

Each rectangle given to you is uniquely identified by the two end-points of one of its diagonals. That is, you are given the lower-left and the top-right points of one of the diagonal of the rectangle.

The rectangles are indexed from 1 to N. In each of the queries below, we will refer to a rectangle by its index.

Input Format

The queries can be of the following type:

- Given two rectangles R1 and R2 output 1 if they interesect, 0 otherwise. The input to this query will be "I index-of-R1 index-of-R2". The output is either 0 or 1 followed by a newline.
- Given two rectangles R1 and R2 output the intersction co-ordinates. The input to this query will be "C index-of-R1 index-or-R2". The output is 2 points (each with their x and y co-ordinates) representing the interesecting rectangle followed by a newline. In case the rectangles R1 and R2 do not interesect, output -1 -1 -1 followed by a newline.
- Given a rectangle R compute its integral area. The input to this query is "A index-of-R". The output is the integral are followed by a newline.
- Sort the given rectangles in non-increasing order of their integral area and print the integral area in sorted order. The input to this query is "S". The output is N sorted values one on each line.
- Given two rectangles R1 and R2 determine the number of paths of length at most 3 between R1 and R2 in the intersection graph (see pdf for the definition of the graph and the paths between two vertices). The input to this query is "P index-of-R1 index-of-R2". The output is the number of paths of length at most 3 followed by a newline.

```
N x11 y11 x12 y12 x21 y21 x22 y22 x31 y31 x32 y32 ... xN1 yN1 xN2 yN2 M I 1 2 I 3 4 A 1 C 2 3 P 3 4 S
```

Constraints

Assume that N <= 500. Assume that all the rectangles have all co-ordinates which are positive.

Output Format

Output format is described with each query above.

Sample Input 0

Sample Output 0

Explanation 0

The input consists of 5 rectangles. The co-ordinates for these are listed on lines 2--6. Each line consists of 4 non-negative points which specify the lower-left and upper right points of the rectangle. For example the first rectangle is given by ((2,2)(8,4)).

Line 7 contains the number of queries. There are 8 queries which are listed on lines 8--15.

- The first 3 queries are interesection queries. The first query asks you to output if R1 and R2 intersect. The output of the first 3 queries is given (one per line) on the first 3 lines of the output file.
- The queries 4,5 and 6 are Co-ordinates queries and their output is listed (one per line) on lines 4, 5, 6 of the output file. Note that since R3 and R4 do not intersect, the output on line 5 is (-1 -1 -1 -1).
- The query 7 is an Area query and asks you to compute integral area of R2. Since there are 15 integral points in the rectangle R2, the output 15 is printed on line 7 of the output file.
- Finally query 8 is a sort query and the sorted values of the integral areas of the 5 rectangles are printed one on each line in the output file.

Sample Input 1

Sample Output 1

7

Explanation 1

The input consists of 2 rectangles whose co-ordinates are listed on lines 2--3 in the input file. There is 1 query which is a path query listed on line 5.

The graph contains two vertices (v1 and v2) and edges ((v1, v1) (v1, v2) (v2, v2)). The number of paths of length 1 between v1 and v2 is one. The number of paths of length 2 between v1 and v2 is two, namely (v1, v2, v2) and (v1, v1, v2). The number of paths of length 3 between v1 and v2 is four, namely (v1, v1, v1, v2) (v1, v1, v2, v2) (v1, v1, v2, v2) (v1, v2, v2 v2). Therefore the total number of paths of length at most 3 between v1 and v2 are 7.

