Week-09-01

Name: Mukesh V **Roll No: 241901059**

Add Alternate Elements of 2-Dimensional Array

Problem Statement:

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.

A00	A01	A02
4	6	9
A10	**A11**	**A12**
2	5	8
A20	**A21**	**A22**
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

Sample Output
25
20

Program:

```c
#include <stdio.h>

int main(){
    int m[3][3];
    int set1_sum = 0, set2_sum = 0;

    for (int i = 0;i < 3;i++){
        for(int j = 0;j < 3;j++){
            scanf("%d",&m[i][j]);
        }
    }

    set1_sum = m[0][0] + m[0][2] + m[1][1] + m[2][0] + m[2][2];
    set2_sum = m[0][1] + m[1][0] + m[1][2] + m[2][1];

    printf("%d\n",set1_sum);
    printf("%d\n",set2_sum);
    return 0;

}
```

	Input	Expected	Got	
✓	1 2 3 4 5 6 7 8 9	25 20	25 20	✓
✓	21 422 423 443 586 645 657 846 904	2591 2356	2591 2356	✓

Passed all tests! ✓

The Wealthy Landlord

Problem Statement:

Shyam Lal, a wealthy landlord from the state of Rajasthan, being an old fellow and tired of doing hard work, decided to sell all his farmland and to live rest of his life with that money. No other farmer is rich enough to buy all his land so he decided to partition the land into rectangular plots of different sizes with different cost per unit area. So, he sold these plots to the farmers but made a mistake. Being illiterate, he made partitions that could be overlapping. When the farmers came to know about it, they ran to him for compensation of extra money they paid to him. So, he decided to return all the money to the farmers of that land which was overlapping with other farmer's land to settle down the conflict. All the portion of conflicted land will be taken back by the landlord.

To decide the total compensation, he has to calculate the total amount of money to return back to farmers with the same cost they had purchased from him. Suppose, Shyam Lal has a total land area of 1000 x 1000 equal square blocks where each block is equivalent to a unit square area which can be represented on the co-ordinate axis. Now find the total amount of money, he has to return to the farmers. Help Shyam Lal to accomplish this task.

Input Format:The first line of the input contains an integer N, denoting the total and pieces he had distributed. Next N line contains the 5 space separated integers (X1, Y1), (X2, Y2) to represent a rectangular piece of land, and cost per unit area C.

(X1, Y1) and (X2, Y2) are the locations of first and last square block on the diagonal of the rectangular region.

Output Format:

Print the total amount he has to return to farmers to solve the conflict.

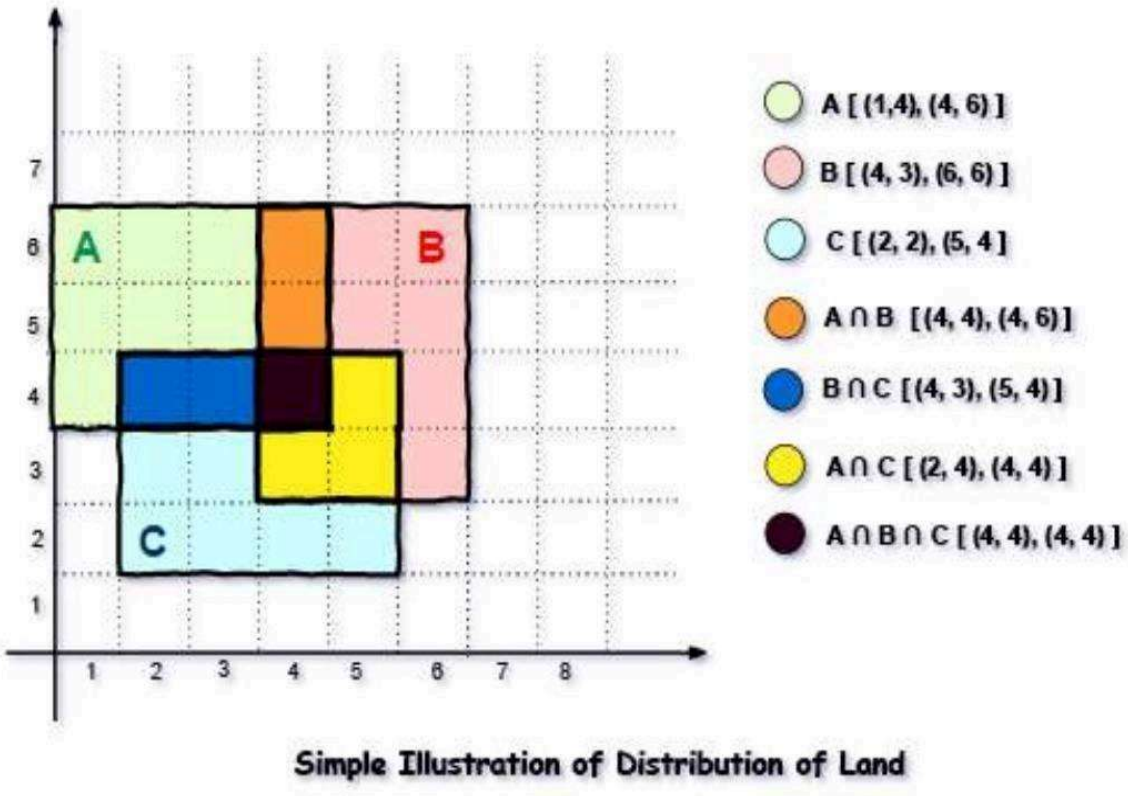
Constraints:

