

## ASSIGNMENT:-4

```

import java.math.BigInteger;
import java.util.Scanner;

class ab24601_A4
{
    static int i_index;
    static int j_index;

    public static void main(String[] args)
    {
        int n;
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the Value of N: ");
        n = scan.nextInt();
        System.out.println("Enter the Base and Exponent for "+n+" rows");
        int merged[][]=new int[n][2];
        int audience[][] = new int[n][2];
        for (int i = 0; i < n; i++)
        {
            audience[i][0] = scan.nextInt();
            audience[i][1] = scan.nextInt();
        }
        System.out.println("Enter Number of Rows to compare");
        int m=scan.nextInt();
        ab24601_A4 obj = new ab24601_A4();
        for (int i = 0; i < m; i++)
        {
            System.out.println("Enter indices of row1 and row2 that you want to compare: ");
            int row1_index = scan.nextInt();
            int row2_index = scan.nextInt();
            int x=obj.comparator(audience, row1_index, row2_index);
            if(x==1)
            {
                System.out.println("ecode of index 1 is greater than ecode of index2");
            }
            else if(x==0)
            {
                System.out.println("ecode of index 1 is less than ecode of index2");
            }
            else
            {
                System.out.println("ecode of index 1 is equal to ecode of index2");
            }
        }
        i_index=0;
        j_index=n-1;
        obj.sorting(audience,n);
        obj.merge(audience,merged,n);
        System.out.println("The merged array is:");
        System.out.println();
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<2;j++)
            System.out.print(merged[i][j]+" ");
            System.out.println();
        }
    }

    void sorting(int audience[][],int N)
    {
        /*int x=2;
        while((comparator(audience, j_index, j_index))==1)
        {

        }*/
        int temp1,temp2;

```

```

while((comparator(audience,i_index,i_index+1)==1)&&i_index<N-1)
    i_index=i_index+1;
    System.out.println("THE i_index: "+i_index);

while((comparator(audience,j_index,j_index-1)==-1)&&j_index>0)
    j_index=j_index-1;
    System.out.println("The j_index: "+j_index);

for(int i=i_index+1;i<j_index-1;i++)
{
    for(int j=i_index+1;j<j_index-1;j++)
    {
        if(comparator(audience,j,j+1)!=1)
        {
            temp1=audience[j][0];
            temp2=audience[j][1];
            audience[j][0]=audience[j+1][0];
            audience[j][1]=audience[j+1][1];

            audience[j+1][0]=temp1;
            audience[j+1][1]=temp2;
        }
    }
}

void merge(int audience[][],int merged[][],int N)
{
    int i=0,j=j_index,k=0;
    while((i<=i_index)&&(j<=N-1))
    {
        if(comparator(audience,i,j)==1){
            merged[k][0]=audience[i][0];
            merged[k][1]=audience[i][1];
            i++;}

        else{
            merged[k][0]=audience[j][0];
            merged[k][1]=audience[j][1];
            j++;
        }
        k++;
    }
    while(i<=i_index)
    {
        merged[k][0]=audience[i][0];
        merged[k][1]=audience[i][1];
        k++;
        i++;
    }

    while(j<=N-1)
    {
        merged[k][0]=audience[j][0];
        merged[k][1]=audience[j][1];
        k++;
        j++;
    }

    for(int p=i_index+1;p<j_index;p++)
    {
        merged[k][0]=audience[p][0];
        merged[k][1]=audience[p][1];
        k++;
    }
}

int comparator(int audience[][], int index1, int index2)
{
    BigInteger index_e1 = new BigInteger(""+audience[index1][0]);
    BigInteger index_e2 = new BigInteger(""+audience[index2][0]);

```

```
BigInteger mod_by = new BigInteger("100");
BigInteger ref, encode1, encode2;
ref = index_e1.pow(audience[index1][1]);
encode1 = ref.mod(mod_by);
ref = index_e2.pow(audience[index2][1]);
encode2 = ref.mod(mod_by);
int x = encode1.compareTo(encode2);
//return -x;
/* double encode1= Math.pow( audience[index1][0], audience[index1][1])%100;
double encode2 = Math.pow( audience[index2][0], audience[index2][1])%100;
System.out.println("Enc "+encode1+" Enc2 "+encode2 +" Power" +Math.pow(
audience[index2][0], audience[index2][1]));
*/
if(x==1)
{
    //System.out.println("ecode of index 1 is greater than ecode of index2");
    return -1;
}
else if(x==-1)
{
    //System.out.println("ecode of index 1 is less than ecode of index2");
    return 1;
}
else
{
    //System.out.println("ecode of index 1 is equal to ecode of index2");
    return 0;
}
}
```