**Divide and Conquer**

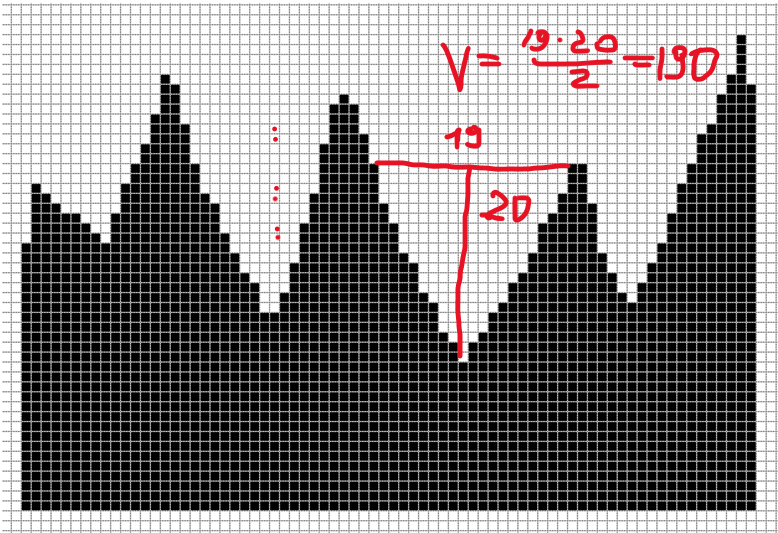
Picture 1 A picture containing text, tiled, net, tile

