You are given a two-dimensional 3*3 array starting from A [0][0]. You should add the alternate elements of the array and print its sum. It should print two different numbers the first being sum of A 0 0, A 0 2, A 1 1, A 2 0, A 2 2 and A 0 1, A 1 0, A 1 2, A 2 1.

Input Format

First and only line contains the value of array separated by single space.

A 0 0	A 0 1	A 0 2
4	6	9
A 1 0	A 1 1	A 1 2
2	5	8
A 2 0	A 2 1	A 2 2
1	3	7

Output Format

First line should print sum of A 0 0, A 0 2, A 1 1, A 2 0, A 2 2

Second line should print sum of A 0 1, A 1 0, A 1 2, A 2 1

SAMPLE INPUT

1 2 3 4 5 6 7 8 9

```c
#include<stdio.h>
int main()
{
    int arr[3][3];
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            scanf("%d",&arr[i][
        }
    }
    int odd=0,even=0;
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            if((i+j)%2!=0)
            {
                odd+=arr[i][j];
            else{
                even+=arr[i][j]
        }
    }
    printf("%d\n%d",even,odd);
    return 0;
}
```

	Input
✓	1 2 3 4 5 6 7 8 9
✓	21 422 423 443 586 645 657 846 904

Passed all tests! ✓

Microsoft has come to hire interns from your college. N students got shortlisted out of which few were males and a few females. All the students have been assigned talent levels. Smaller the talent level, lesser is your chance to be selected. Microsoft wants to create the result list where it wants the candidates sorted according to their talent levels, but there is a catch. This time Microsoft wants to hire female candidates first and then male candidates.

The task is to create a list where first all-female candidates are sorted in a descending order and then male candidates are sorted in a descending order.

Input Format

The first line contains an integer N denoting the number of students. Next, N lines contain two space-separated integers, ai and bi.
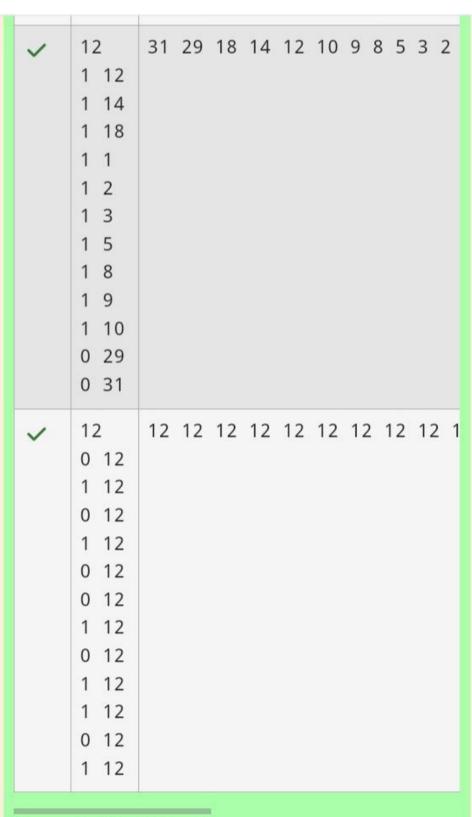
The first integer, ai will be either 1(for a male candidate) or 0(for female candidate).

The second integer, bi will be the candidate's talent level.
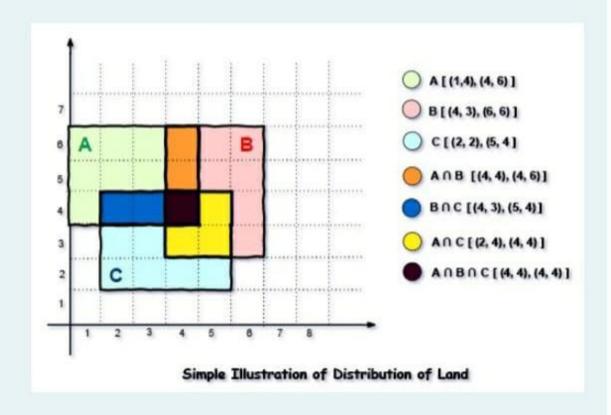
Constraints

```c
#include<stdio.h>
struct data
{
    int gen;int tal;
};
int main()
{
    int n;
    scanf("%d",&n);
    struct data a[n];
    for(int i=0;i<n;i++)
    scanf("%d %d",&a[i].gen,&a[:
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<n-i-1;++j
        {
            if(a[j].tal<a[j+1].
            {
                struct data temp
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    for(int i=0;i<n;i++)
    {
        if(a[i].gen==0)
        printf("%d ",a[i].tal);
    }
    for(int i=0;i<n;++i)
    {
        if(a[i].gen==1)
        printf("%d ",a[i].tal);
    }
}
```

	Input	Expected
✓	5 0 3 1 6 0 2 0 7 1 15	7 3 2 15 6
✓	6 0 1 0 26 0 39 0 37 0 7 0 13	39 37 26 13 7 1
✓	12 1 12 1 14 1 18 1 1 1 2 1 3 1 5 1 8 1 9 1 10 0 29 0 31	31 29 18 14 12 10 9 8 5 3 2
✓	12 0 12 1 12 0 12	12 12 12 12 12 12 12 12 12 1

✓	12	31 29 18 14 12 10 9 8 5 3 2
	1 12	
	1 14	
	1 18	
	1 1	
	1 2	
	1 3	
	1 5	
	1 8	
	1 9	
	1 10	
	0 29	
	0 31	
✓	12	12 12 12 12 12 12 12 12 12 1
	0 12	
	1 12	
	0 12	
	1 12	
	0 12	
	0 12	
	1 12	
	0 12	
	1 12	
	1 12	
	0 12	
	1 12	

Passed all tests! ✓

Explanation

Simple Illustration of Distribution of Land

For given sample input (see given graph for reference), compensation money for different farmers is as follows:

Farmer with land area A: $C_1 = 5 * 1 = 5$

Farmer with land area B: $C_2 = 6 * 2 = 12$

Farmer with land area C: $C_3 = 6 * 3 = 18$

Total Compensation Money = $C_1 + C_2 + C_3 = 5 +$

```c
#include<stdio.h>
int main()
{
    int i,j,n,x1,x2,y1,y2,t=0;
    long long total=0;
    int arr[1001][1001]={0};
    scanf("%d",&n);
    while(n--)
    {
        scanf("%d %d %d %d %d",&
        for(i=x1;i<=x2;i++)
        {
            for(j=y1;j<=y2;j++)
            {
                if(arr[i][j]==0
                arr[i][j]+=t;
                else if(arr[i][
                arr[i][j]=(-1)*
                else if(arr[i][
                arr[i][j]-=t;
            }
        }
    }
    for(i=1;i<1001;i++)
    {
        for(j=1;j<1001;j++)
        {
            if(arr[i][j]<0)
            total+=arr[i][j];
        }
}

        printf("%lld\n",(-1)*to
        return 0;


}
```

Input	Expected	Got	
3 1 4 4 6 1 4 3 6 6 2 2 2 5 4 3	35	35	✓
1 48 12 49 27 8	0	0	✓
3 88 34 99 76 44 82 65 94 100 81 58 16 65 66 7	10500	10500	✓

Passed all tests! ✓