A group of cows grabbed a truck and ventured on an expedition deep into the jungle. Being rather poor drivers, the cows unfortunately managed to run over a rock and puncture the truck's fuel tank. The truck now leaks one unit of fuel every unit of distance it travels.   
  
To repair the truck, the cows need to drive to the nearest town (no more than 1,000,000 units distant) down a long, winding road. On this road, between the town and the current location of the truck, there are N (1 <= N <= 10,000) fuel stops where the cows can stop to acquire additional fuel (1..100 units at each stop).   
  
The jungle is a dangerous place for humans and is especially dangerous for cows. Therefore, the cows want to make the minimum possible number of stops for fuel on the way to the town. Fortunately, the capacity of the fuel tank on their truck is so large that there is effectively no limit to the amount of fuel it can hold. The truck is currently L units away from the town and has P units of fuel (1 <= P <= 1,000,000).   
  
Determine the minimum number of stops needed to reach the town, or if the cows cannot reach the town at all.

Input

\* Line 1: A single integer, N   
  
\* Lines 2..N+1: Each line contains two space-separated integers describing a fuel stop: The first integer is the distance from the town to the stop; the second is the amount of fuel available at that stop.   
  
\* Line N+2: Two space-separated integers, L and P

Output

\* Line 1: A single integer giving the minimum number of fuel stops necessary to reach the town. If it is not possible to reach the town, output -1.

Sample Input

4

4 4

5 2

11 5

15 10

25 10

Sample Output

2

Hint

INPUT DETAILS:   
  
The truck is 25 units away from the town; the truck has 10 units of fuel. Along the road, there are 4 fuel stops at distances 4, 5, 11, and 15 from the town (so these are initially at distances 21, 20, 14, and 10 from the truck). These fuel stops can supply up to 4, 2, 5, and 10 units of fuel, respectively.   
  
OUTPUT DETAILS:   
  
Drive 10 units, stop to acquire 10 more units of fuel, drive 4 more units, stop to acquire 5 more units of fuel, then drive to the town.

题意：一辆卡车要行驶L单位距离。最开始时，卡车上有P单位汽油，每向前行驶1单位距离消耗1单位汽油。如果在途中车上的汽油耗尽，卡车就无法继续前行，即无法到达终点。途中共有N个加油站，加油站提供的油量有限，卡车的油箱无限大，无论加多少油都没问题。给出每个加油站距离终点的距离和能够提供的油量，问卡车从起点到终点至少要加几次油？如果不能到达终点，输出-1。

分析：由于N比较大，应该找一个高效的解法。稍微转换一下思考方式：在卡车开往终点的途中，只有在加油站才可以加油。但是，如果认为“在到达加油站i时，就获得了一次在之后的任何时候都可以加Bi单位汽油的权利”，在解决问题上也是一样的。而在之后需要加油时，就认为是在之前经过的加油站加的油就可以了。因为希望加油次数尽可能少，所以当燃料为0了之后再加油是最好的选择。基于贪心的思想，当燃料为0时，选择能加油量最大的加油站。所以可以用一个优先队列来保存经过的加油站的油量，当需要加油时，取出队列中的最大元素即可。

#include<iostream>

#include<stdio.h>

#include<cstring>

#include<algorithm>

#include<queue>

using namespace std;

struct node

{

int x,v;

};

bool cmp(node a,node b)

{

return a.x<b.x;

}

int main()

{

int n,L,P,i;

node a[10010];

cin>>n;

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

cin>>a[i].x>>a[i].v;

cin>>L>>P;

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

a[i].x=L-a[i].x;//转化为加油站到起点的距离

a[n].x=L;

a[n].v=0;

n++;//把终点也认为是加油站

sort(a,a+n,cmp);

priority\_queue<int,vector<int>,less<int> > q; //值大的优先

int pos=0;//卡车所在的上一个位置

int res=P; //剩余油量

int ans(0);//加油次数

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

{

int dis=a[i].x-pos; //从上个加油站到此位置要行驶的距离

while(res<dis)//剩余油量不能到达此位置

{

if(q.empty()) //没有油可加

{

ans=-1;

break;

}

else

{

res+=q.top();

q.pop();

ans++;

}

}

if(ans==-1)break;

res-=dis;

pos=a[i].x;

q.push(a[i].v);

}

// cout<<i<<endl;

cout<<ans<<endl;

return 0;

}