Sample Input 2

```
4
0 0 4 4
1 1 2 2
0 2 3 5
5 6 5 6
3
I 2 1
```

I 3 2 I 3 4

Sample Output 2

1 1 0

Sample Input 3

Sample Output 3

1 1 0

Sample Input 4

Sample Output 4

1 1 2 2 1 2 2 2 -1 -1 -1 -1

Sample Input 5

Sample Output 5

2 2 4 4 4 4 4 4 -1 -1 -1 -1

Sample Input 6

A 2 A 3

Sample Output 6

25 16 1

Sample Input 7

> 36 50 55 50 24 25 54 50

8 9 49 14

> 2 27 35 46 19 51 40 88

- 2 20 18 58 9 42 41 78 37 45 89 48 43 1 81 40 35 14 77 54 45 50 69 98 36 46 60 48 46 19 90 46 7 13 46 62 51 32 85 67 12 38 36 40 10 41 49 65 8 43 10 43 6 27 46 69 44 24 65 40 12 43 55 95 8 48 11 54 16 41 43 69 52 30 78 71 45 12 96 44 32 32 67 70 1 7 40 52 7 40 17 91 32 50 63 69 13 26 58 56 52 37 98 88 52 18 67 41 45 2 54 26 44 42 57 88 52 17 101 55 38 37 43 54 36 36 70 64 6 5 47 10 35 35 78 48 37 36 37 65 9 32 26 65 30 26 51 35 49 47 71 81 27 30 58 72 45 18 59 69 34 35 57 83 45 8 62 16 17 16 40 21 40 38 89 88 17 38 60 52 13 27 45 67 43 30 48 44 51 12 99 44 50 28 97 71 43 30 58 37 39 44 86 49 43 34 73 73 20 0 39 12 31 0 38 21 12 14 21 24 36 42 62 75 52 2 101 44 12 28 45 58 27 14 46 63 28 35 56 84 47 40 62 44 22 25 62 59 28 23 67 34 44 45 56 67 41 11 67 11 6 9 24 21 38 8 40 21 30 41 53 45 17 8 46 47 48 52 61 76 31 14 53 37 13 24 19 25 12 33 27 51 33 10 54 28 24 11 74 44 20 37 72 74 7 2 53 37 6 6 39 43 34 39 54 66 6 7 17 25 26 15 67 21 41 4 66 48 2 25 17 47 46 39 87 72 34 15 54 34 33 4 33 18 52 41 74 46
- https://www.hackerrank.com/contests/aplmidsem2017/challenges/letstangle

Sample Output 7

Sample Input 8

18 7 61 24 30 50 47 61 11 20 54 24

- 10 24 33 59 22 12 72 41 43 19 88 51 2 37 5 56 6 14 19 17 35 20 56 64 17 3 34 31 39 32 41 82 25 11 60 59 33 3 35 5 47 35 48 85 36 48 87 69 44 24 53 29 9 50 53 77 41 34 73 61 34 3 48 9 49 33 63 41 16 43 32 54 17 40 21 41 2 22 37 68 24 31 71 64

- 17 39 45 76
- 39 44 60 96 14 37 61 79
- 41 51 91 87
- 13 26 54 56
- 45 31 58 57
- 28 52 61 83
- 27 11 57 55
- 37 8 88 10
- 51 1 103 13
- 46 1 63 35 14 50 66 56
- 19 37 42 80
- 37 17 84 61 27 22 43 55
- 25 4 37 13
- 42 11 54 51
- 11 52 55 88
- 8 50 9 52
- 0 8 47 60
- 17 42 41 75
- 51 24 89 28
- 33 17 79 22 6 14 27 62
- 12 15 16 38
- 27 38 41 53
- 15 17 50 65
- 29 47 33 73
- 8 6 23 44
- 1 42 32 76
- 3 19 41 28
- 6 37 39 55
- 17 20 69 64
- 45 6 51 14 20 35 22 64
- 39 2 68 49
- 23 11 31 35
- 39 38 39 60
- 39 10 43 34
- 16 22 63 55
- 25 4 46 54 6 5 48 32
- 45 3 85 34
- 48 4 85 22
- 45 22 60 33
- 47 33 54 45
- 43 16 86 22 17 19 55 40
- 38 18 40 63
- 44 51 67 66
- 42 46 75 83 17 35 46 44
- 44 52 48 70
- 43 30 75 63 23 18 48 58
- 10 40 47 68
- 39 20 45 30 8 25 58 76
- 30 14 49 61
- 27 35 55 86 32 16 67 17
- 14 34 39 50
- 51 35 83 75 16 15 39 50
- 8 45 43 61
- 11 14 28 55