$1 \le N \le 100$
$1 \le X1 \le X2 \le 1000$
$1 \le Y1 \le Y2 \le 1000$
$1 \le C \le 1000$

Sample Input
3
1 4 4 6 1
4 3 6 6 2
2 2 5 4 3

Sample Output
35

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: C1 = 5 * 1 = 5
Farmer with land area B: C2 = 6 * 2 = 12
Farmer with land area C: C3 = 6 * 3 = 18

Total Compensation Money = C1 + C2 + C3 = 5 + 12 + 18 = 35

Program:

```c
#include <stdio.h>
#include <stdlib.h>

int compare(const void *a,const void *b){
    return (*(int*)b - *(int*)a);
}

int main(){
    int N;
    scanf("%d",&N);
    int fT[N], mT[N];
    int fC = 0, mC = 0;

    for (int i = 0;i< N;i++){
        int ai,bi;
        scanf("%d %d",&ai,&bi);

        if(ai == 0){
            fT[fC++] = bi;
        }
        else
        mT[mC++] = bi;
    }
    qsort(fT,fC,sizeof(int),compare);
    qsort(mT,mC,sizeof(int),compare);

    for(int i = 0; i < fC; i++){
        printf("%d ",fT[i]);
    }
    for(int i = 0;i < mC; i++){
        printf("%d ",mT[i]);
    }
    return 0;
}
```

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

Passed all tests! ✓

Priority Interview

Problem Statement:

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: 1 <= N <= 105, 0 <= ai <= 1, 1 <= bi <= 109

Output Format
Output space-separated integers, which first contains the talent levels of all female candidates sorted in descending order and then the talent levels of male candidates in descending order.

Sample Input
5
0 3
1 6
0 2
0 7
1 15
Sample Output
7 3 2 15 6

Program:

```c
#include <stdio.h>

#define MAX_SIZE 1001

int main(){
    int i,j,n,x1,y1,x2,y2,t= 0;
    long long total = 0;
    int arr[MAX_SIZE][MAX_SIZE] = {0};
    scanf("%d",&n);
    while(n -- ){
        scanf("%d %d %d %d %d",&x1,&y1,&x2,&y2,&t);

        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][j] > 0)
                arr[i][j] = (-1) * (arr[i][j] + t);
                else if(arr[i][j] < 0)
                arr[i][j] -= t;
            }
        }
    }
    for(i = 1; i < MAX_SIZE;i++){
        for(j = 1;j < MAX_SIZE;j++){
            if(arr[i][j] < 0){
                total += arr[i][j];
            }
        }
    }
    printf("%lld\n",(-1)*total);
    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! ✓