            if alist[i]>alist[i+1]:
                temp = alist[i]
                alist[i] = alist[i+1]
                alist[i+1] = temp

"""
```

```
alist = [54,26,93,17,77,31,44,55,20]
bubbleSort(alist)
print(alist)"""

#This is a SELECTION SORT. It goes through a list and fints it biggest
#Value. then it puts that value at the end of the list.Keeps repeating
def selectionSort(alist):
   for fillslot in range(len(alist)-1,0,-1):
       positionOfMax=0
       for location in range(1,fillslot+1):
           if alist[location]>alist[positionOfMax]:
               positionOfMax = location

       temp = alist[fillslot]
       alist[fillslot] = alist[positionOfMax]
       alist[positionOfMax] = temp

"""alist = [54,26,93,17,77,31,44,55,20]
selectionSort(alist)
print(alist)"""

#This is the INSERTION ALGORITHM. It keeps track of a value and sees
#if the one behind it is bigger. if it is then they swap places
def insertionSort(alist):
   for index in range(1,len(alist)):

     currentvalue = alist[index]
     position = index

     while position>0 and alist[position-1]>currentvalue:
         alist[position]=alist[position-1]
         position = position-1

     alist[position]=currentvalue

"""alist = [54,26,93,17,77,31,44,55,20]
insertionSort(alist)
print(alist)"""
```