> 28 26 60 69 3 40 8 91

40 7 43 19

- 16 16 44 36 45 18 91 54 36 9 77 61
- 3 17 9 24
- 47 12 79 42
- 28 49 42 63
- 28 11 68 50
- 41 42 58 78
- 45 35 82 72 30 52 62 97
- 21 24 29 48
- 47 10 88 51
- 20 50 42 77
- 36 42 61 86
- 45 30 77 42
- 28 48 80 68
- 36 37 66 83
- 10 17 46 48
- 24 34 65 52
- 29 38 52 66
- 46 3 60 11
- 19 52 43 63
- 35 23 44 33 8 9 58 53
- 27 16 73 32
- 9 26 43 59
- 22 30 61 82
- 10 6 57 9
- 8 11 49 38 12 17 55 64
- 52 36 92 75
- 37 30 83 74
- 39 7 65 34
- 7 39 20 68
- 52 7 68 16
- 10 44 24 95
- 2 25 5 72
- 25 20 68 44
- 9 10 13 36
- 50 17 69 32
- 24 22 27 53
- 16 8 24 24 25 4 72 18
- 40 12 89 34
- 31 16 48 19
- 39 6 54 54 51 21 99 48
- 46 22 63 40
- 21 49 65 86
- 52 32 88 35
- 10 18 47 68
- 12 19 21 41
- 26 25 40 37
- 9 40 19 47
- 5 35 45 65
- 5 48 9 74
- 48 42 72 89
- 39 51 61 79 38 26 86 55
- 4 51 28 61
- 18 1 20 28
- 12 34 32 83 8 26 24 39
- 52 39 53 65
- 4 21 11 43
- 29 51 80 97
- 25 1 28 9
- 6 18 58 42 1 51 53 96
- 19 41 52 47
- 28 19 75 26 48 33 85 64
- 45 32 78 32
- 10 26 40 61 12 44 39 63
- 27 43 68 50
- 19 0 40 18
- 33 24 53 64
- 14 47 58 88 7 19 35 71
- 40 32 92 83
- 8 12 13 12
- 11 19 14 57 24 25 65 47
- 41 40 45 61
- 24 9 36 48 35 27 70 48

