import array as ary

class Sortings:

    def insertion\_sort(self):

        arr= ary.array('i', [5, 2, 3, 1, 4])

        for i in range(1, len(arr)):

            temp = arr[i]

            j = i-1

            while(j >= 0 and arr[j] > temp):

                arr[j+1] = arr[j]

                j-=1

            arr[j+1] = temp

        print(arr)

    def bubble\_sort(self):

        arr= ary.array('i', [5, 2, 3, 1, 4])

        for i in range(len(arr)):

            for j in range(0,len(arr)-i-1):

                if(arr[j]>arr[j+1]):

                    temp=arr[j]

                    arr[j]=arr[j+1]

                    arr[j+1]=temp

        print(arr)

    def \_\_swap(self,arr,f,m):

        temp=arr[f]

        arr[f]=arr[m]

        arr[m]=temp

    def selection\_sort(self):

        A= ary.array('i', [5, 2, 3, 1, 4])

        l=len(A)

        for i in range(0,l-1):

            min=i

            for j in range(i+1,l):

                if(A[j]<A[min]):

                    min=j

            if(min!=i):

                self.\_\_swap(A,i,min)

        print(A)

    def \_\_partition(self,array,f,l):

        p=array[f]

        i=f+1

        j=l

        while(i<=j):

            while(i<=j and p>array[i]):

                i+=1

            while(i<=j and p<array[j]):

                j-=1

            if(i<=j):

                self.\_\_swap(array,i,j)

        self.\_\_swap(array,f,j)

        return j

    def quick\_sort(self,array,f,l):

        if(f<=l):

            key=self.\_\_partition(array,f,l)

            self.quick\_sort(array,f,key-1)

            self.quick\_sort(array,key+1,l)

        return array

    def \_\_merge(self,arr, f , mid , l):

        i = f

        j = mid+1

        n1 = mid

        n2 = l

        ans = []

        while(i<=n1 and j<=n2):

            if(arr[i] < arr[j]):

                ans.append(arr[i])

                i+=1

            else:

                ans.append(arr[j])

                j+=1

        while(i<=n1):

            ans.append(arr[i])

            i+=1

        while(j<=n2):

            ans.append(arr[j])

            j+=1

        k = 0

        for i in range(f, l+1):

            arr[i] = ans[k]

            k+=1

    def mergeSort(self,arr, f , l):

        if(f >= l):

            return

        mid = int((f+l)/2)

        self.mergeSort(arr, f ,mid)

        self.mergeSort(arr, mid+1, l)

        self.\_\_merge(arr, f , mid , l)

        return arr

obj=Sortings()

print("Insertion sort:")

obj.insertion\_sort()

print("Bubble sort:")

obj.bubble\_sort()

print("Selection sort:")

obj.selection\_sort()

print("Quick sort:")

A=ary.array('i',[])

size=int(input("enter the size of the array:"))

print("Enter the elemnts:")

for i in range(size):

    A.append(int(input()))

print(obj.quick\_sort(A,0,len(A)-1))

print("Merge sort:")

ary1 = ary.array('i', [])

size = int(input("Enter size of array: "))

print("Enter the elemnts:")

for i in range(size):

    ary1.append(int(input()))

print(obj.mergeSort(ary1, 0, size-1))