Description automatically generated

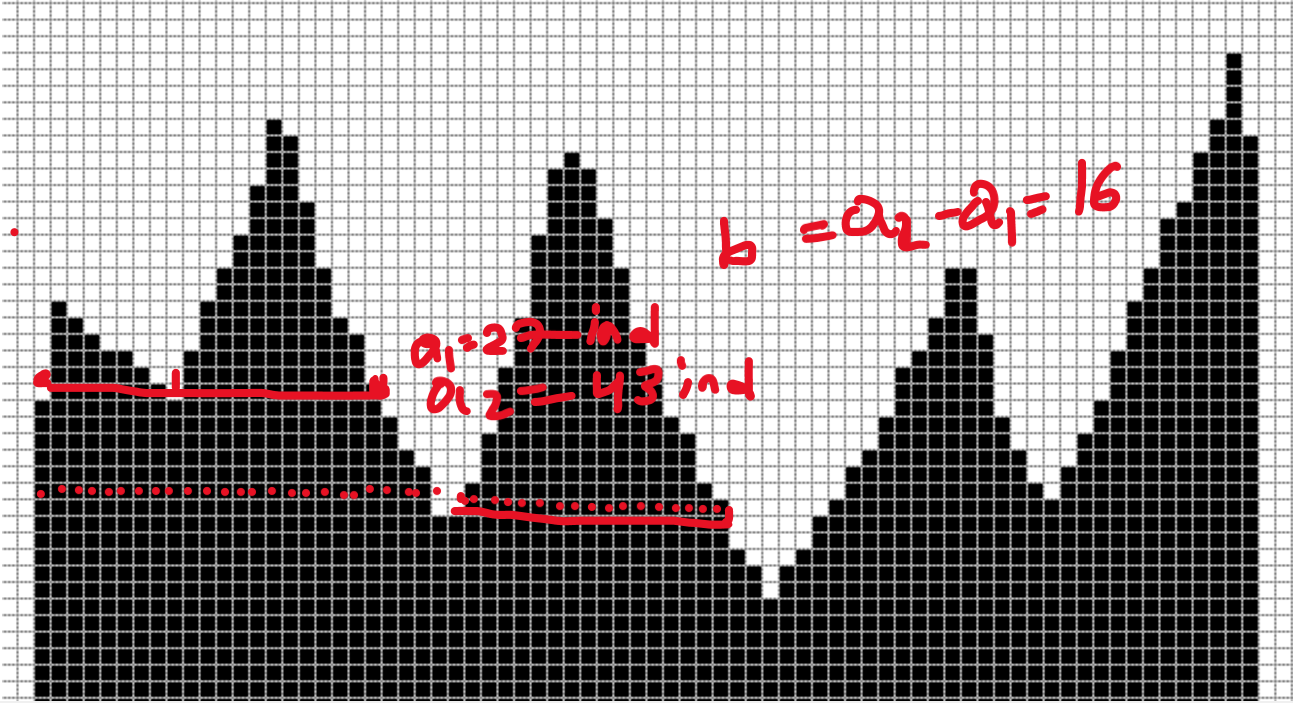


1. You are given an integer N and an array of N integers. The data is a cubed profile of a mountain range in a flat world (Figure 1). Consider a lake of water bounded between two peaks with sufficient height and not divided into parts. The volume of a lake is ***counted*** in the number of squares it consists of. For a given profile of mountains, output the maximum possible volume of one lake that can appear in this profile.

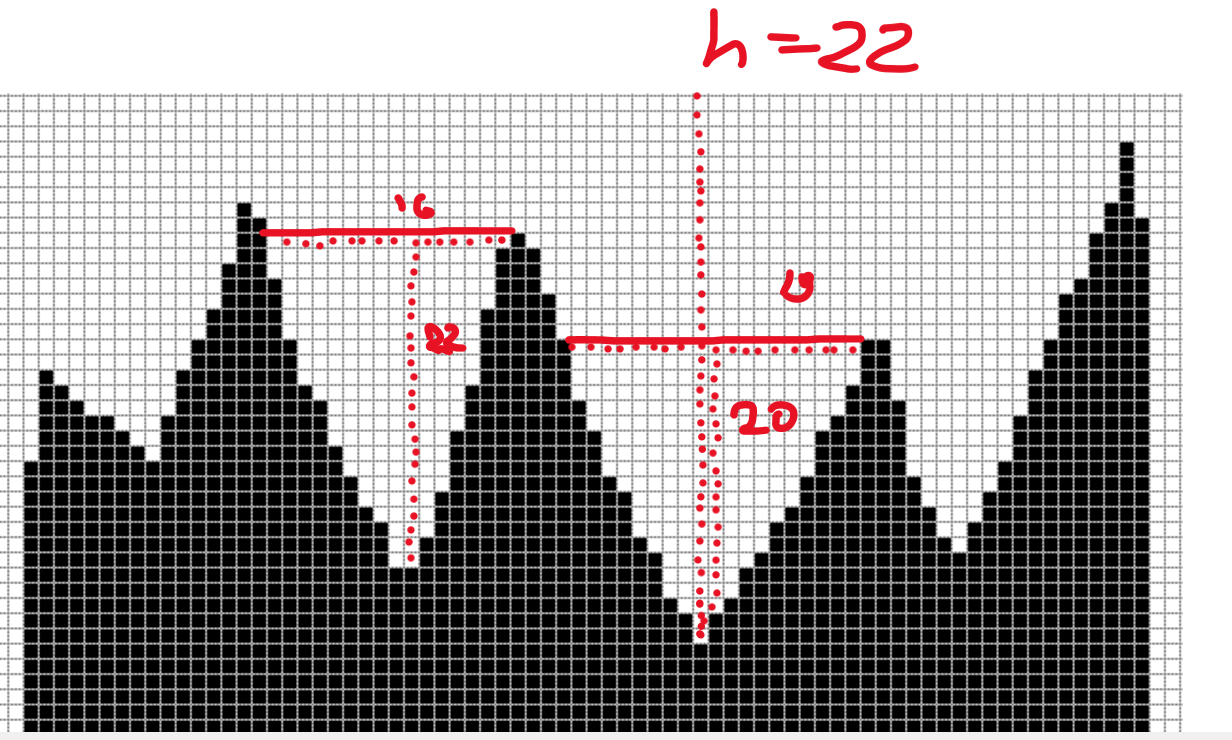




import java.util.Scanner;  
  
