Hammad Ansari **Practical Test** **2018450002**

BATCH 'A'

Q)

The Siruseri Singing Championship

(Zonal Computing Olympiad 2019)

The Siruseri Singing Championship is going to start, and Lavanya wants to figure out the outcome before the tournament even begins! Looking at past tournaments, she realizes that the judges care only about the pitches that the singers can sing in, and so she devises a method through which she can accurately predict the outcome of a match between any two singers.

She represents various pitches as integers and has assigned a lower limit and an upper limit for each singer, which corresponds to their vocal range. For any singer, the lower limit will always be less than the upper limit. If a singer has lower limit L and upper limit U ($L < U$), it means that this particular singer can sing in all the pitches between L and U, that is they can sing in the pitches {L, L+1, L+2, …, U}.

The lower bounds and upper bounds of all the singers are distinct. When two singers S_i and S_j with bounds (L_i, U_i) and (L_j, U_j) compete against each other, S_i wins if they can sing in every pitch that S_j can sing in, and some more pitches. Similarly, S_j wins if they can sing in every pitch that S_i can sing in, and some more pitches. If neither of these conditions are met, the match ends in a draw. ***In this problem, you can assume that no match ends in a draw***.

N singers are competing in the tournament. Each singer competes in N-1 matches, one match against each of the other singers. The winner of a match scores 2 points, and the loser gets no points. But in case of a draw, both the singers get 1 point each.

You are given the lower and upper bounds of all the N singers. You need to output the total scores of each of the N singers at the end of the tournament.

Solution hint

Since no match ends in a draw, for any pair of singers S_i and S_j, one of their vocal ranges is strictly included in the other. Deduce that, across all singers, the vocal ranges from a sequence where each interval is strictly included in the previous one. You can then sort the starting points of the vocal ranges and determine how many matches each singer wins from the position of their starting point in this sorted sequence.

Input format

The first line of of the input contains a single integer, N, which is the number of singers. N lines follow, the i-th of which contains two space-separated integers: L_i and U_i, which correspond to the lower bound and upper bound of the i-th singer.

Output format

Output a single line containing N space-separated integers, the i-th of which should be score of the i-th singer at the end of the tournament.

Test data

$2 \leq N \leq 10^5$.

$1 \leq L_i < U_i \leq 10^9$.

All the 2N integers (lower bounds and upper bounds) are distinct. No matches end in a draw.

