Najmudin Rain

[Najmudinrain786@gmail.com](mailto:Najmudinrain786@gmail.com)

RegNo: 18BCE2505

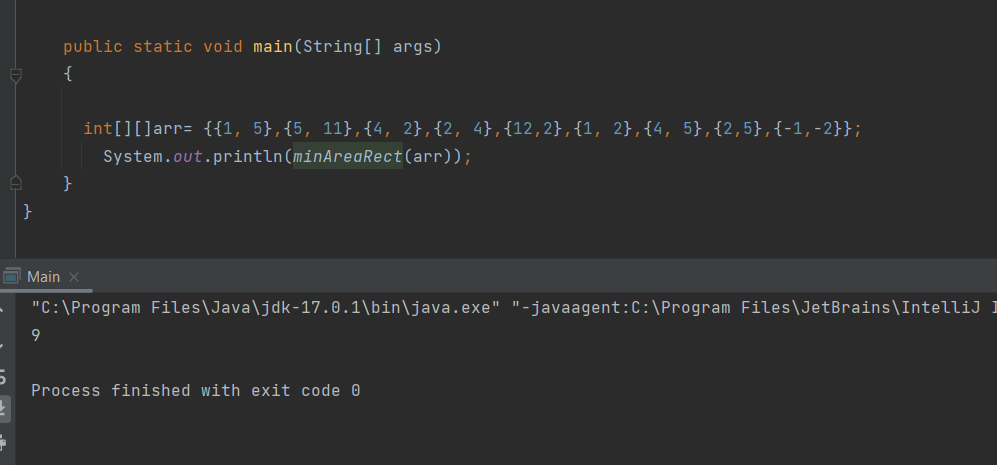
**Problem-1**

**Minimum area of rectangle:**

import java.util.\*;  
  
class Main {  
  
 // function to find minimum area of Rectangle  
 public static int minAreaRect(int[][] points) {  
 Map<Integer, Set<Integer>> map = new HashMap<>();  
 for (int[] p : points) {  
 if (!map.containsKey(p[0])) {  
 map.put(p[0], new HashSet<>());  
 }  
 map.get(p[0]).add(p[1]);  
 }  
 int min = Integer.*MAX\_VALUE*;  
 for (int[] p1 : points) {  
 for (int[] p2 : points) {  
 if (p1[0] == p2[0] || p1[1] == p2[1]) { // if have the same x or y  
 continue;  
 }  
 if (map.get(p1[0]).contains(p2[1]) && map.get(p2[0]).contains(p1[1])) { // find other two points  
 min = Math.*min*(min, Math.*abs*(p1[0] - p2[0]) \* Math.*abs*(p1[1] - p2[1]));  
 }  
 }  
 }  
 return min == Integer.*MAX\_VALUE* ? 0 : min;  
 }  
  
 public static void main(String[] args)  
 {  
  
 int[][]arr= {{1, 5},{5, 11},{4, 2},{2, 4},{12,2},{1, 2},{4, 5},{2,5},{-1,-2}};  
 System.*out*.println(*minAreaRect*(arr));  
 }  
}

Snapshots:





**Problem:2**

Calculate the number of non-empty consecutive subsequence

of size K of the array A, such that the value of the sum is even.

Code:

public class NoOfEvenSum {  
 public static void main(String[]args){  
 int []arr={1,-3,0,5,7};  
 int k=2;  
 System.*out*.println(*NumEvenSum*(arr,k));}  
 static int NumEvenSum(int arr[], int k){  
 int count=0;  
 int len=arr.length;  
 for(int i=0; i<len-k+1;i++){  
 int j=0,sum=0;  
 int ind=i  
 while(j<k &&ind<len){  
 sum= sum+arr[ind];  
 ind++;  
 j++;}  
 if(sum%2==0)count++;}  
 return count;}  
}

snapshots:

