**Taxi Booking**

**Easy**

You are going to book a taxi. There are infinite number of points 1, 2, 3... on the X axis and your current position is **cur**. There are **N** Taxis near you, and the position of those taxis is given as an array **pos**. Where **pos[i]** denotes the position of the **ith** taxi. You are also given an array **time**. Where **time[i]** denotes the time taken by the **ith** taxi to cover **1 unit** of distance. Your task is to find the **minimum** time to board a taxi.

**Example 1:**

**Input:**

N = 3, cur = 4

pos = [1, 5, 6]

time = [2, 3, 1]

**Output:**

2

**Explanation:**

Total time taken by the 1st taxi will be: (4-1)\*2 = 6

Total time taken by the 2nd taxi will be: (5-4)\*3 = 3

Total time taken by the 3rd taxi will be: (6-4)\*1 = 2

So, the minimum time will be 2 sec.

**Example 2:**

**Input:**

N = 2, cur = 1

pos = [1, 6]

time = [10, 3]

**Output:**

0

**Explanation:**

Total time taken by the 1st taxi will be: (1-1)\*10 = 0

Total time taken by the 2nd taxi will be: (6-1)\*3 = 15

So, the minimum time will be 0 sec.

//{ Driver Code Starts

import java.io.\*;

import java.util.\*;

class IntArray

{

public static int[] input(BufferedReader br, int n) throws IOException

{

String[] s = br.readLine().trim().split(" ");

int[] a = new int[n];

for(int i = 0; i < n; i++)

a[i] = Integer.parseInt(s[i]);

return a;

}

public static void print(int[] a)

{

for(int e : a)

System.out.print(e + " ");

System.out.println();

}

public static void print(ArrayList<Integer> a)

{

for(int e : a)

System.out.print(e + " ");

System.out.println();

}

}

class CodingMaxima {

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t;

t = Integer.parseInt(br.readLine());

while(t-- > 0){

int N;

N = Integer.parseInt(br.readLine());

int cur;

cur = Integer.parseInt(br.readLine());

int[] pos = IntArray.input(br, N);

int[] time = IntArray.input(br, N);

Solution obj = new Solution();

int res = obj.minimumTime(N, cur, pos, time);

System.out.println(res);

}

}

}

// } Driver Code Ends

class Solution {

public static int minimumTime(int N, int cur, int[] pos, int[] time) {

int ans=Integer.MAX\_VALUE;

for(int i=0;i<N;i++){

ans=Math.min(ans,Math.abs(cur-pos[i])\*time[i]);

}

return ans;

}

}