Sample input 1

```
        5
1. 23
2. 20
```

```
1. 16
1. 19
1. 25
```

Sample output 1

```
6 4 0 2 8
```

Sample Test Cases	Input	Output

Test Case 1	5 23
	3
	4 20
	11 16
	5 19
	1 25
	7 22
		6 4 0 2 8

Test Case 2	3
	9 17
	6 19
	13 16
	2 25
	14 15
	5 21
		10 4 6 2 12 0 8

10

1. 20
1. 12
1. 21

1. 13

Test Case 3

6 16

2 22

9 14

7 15

5 19

1 25

12 0 14 2 8 16 4 6 10 18

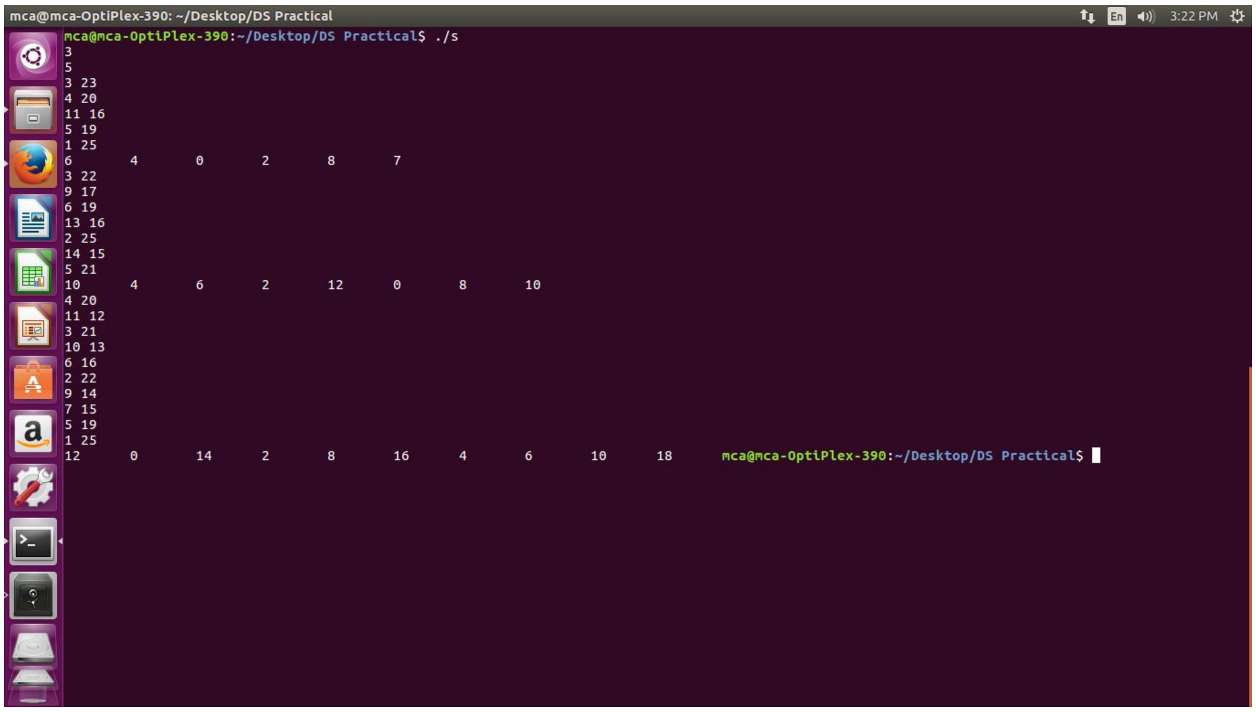
Code :

```
#include<stdio.h>
#define ML(t) while(t--)
int main()
{
        int n,t,i,j;
```

```
	scanf("%d",&t);					// Number of test because i wanted 
to submit it on codechef.
	ML(t)
	{
		scanf("%d",&n);
		int singer[n][3];			///////Singer[n][0] for lower cord, Singer[n][1] for upper 
cord, Singer [n][2] for score
		for(i=0; i<n; i++)
		{
			for(j=0; j<2; j++)
			{
				scanf("%d",&singer[i][j]);	//lower limit then upper limit
				singer[i][2]=0;
			}

		}

		for(i=0; i<n; i++)
	{
	   for(j=0; j<n; j++)
	   {
				if(i==j)			//so it'll not compare with same singer
				{continue;}
		    if(singer[j][0]<singer[j][1] && singer[i][0]<singer[i][1])
				{
					if(singer[j][0]==singer[i][0] && singer[j][1]==singer[i][1])
					{
						singer[i][2]++;				//if its equal 
increment score by 1
						singer[j][2]++;
					}else {if(singer[i][0]<singer[j][0])
						{
							if(singer[i][1]>=singer[j][1])
							{
								singer[i][2] = singer[i][2] + 2; // if 
upper limit is greater then increment score by 2
							}
						}
					}
				}
			}

	}
```

```
                for(i=0; i<n; i++)
                {
                        printf("%d\t",singer[i][2]);
                }
        }
}
```

Screenshot :



```
mca@mca-OptiPlex-390: ~/Desktop/DS Practical
mca@mca-OptiPlex-390:~/Desktop/DS Practical$ ./s
3
5
3 23
4 20
11 16
5 19
1 25
6       4       0       2       8       7
3 22
9 17
6 19
13 16
2 25
14 15
5 21
10      4       6       2       12      0       8       10
4 20
11 12
3 21
10 13
6 16
2 22
9 14
7 15
5 19
1 25
12      0       14      2       8       16      4       6       10      18      mca@mca-OptiPlex-390:~/Desktop/DS Practical$
```