public class Task1 {  
 public static int max(int arr[],int height[],int n){  
 int max=0;  
 int temp=0;  
 for (int i = 0; i < n; i++) {  
 temp = (arr[i]\*i\* height[i]) / 2;  
 if (temp > max) {  
 max = temp;  
 }  
 }  
 return max;  
 }  
 public static void main (String[] args) {  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.println("Please enter length");  
 int n=input.nextInt();  
 int [] arr = new int[n];  
 int [] height=new int[n];  
 for (int i = 0; i < n; i++)  
 {  
 System.*out*.println("Please enter number");  
 int temp = input.nextInt();  
 arr[i]=temp;  
 System.*out*.println("Please enter height");  
 int s = input.nextInt();  
 height[i]=s;  
 }  
 System.*out*.println( *max*(arr,height,n));  
 }  
}

1. You are given an integer N and an array of N integers. The data is a cubed profile of a mountain range in a flat world (Figure 1). Dwarfs build their mines in these mountains to extract ore. These shafts always run horizontally, and at the same time from both sides they must go to the surface (not going beyond the array). It is believed that the mine produces ore in an amount proportional to the volume of rocks above the mine, this volume is calculated by the number of cubes. Print the indices of 2 array elements between which you can draw the most profitable gnome mine. 
2. You are given an integer N and an array of N integers. The data is a profile of ice icicles on the roof slope (Figure 1, white part). The icicle can only break off along a horizontal line, and the length of the icicle must be greater than or equal to the width. The ends of the array are not counted as the edges of the icicle. What is the longest icicle that can fall?

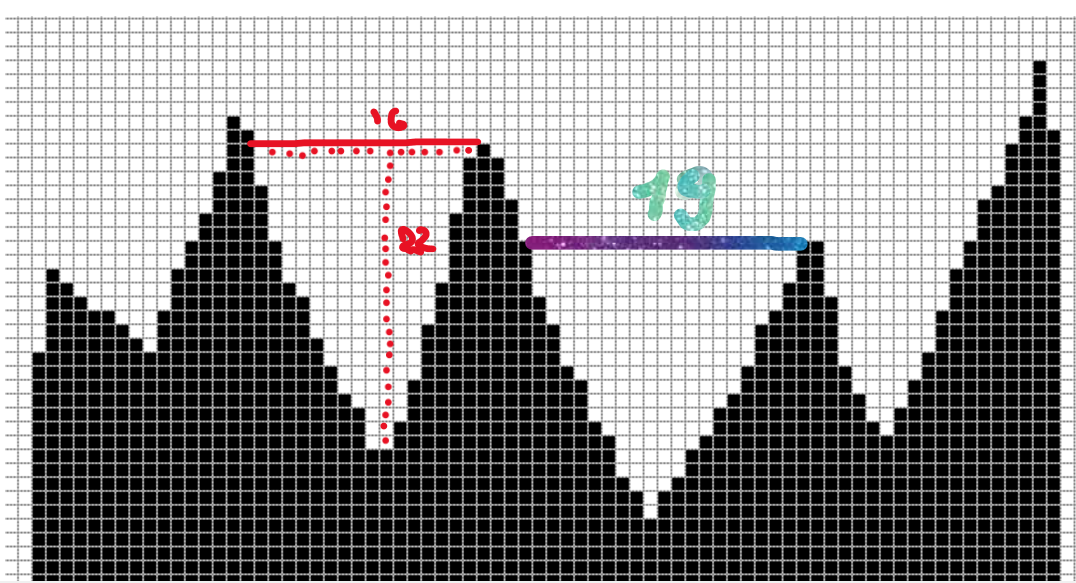




import java.awt.\*;  
import java.util.Scanner;  
  
public class Task3 {  
 double x, y;  
  
 void Point(double x, double y) {  
 this.x = x;  
 this.y = y;  
 }  
  
 public double getX() {  
 return x;  
 }  
  
 public double getY() {  
 return y;  
 }  
  
 public static class Line {  
 Point p1, p2;  
 Line(Point point1, Point point2) {  
 this.p1 = point1;  
 this.p2 = point2;  
  
