GE23131-Programming Using C-2024

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Monday, 23 December 2024, 2:20 PM
Duration	3 hours 12 mins

Question **1**
Correct
Marked out of 1.00
Flag question

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

SAMPLE OUTPUT

25
20

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int arr[3][3];
    int sum1 = 0, sum2 = 0;

    // Input the 2D array
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            scanf("%d", &arr[i][j]);
        }
    }

    // Calculate the sums of alternate elements
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if ((i + j) % 2 == 0) {
                sum1 += arr[i][j];
            } else {
                sum2 += arr[i][j];
            }
        }
    }

    // Output the results
    printf("%d\n", sum1);
    printf("%d\n", sum2);

    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! ✓

Question **2**
Correct
Marked out of 5.00
Flag question

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 <= 10^5^

0 <= ai <= 1

1 <= bi <= 10^9^

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

Answer: (penalty regime: 0 %)

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

struct Candidate {
    int gender; // 0 for female, 1 for male
    int talent;
};

int compare(const void *a, const void *b) {
    struct Candidate *c1 = (struct Candidate *)a;
    struct Candidate *c2 = (struct Candidate *)b;

    // Sort females first, then males
    if (c1->gender != c2->gender) {
        return c1->gender - c2->gender;
    }

    // Sort by talent in descending order
    return c2->talent - c1->talent;
}

int main() {
    int N;
    scanf("%d", &N);

    struct Candidate candidates[N];
    for (int i = 0; i < N; i++) {
        scanf("%d %d", &candidates[i].gender, &candidates[i].talent);
    }

    qsort(candidates, N, sizeof(struct Candidate), compare);

    for (int i = 0; i < N; i++) {
        printf("%d ", candidates[i].talent);
    }
    printf("\n");

    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! ✓

Question **3**
Correct
Marked out of 1.00
Flag question

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 number of land 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 ≤ N ≤ 100

1 ≤ X1 ≤ X2 ≤ 1000

1 ≤ Y1 ≤ Y2 ≤ 1000

1 ≤ C ≤ 1000

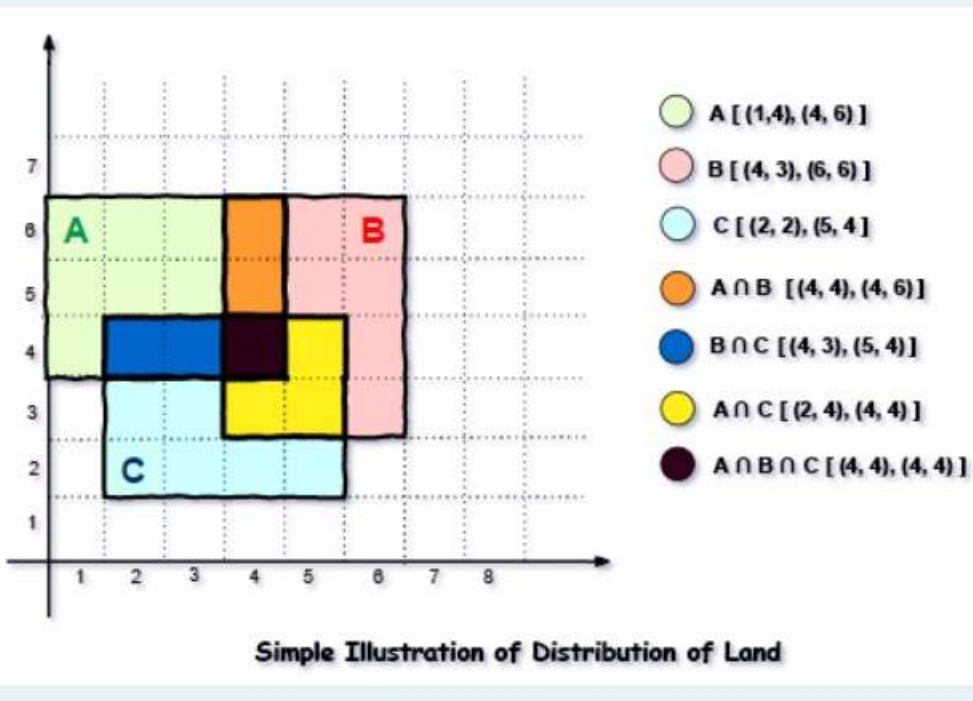
SAMPLE INPUT

3
1 4 4 6 1
4 3 6 6 2
2 2 5 4 3

SAMPLE OUTPUT

35

Explanation



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 = $C_1 + C_2 + C_3 = 5 + 12 + 18 = 35$

Answer: (penalty regime: 0 %)

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

// Grid size
#define MAX 1000

int main() {
    int N; // Number of land pieces
    scanf("%d", &N);

    // 2D arrays for marking the total cost and overlap count on each cell
    int cost[MAX + 1][MAX + 1] = {0};
    int overlap[MAX + 1][MAX + 1] = {0};

    // Reading input and marking the grid
    for (int i = 0; i < N; i++) {
        int x1, y1, x2, y2, c;
        scanf("%d %d %d %d %d", &x1, &y1, &x2, &y2, &c);

        // Normalize coordinates (ensure x1 <= x2 and y1 <= y2)
        int minX = x1 < x2 ? x1 : x2;
        int maxX = x1 > x2 ? x1 : x2;
        int minY = y1 < y2 ? y1 : y2;
        int maxY = y1 > y2 ? y1 : y2;

        // Mark the grid for this rectangle
        for (int x = minX; x <= maxX; x++) {
            for (int y = minY; y <= maxY; y++) {
                cost[x][y] += c;       // Add the cost of this rectangle
                overlap[x][y] += 1;   // Increment overlap count
            }
        }
    }

    // Calculate the total compensation for overlapping regions
    long long totalCompensation = 0;
    for (int x = 1; x <= MAX; x++) {
        for (int y = 1; y <= MAX; y++) {
            if (overlap[x][y] > 1) { // If there's an overlap
                totalCompensation += cost[x][y];
            }
        }
    }

    // Output the total compensation
    printf("%lld\n", totalCompensation);
    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! ✓