Sample Output 8

52

48

45 44

37 36 36

35 35 34

34 33

31 30 30

27 27 25

23 23

20 18 18

11 10

Sample Input 9

500 3 45 3 48

46 11 79 63 14 22 54 66

36 32 76 67

49 1 95 6 16 16 50 39

17 10 34 58

34 52 42 90

41 20 85 54

4 34 36 52

19 9 54 43

22 48 42 66

45 2 94 11

30 32 48 58

36 0 57 17 30 34 61 52

2 31 4 37

13 3 25 14 47 46 60 94

12 13 53 18

17 14 65 40

45 0 70 7 26 25 58 28

11 0 49 45

40 51 50 83

23 44 24 93

46 45 83 83 4 43 41 64

11 48 47 52 40 11 67 24

38 48 53 76

1 20 28 40 38 0 88 40

52 36 75 60

7 9 14 20 45 12 76 47

46 39 85 73

46 39 85 73 13 26 64 56

13 6 35 52

28 46 54 91 24 32 70 34

16 5 64 7

2 45 49 71 25 9 29 28

43 32 70 67

4 13 9 63

30 30 39 30

30 29 58 58

- 1 34 16 54
- 9 46 53 80
- 46 37 79 78
- 12 33 55 67
- 6 47 27 65
- 3 22 4 62
- 35 45 63 57
- 35 21 35 57
- 15 3 49 6
- 25 38 54 57
- 9 39 10 60
- 44 34 63 63
- 19 26 47 48
- 29 9 77 52
- 2 35 35 73
- 12 21 15 28
- 16 10 20 52
- 33 40 60 61
- 7 22 33 52
- 24 38 27 81
- 11 12 22 52
- 7 10 7 19
- 12 26 57 50
- 27 31 75 74
- 52 11 73 22 27 32 78 46
- 3 44 4 71
- 5 49 14 97 20 35 60 41
- 40 48 64 79
- 16 3 37 25
- 48 45 61 92
- 13 37 16 57
- 34 13 51 50
- 50 44 54 78
- 45 29 85 42
- 28 51 72 98
- 15 26 61 37
- 47 12 55 54
- 21 16 25 30
- 8 13 8 55
- 49 5 75 52
- 9 7 58 9 16 47 53 70
- 7 17 52 39
- 18 12 40 24
- 30 1 55 32
- 29 45 46 61 50 37 87 83
- 10 19 53 38
- 18 0 24 13
- 43 36 90 86
- 19 51 52 88
- 2 49 13 82
- 15 11 65 34
- 47 40 82 84
- 19 16 43 24
- 46 6 61 50
- 21 4 28 15
- 33 40 74 92 21 36 38 38
- 7 35 39 57
- 9 46 55 49
- 7 26 19 52 15 13 36 22
- 36 0 56 4
- 6 47 42 86
- 24 18 58 42 52 27 53 33
- 13 8 54 9
- 14 36 47 57
- 24 27 33 66 30 27 70 73
- 39 50 45 95
- 43 11 88 57 26 50 55 54
- 9 43 12 65
- 44 2 95 7
- 10 6 48 40 26 20 38 24
- 31 50 79 68
- 33 42 59 65 18 17 50 41
- 8 7 54 25 49 19 99 59
- 27 25 49 62
- 39 18 49 31

41 23 63 75

41 35 57 44

```
19 15 58 44
24 6 26 26
4 12 32 23
42 48 44 86
29 33 40 63
38 51 62 69
34 27 49 28
31 11 41 41
49 6 54 58
7 20 51 31
19 1 51 42
13 26 62 68
39 52 77 76
9 21 59 43
44 24 93 26
38 32 52 45
19 13 36 39
8 16 20 22
44 47 61 51
```

Sample Output 9

Sample Input 10

41 42 86 76

> 11 46 11 97 40 26 64 49

- 39 20 62 61 44 17 93 52 47 50 82 66 0 39 30 56
- 4 0 56 48 37 7 52 23
- 2 0 37 18
- 21 29 49 68
- 51 27 100 69
- 24 3 48 22
- 6 35 6 41
- 9 23 30 69
- 52 20 76 56
- 39 32 45 52
- 41 38 73 79
- 39 27 54 43
- 11 37 45 52
- 5 34 24 45
- 19 49 35 56
- 38 0 57 16
- 22 32 22 72
- 6 8 17 56
- 22 15 47 55
- 40 35 83 66
- 3 46 22 54
- 38 19 65 55
- 22 43 69 82
- 13 3 36 18 23 2 37 31
- 25 3 67 29
- 15 13 66 47
- 20 12 68 35
- 46 31 83 41
- 43 46 72 59
- 23 52 38 67
- 17 30 51 49
- 48 28 60 80
- 1 25 11 41
- 8 51 25 79
- 44 51 86 67
- 27 26 35 43
- 37 9 88 48
- 46 2 87 12
- 1 9 12 59
- 23 28 60 53 47 41 80 76
- 29 10 47 31
- 8 37 49 51
- 28 10 71 55
- 41 31 61 65
- 40 24 52 65 24 17 57 43
- 17 51 63 94
- 26 25 45 27
- 16 2 31 26
- 35 38 54 48
- 9 2 37 31
- 48 11 61 46 25 2 60 30
- 47 2 67 45
- 45 33 77 83
- 30 31 68 78
- 0 50 34 85
- 0 24 15 65
- 43 17 70 55 40 52 47 96
- 21 20 54 35
- 0 5 1 29
- 33 21 51 64 3 37 3 72
- 34 17 68 30
- 49 33 90 51 47 3 97 37
- 5 5 39 10
- 47 21 72 68
- 5 19 10 36 23 7 42 12
- 19 40 63 72
- 5 45 9 78 25 52 29 71
- 2 33 36 72
- 48 44 86 86 42 15 86 62
- 50 11 63 63
- 33 5 84 57 21 32 66 58
- 50 39 53 40