 }  
  
 @Override  
 public String toString() {  
 return "Line{" +  
 "p1=" + p1 +  
 ", p2=" + p2 +  
 '}';  
 }  
  
 public double canad() {  
 if (Math.*abs*(p2.getY() -p1.getY())>Math.*abs*(p1.getX() - p2.getX())) {  
 return Math.*abs*(  
 (p2.getY() -p1.getY())  
 );  
 } else return 0;}  
  
 public static void main(String args[]) {  
 int N=2;  
 int[] length=new int[N];  
 Scanner input = new Scanner(System.*in*);  
 Point[] d=new Point[N];  
 for (int i = 0; i < N; i++) {  
 d[i]=new Point(input.nextInt(), input.nextInt());  
 }  
  
 Line[] lines=new Line[N];  
 Line s=new Line(new Point(0,0),new Point(0,0));  
 for(int i=0;i<N-1;i++) {  
 for (int j = 0; j < N; j++) {  
 s = new Line(new Point((int) d[i].getX(), (int) d[i].getY()), new Point((int) d[j].getX(), (int) d[j].getY()));  
 }  
 }  
 System.*out*.println(s.toString());  
  
 double as = s.canad();  
 System.*out*.println("Length = " + as);  
 }  
 }  
}

4. You are given an integer N and an array of N integers. The data is a cubed profile of a mountain range in a flat world (Figure 1). Let the goat move through the mountains from left to right. Because of the wind and sun, the grass grows only on the southern side of the mountains (let it be on the right for us). Therefore, if in a step to the left to the right the goat rises higher, it will lose one energy, if it goes down, it will receive one energy, if the level does not change, then the energy level of the goat will not change. Where will he get the most energy benefit? Its energy indicator on a segment can have negative values, because he has a sufficient initial supply of energy.

5. Задано целое число N и массив из N целых чисел. Данные представляют собой профиль горной гряды в плоском мире в виде кубиков (Рисунок 1). Канатоходец хочет поставить рекорд и пройти по канату максимальную дистанцию. Он может протянуть канат параллельно земле, между двумя точками на поверхности горы, если ему ничего не мешает. По заданному профилю гор найдите максимально длинный канат, который он может протянуть.

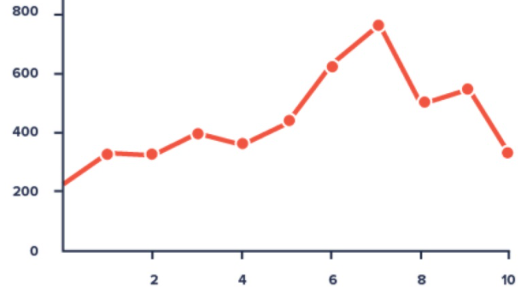


import java.awt.\*;  
import java.util.Scanner;  
  
public class Task5 {  
 double x, y;  
  
 void Point(double x, double y) {  
 this.x = x;  
 this.y = y;  
 }  
  
 public double getX() {  
 return x;  
 }  
  
 public double getY() {  
 return y;  
 }  
  
 public static class Line {  
 Point p1, p2;  
 Line(Point point1, Point point2) {  
 this.p1 = point1;  
 this.p2 = point2;  
  
 }  
  
 @Override  
 public String toString() {  
 return "Line{" +  
 "p1=" + p1 +  
 ", p2=" + p2 +  
 '}';  
 }  
  
 public double canad() {  
 if (p1.getY() == p2.getY()) {  
 return Math.*abs*(  
 (p1.getX() - p2.getX())  
 );  
 } else return 0;  
 }  
  
 public static void main(String args[]) {  
 int N=5;  
 int[] length=new int[N];  
 Scanner input = new Scanner(System.*in*);  
 Point[] d=new Point[N];  
 for (int i = 0; i < N; i++) {  
 d[i]=new Point(input.nextInt(), input.nextInt());  
 }  
  
