import java.io.\*;

class MedainOfSortedArray

{

static int []a = new int[]{100};

static int []b = new int[]{11, 12, 14};

// Function to find max

static int maximum(int a, int b)

{

return a > b ? a : b;

}

// Function to find minimum

static int minimum(int a, int b)

{

return a < b ? a : b;

}

static double findMedianSortedArrays(int n,

int m)

{

int min\_index = 0,

max\_index = n, i = 0,

j = 0, median = 0;

while (min\_index <= max\_index)

{

i = (min\_index + max\_index) / 2;

j = ((n + m + 1) / 2) - i;

if (i < n && j > 0 && b[j - 1] > a[i])

min\_index = i + 1;

else if (i > 0 && j < m && b[j] < a[i - 1])

max\_index = i - 1;

// we have found the desired halves.

else

{

if (i == 0)

median = b[j - 1];

else if (j == 0)

median = a[i - 1];

else

median = maximum(a[i - 1],

b[j - 1]);

break;

}

}

if ((n + m) % 2 == 1)

return (double)median;

if (i == n)

return (median + b[j]) / 2.0;

if (j == m)

return (median + a[i]) / 2.0;

return (median + minimum(a[i],

b[j])) / 2.0;

}

public static void main(String args[])

{

int n = a.length;

int m = b.length;

if (n < m)

System.out.print("The median is : " +

findMedianSortedArrays(n, m));

else

System.out.print("The median is : " +

findMedianSortedArrays(m, n));

}

}