```java

/**
  A PROGRAM TO INPUT THE SIZE OF A DOUBLE DIMENSIONAL ARRAY (m * n ).
  i.e, m & n > 2 but m & n < 20.
  ARRANGE THE ARRAY ELEMENTS IN ACENDING ORDER AND PRINT THE BOUNDARY ELEMENTS ONLY
  IN A DOUBLE DIMENSIONAL FORM.
 */
//SUCHIT TE XII A
import java.util.*;
public class DDABoundary
{
    public static void main()
    {
        Scanner sc = new Scanner (System.in);
        System.out.println("A PROGRAM TO FIND THE SUM AND PRINT THE BOUNDARY ELEMENTS OF A DOUBLE DIMENSIONAL ARRAY");
        System.out.println("PLEASE ENTER THE SIZE OF THE DDA SUCH THAT THE no.of ROWS AND no.of COLUMNS SHOULD BE");
        System.out.println("GREATER THAN 2 BUT LESSER THAN 20");
        System.out.println("ENTER ROW SIZE : ");
        int row = sc.nextInt();
        System.out.println("ENTER COLUMN SIZE : ");
        int column = sc.nextInt();
        if((row<=2||row>=20)||(column<=2||column>=20))
        {
            System.out.println("INVALID INPUT!");
            System.exit(0);
        }
        int arr[][]=new int[row][column];
        int temp[]=new int[row*column];
        int ctr1 = 0;
        //INPUT FROM USER
        System.out.println("ENTER THE ELEMENTS OF THE ARRAY");
        for(int i=0;i<row;i++)
        {
            System.out.println("FOR ROW "+(i+1));
            for(int j=0;j<column;j++)
            {
                arr[i][j]=sc.nextInt();
                temp[ctr1]=arr[i][j];   // STORING DDA IN SINGLE DIMENSIONAL FORM.
                ctr1++;
            }
        }
        //SORTING IN ASCENDING ORDER.
        for(int i=0;i<temp.length-1;i++)
        {
            for(int j=(1+i);j<temp.length;j++)
            {
                if(temp[j]<temp[i])
                {
                    int hold=temp[i];
                    temp[i]=temp[j];
                    temp[j]=hold;
                }
            }
        }
        //PRINT ORIGINAL ARRAY
        System.out.println("ORIGINAL MATRIX");
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<column;j++)
            {
                System.out.print(arr[i][j]+"  ");
            }
            System.out.println();
        }
        //TO REPLACE OLD DDA ELEMENTS IN ASCENDING ORDER
        int ctr2=0;
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<column;j++)
            {
                arr[i][j]=temp[ctr2];
                ctr2++;
            }
            System.out.println();
        }
        //PRINT ARRAY WITH ELEMENTS IN ASCENDING ORDER
        System.out.println("NEW MATRIX");
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<column;j++)
            {
                System.out.print(arr[i][j]+"  ");
                System.out.print("\t");
            }
            System.out.println();
        }
        int finale[][]=new int[row][column];
        int sumBoundary=0;
        //TO PRINT AND FIND THE SUM OF BOUNDARY ELEMENTS
        System.out.println("BOUNDARY ELEMENTS");
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<column;j++)
            {
                if(i==0||i==(row-1)||j==0||j==(column-1))
                {
                    System.out.print(arr[i][j]+"  ");
                    System.out.print("\t");
                    sumBoundary=sumBoundary+arr[i][j];
                }
                else
                {
                    System.out.print("   ");
                }
            }
            System.out.println();
        }
        System.out.println("THE SUM OF THE BOUNDARY ELEMENTS ARE : "+s
umBoundary);
    }
}
```