Programming Using C

week **09** practice session coding

Name:K.Kamaleshwaran
Department:AIML-'B'
Roll No.:242501079

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] and A[2][2] and the second sum 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
A 1 0	A 1 1	A 1 2
A 2 0	A 2 1	A 2 2

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];
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            scanf("%d",&arr[i][j]);
        }
    }
    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=odd+arr[i][j];
            else
                even=even+arr[i][j];
        }
    }
    printf("%d\n%d",odd,even);
}
```

	Input	Expected	Got	
✓	1 2 3 4 5 6 7 8 9	25 20	25 20	✓
✓	...	...	...	✓

Passed all tests! ✓

The first line contains an integer N denoting the number of students. Next, N lines contain two space-separated integers, ci and bi.

The first integer, ci will be either '0'(is a male candidate) or 0(is female candidate)

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

Constraints:

$1 <= N <= 10^5$

$0 <= ci <= 1$

$1 <= bi <= 10^9$

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

4
0 5
1 6
0 2
0 7
1 9

SAMPLE OUTPUT

7 5 2 9 6

Answer: (penalty regime 0 %)

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

	Input	Expected	Got	
✓	4	7 5 2 9 6	7 5 2 9 6	✓
✓	8	90 50 40 30 20 9 6	90 50 40 30 20 9 6	✓
✓	11	90 80 50 30 20 9 8 7 6 5	90 80 50 30 20 9 8 7 6 5	✓
✓	17	...	...	✓

Print the total amount the loot be taken to Saveeta in value then 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 3 6 7
4 7 5 8 7
2 7 4 8 3

SAMPLE OUTPUT

91

Explanation:

Graph Illustration of Distribution of Loot

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

Farmer with Loot area A: $C_a = 6 * 3 = 6$

Farmer with Loot area B: $C_b = 6 * 2 = 12$

Farmer with Loot area C: $C_c = 6 * 3 = 18$

Total Compensation Money = $C_a + C_b + C_c = 6 + 12 + 18 = 91$

Answer: (penalty regime 0 %)

```c
#include<stdio.h>
int main()
{
    int i,j,n,x1,x2,y1,y2,n,k;
    long long total=0;
    int arr[1001][1001]={0};
    scanf("%d",&n);
    while(n--)
    {
        scanf("%d %d %d %d %d",&x1,&y1,&x2,&y2,&k);
        for(i=x1;i<x2;i++)
        {
            for(j=y1;j<y2;j++)
            {
                if(arr[i][j]==0)
                    arr[i][j]=k;
                else if(arr[i][j] < k) arr[i][j]=k;
                else if(arr[i][j])
                    arr[i][j]= k;
            }
        }
    }
    for(i=1;i<1001;i++)
    {
        for(j=1;j<1001;j++)
        {
            if(arr[i][j]>0)
                total+=arr[i][j];
        }
    }
    printf("%lld\n",( 2)*total);
    return 0;
}
```

	Input	Expected	Got	
✓	3 1 4 3 6 7 4 7 5 8 7 2 7 4 8 3	91	91	✓
✓	1 60 11 45 29 6	0	0	✓
✓	2 60 19 95 74 24 45 45 100 59 7	14456	14456	✓

Passed all tests ✓