Sample Output 10

S

Sample Input 11

22 39 27 65 6 38 7 65

48 2 64 11 11 52 11 73 15 32 65 44

38 38 66 58 4 21 41 66 47 9 81 57

39 0 87 25 29 52 68 76

34 33 35 79 33 46 66 94 9 40 13 66

12 46 37 62 41 8 88 22

28 9 45 56 45 19 87 40

7 45 25 86 51 13 56 45 17 6 56 11

43 32 68 34

29 29 30 73

> 5 3 47 26 1 32 17 65

> 26 26 27 35 41 7 75 16

6 13 11 20

8 2 23 25

Sample Output 11

Sample Input 12

36 22 56 28 12 46 14 94

https://www.hackerrank.com/contests/aplmidsem2017/challenges/letstangle

27 31 58 64

49 24 90 58 31 38 41 57 43 12 80 55

22 44 73 80

```
49 28 84 55
40 52 58 78
11 40 46 58
49 45 54 77
1 4 24 47
41 30 79 81
1 14 24 16
22 18 28 36
20 24 45 31
43 12 67 45
38 52 38 65
37 45 60 84
25 29 53 42
1
S
```

Sample Output 12

6 4 3

Sample Input 13

> 35 49 76 77 25 29 58 57

14 42 33 57

Sample Output 13

35154

P 10 20

Sample Input 14

500 48 46 67 61 39 1 70 47 7 11 8 20

36 52 71 79

52 30 78 71

41 51 83 101

> 9 2 41 4 28 35 72 48

Sample Output 14

P 111 128

6266

Sample Input 15

- 24 42 42 70 43 39 59 43 40 2 83 53 28 15 60 63 45 41 87 44 9 1 38 8 18 13 42 21 5 13 13 42 5 5 40 54 30 0 74 17 14 47 48 89 45 37 87 53 31 20 36 60 13 47 34 78 45 18 53 47 48 5 58 5 45 28 55 29 34 18 42 66 42 16 87 30 34 30 35 42 38 52 88 82 18 8 64 50 33 18 59 25 15 8 18 16 45 41 60 67 23 0 30 45 52 6 69 39 50 24 86 38 13 44 36 54 48 0 79 29 36 36 86 87 0 38 23 84 18 51 45 71 13 44 44 88 27 3 77 6 13 49 40 75 50 15 69 39 50 32 65 44 45 42 93 66 50 17 78 32 34 15 82 62 5 38 11 70 36 14 56 63 47 1 57 24 29 47 45 52 11 49 17 84 1 6 39 57 45 46 69 72 51 52 59 55 14 14 30 64 36 25 64 55 17 32 22 57 1 30 28 42 33 26 38 39 43 43 75 79 35 41 71 74 23 15 64 53 4 14 33 55 36 50 55 50 24 25 54 50 36 16 70 64 39 8 81 38 22 45 52 49 14 17 47 54 0 1 32 6 41 25 89 49 14 4 36 42 31 43 39 89 43 41 81 70 28 38 56 67 6 12 15 64 22 36 51 58 41 27 58 56 12 32 43 58 33 43 48 54 24 5 36 19 51 22 76 48 33 34 73 52 28 17 74 67 23 20 55 63 49 51 75 91 37 45 67 62 25 7 40 56 27 42 39 67 17 30 60 59 9 47 52 85 37 14 48 53
- https://www.hackerrank.com/contests/aplmidsem2017/challenges/letstangle

