```
Answer: (penalty regime: 0 %)
      #include<stdio.h>
       int main()
    2
    3 ▼
       {
    4
           int arr[3][3];
           int sum1 = 0, sum2 = 0;
    5
           for(int i=0; i<3; i++)
    6
    7 🔻
            {
                for(int j=0; j<3; j++
    8
    9 •
   10
                    scanf("%d",&arr[i
   11
   12
           for(int i=0; i<3; i++)
  13
  14 ▼
                for(int j=0; j<3; j++
  15
   16 •
                {
                    if((i+j) \% 2 == 0
  17
  18 •
                    {
  19
                         sum1 += arr[i
  20
  21
                    else
  22 *
                    {
                         sum2 += arr[i
  23
   24
                    }
  25
  26
           printf("%d",sum1);
  27
           printf("\n%d", sum2);
  28
           return 0;
  29
  30
      }
```

	Input		
<b>~</b>	1 2 3 4 5 6 7 8 9		



```
ude<stdio.h>
 1
   ain()
 2
 3 ▼
   nt arr[3][3];
   nt sum1 = 0, sum2 = 0;
   or(int i=0; i<3; i++)
 7 🔻
       for(int j=0; j<3; j++)
8
9 🔻
           scanf("%d",&arr[i][j]);
10
11
       }
12
   or(int i=0; i<3; i++)
13
14 ▼
15
       for(int j=0; j<3; j++)
16 ▼
       {
17
           if((i+j) \% 2 == 0)
18 •
                sum1 += arr[i][j];
19
20
           }
           else
21
22 *
           {
                sum2 += arr[i][j];
23
24
           }
25
       }
26
   rintf("%d", sum1);
27
   rintf("\n%d",sum2);
28
29
   eturn 0;
30
```

Input

1 2 3 4 5 6 7 8 9

```
or(int i=0; i<3; i++)
13
14 ▼
       for(int j=0; j<3; j++)</pre>
15
16 ▼
           if((i+j) \% 2 == 0)
17
18 •
                sum1 += arr[i][j];
19
20
21
           else
22 🔻
           {
                sum2 += arr[i][j];
23
24
           }
25
       }
26
   rintf("%d",sum1);
27
   rintf("\n%d",sum2);
28
    eturn 0;
29
30
```

	Input
~	1 2 3 4 5 6 7 8 9
~	21 422 423 443 586 645 657 846
Passe	d all tests! 🗸

```
Question \bf 2
```

Correct

Marked out of 5.00

Flag question

## **Answer:** (penalty regime: 0 %)

```
#include<stdio.h>
 2
    #include<stdlib.h>
 3
    typedef struct
 4 ▼
    {
 5
        int gender;
 6
        int talent;
 7
 8
    Student;
 9
    int compare_students(const vo
10 ▼
11
        const Student *student_a
12
        const Student *student_b
13
        if(student_a -> gender <</pre>
14 ▼
15
             return -1;
16
        else if(student_a -> gend
17
18 •
19
             return 1;
20
         }
        else
21
22 ▼
23
             return student_b -> t
24
        }
25
26
    int main()
27 ▼
    {
28
        int N;
29
        scanf("%d",&N);
30
        Student * students = (Stu
31
        for(int i=0; i<N; i++)
32 ▼
        {
33
             scanf("%d %d",&studen
34
35
        qsort(students,N,sizeof(S
36
        for(int i=0; i<N; i++)
37 ▼
         {
             printf("%d ",students
38
39
        printf("\n");
40
```

```
Answer: (penalty regime: 0 %)
    1
    2
    3
   4 ▼
   5
    6
    7
   8
       void *a,const void *b)
   9
   10 ▼
      _a = (const Student *)a;
   11
       _b = (const Student *)b;
  12
       < student_b -> gender)
  13
  14 ▼
  15
  16
      gender > student_b -> gender)
  17
  18 ▼
  19
  20
  21
  22 🔻
      >> talent - student_a -> talen
  23
  24
  25
  26
  27 ▼
  28
  29
      Student *)malloc (N * sizeof(
  30
  31
  32 ▼
      idents[i].gender,&students[i].
  33
  34
  35
      f(Student),compare_students);
  36
```



ents[i].talent);

37 ▼

```
24
        qsort(students,N,sizeof(S
35
        for(int i=0; i<N; i++)</pre>
36
37 ▼
        {
            printf("%d ",students
38
39
        printf("\n");
40
41
        free(students);
        return 0;
42
43
```

-		
	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	31 29 18 14 12 10 9 8 5 3



```
Answer: (penalty regime: 0 %)
      #include<stdio.h>
   1
      #include<stdlib.h>
   2
   3 ▼ typedef struct {int x1,y1,x2,
   4
   5
      land;
   6
      int overlap(land a,land b)
   7 ▼
   8
           return a.x1 <= b.x2 && b.
   9
  10
      int overlap_area(land a,land
  11 ▼ {
  12
           int x_overlap = (a.x2,b.x)
           int y_overlap = min(a.y2,
  13
           return x_overlap * y_over
  14
  15
  16
      int main()
  17 ▼ {
  18
           int n;
           scanf("%d",&n);
  19
  20
           land lands[n];
  21
           for(int i=0; i<n; i++)
  22 *
           {
               scanf("%d %d %d %d %d
  23
  24
  25
           int total_compensation =
           for(int i=0; i<n; i++)
  26
  27 🔻
           {
  28
               for(int j=i+1; j<n; j
  29 •
               {
                    if(overlap(lands[
  30
  31 ▼
                    {
  32
                        int area_over
  33
                        int cost_per_
  34
                        total_compens
  35
  36
```



```
Answer: (penalty regime: 0 %)
    1
    2
    3 ▼
   4
    5
    6
    7 ▼
    8
      & a.y1 <= b.y2 && b.y1 <= a.y2
    9
  10
   11 ▼
   12
      1,b.x1);
      a.y1,b.y1);
   13
   14
   15
  16
  17 ▼
   18
   19
  20
  21
  22 *
      x1,&lands[i].y1,&lands[i].x2,{
  23
  24
  25
  26
  27 ▼
  28
  29 •
  30
  31 ▼
      p_area(lands[i],lands[j]);
  32
      s[i].cost + lands[j].cost) / 2
  33
      a_overlap * cost_per_unit;
  34
  35
```