 Line[] lines=new Line[N];  
 Line s=new Line(new Point(0,0),new Point(0,0));  
 for(int i=0;i<N-1;i++) {  
 for (int j = 0; j < N; j++) {  
 if (d[i].getY() == d[j].getY()) {  
 s = new Line(new Point((int) d[i].getX(), (int) d[i].getY()), new Point((int) d[j].getX(), (int) d[j].getY()));  
 }  
 }  
 }  
 System.*out*.println(s.toString());  
  
 double as = s.canad();  
 System.*out*.println("Length = " + as);  
 }  
 }  
}

6. You are given an integer N and an array of N integers. The data is a cubed profile of a mountain range in a flat world (Figure 1). Let the camel move through the mountains from left to right. We assume that there is water only in ravines, that is, at points of local minimum. For each step, the camel spends 1 unit of water, if at the same time it rises higher, then it spends another 1 unit of water. At the points of the local minimum, it is filled with 10 units of water. Find where it will get the most water benefit. It can be assumed that he will never be dehydrated, because it has an initial hump headroom.

Approximately 115.

import java.util.Vector;  
  
public class Task6 {  
 public static int localMin(int n,  
 int[] arr, int step) {  
 Vector<Integer> mn = new Vector<Integer>();  
 int temp = 0;  
  
  
 if (arr[0] < arr[1]) {  
 mn.add(0);  
 temp += 2;  
 }  
  
   
 for (int i = 1; i < n - 1; i++) {  
 if ((arr[i - 1] > arr[i]) &&  
 (arr[i] < arr[i + 1])) {  
 mn.add(i);  
 temp += 2;  
 step+=10;  
 }  
 }  
 return step;  
   
 }  
  
 public int descend(int step) {  
  
 step += 1;  
 return step;  
 }  
  
 public int сlimb(int step) {  
 step += 1;  
 return step;  
 }  
  
 public int Step(int step) {  
 step += 1;  
 return step;  
 }  
}

Figure 2 

7. You are given an integer N and an array of N integers. These numbers represent a graph of the function (Figure 2) with integer values ​​in integer coordinates. In data analysis, it is often necessary to find cases of stability of some indicator, i.e. minimal changes over time. Find the longest interval in which the values ​​of the function change by no more than 3 (the difference between the maximum and minimum).

8. You are given an integer N and an array of N integers. These numbers represent a graph of the function (Figure 2) with integer values ​​in integer coordinates. In data analysis, it is often necessary to find cases of fluctuations of some indicator, i.e. frequent changes in a short time. Let the fluctuation indicator be calculated as follows, for each value in the integer coordinate, which is a local maximum or minimum, 2 points are added, for those that are not such, 1 point is subtracted. Find the interval in which the fluctuation index is maximum.

9. You are given an integer N and an array of N integers. These numbers represent a graph of the function (Figure 2) with integer values ​​in integer coordinates. We will consider a special efficiency indicator as follows, efficiency +2: if the function has increased in a step by more than 5%, efficiency is -2: if the function has decreased in a step by more than 5%, and -1: if the function has changed by less than 5% ... During what period was the growth efficiency at its maximum?

10. You are given an integer N and an array of N integers. These numbers represent a graph of the function (Figure 2) with integer values ​​in integer coordinates. In data science, there is often a need to identify anomalies. Consider the following method: for adjacent two elements, enter the anomaly indicator equal to 4, if their difference is more than the difference in absolute value of the previous pair by more than 3 times, otherwise the anomaly indicator for this pair is considered equal to -1. Find the area with the highest concentration of anomalies, i.e. at which the sum of the anomaly indicators will be maximum.