45 5 96 56

https://www.hackerrank.com/contests/aplmidsem2017/challenges/letstangle

https://www.hackerrank.com/contests/aplmidsem2017/challenges/letstangle

- 45 2 54 26
- 44 42 57 88
- 52 17 101 55
- 38 37 43 54
- 36 36 70 64
- 6 5 47 10
- 35 35 78 48
- 37 36 37 65
- 9 32 26 65
- 30 26 51 35
- 49 47 71 81 27 30 58 72
- 45 18 59 69
- 34 35 57 83
- 45 8 62 16
- 17 16 40 21
- 40 38 89 88
- 17 38 60 52
- 13 27 45 67
- 43 30 48 44
- 51 12 99 44
- 50 28 97 71 43 30 58 37
- 39 44 86 49
- 43 34 73 73 20 0 39 12
- 31 0 38 21
- 12 14 21 24
- 36 42 62 75
- 52 2 101 44
- 12 28 45 58
- 27 14 46 63
- 28 35 56 84
- 47 40 62 44
- 22 25 62 59
- 28 23 67 34
- 44 45 56 67
- 41 11 67 11
- 6 9 24 21 38 8 40 21
- 30 41 53 45
- 17 8 46 47
- 48 52 61 76
- 31 14 53 37
- 13 24 19 25
- 12 33 27 51 33 10 54 28
- 24 11 74 44
- 20 37 72 74
- 7 2 53 37
- 6 6 39 43
- 34 39 54 66
- 6 7 17 25 26 15 67 21
- 41 4 66 48
- 2 25 17 47
- 46 39 87 72
- 34 15 54 34
- 33 4 33 18
- 52 41 74 46
- 33 2 82 40
- 16 24 66 60 1 28 29 31
- 22 5 54 52
- 52 27 96 40
- 43 32 64 55
- 21 17 36 38
- 38 5 43 55 33 15 41 43
- 37 12 76 50
- 14 42 33 57
- 30 9 77 17
- 17 21 33 28 37 10 38 48
- 48 6 75 39
- 11 10 22 34
- 51 52 76 87 11 0 54 4
- 41 51 83 101
- 15 6 54 38
- 2 18 29 37 7 4 35 7
- 34 15 76 61 32 17 57 48
- 29 45 45 64
- 39 23 84 50
- 13 4 34 11

Sample Output 15

5120

Sample Input 16

5 13 13 42

24 29 32 36

- 52 16 100 65 52 25 82 28 27 14 34 47 46 28 74 50 25 8 55 22 5 18 39 62 27 22 48 73 29 38 80 67 25 4 58 28 10 7 18 23 15 49 36 89 49 9 74 9 2 46 51 86 30 31 41 68 32 14 32 23 30 17 61 51 11 37 11 37 8 17 52 20 12 43 25 83 17 20 49 72 7 39 20 77 25 1 42 5 49 45 96 71 26 7 67 44 41 17 85 46 15 32 49 38 47 47 48 91 33 43 79 84 6 5 35 15 46 15 84 57 32 47 71 84 7 1 40 28 24 3 42 21 52 25 66 50 16 16 35 65 44 16 50 66 14 8 14 15 17 28 19 57 20 13 21 40 34 2 48 40 32 3 69 13 52 36 80 51 26 43 57 60 38 47 77 99 39 39 41 75 36 12 82 15 48 31 73 60 40 46 52 97 49 9 99 57 4 10 28 41 2 27 35 46 19 51 40 88 2 20 18 58 9 42 41 78 37 45 89 48 43 1 81 40 35 14 77 54 45 50 69 98 36 46 60 48 46 19 90 46 7 13 46 62 51 32 85 67 12 38 36 40 10 41 49 65 8 43 10 43 6 27 46 69 44 24 65 40 12 43 55 95 8 48 11 54 16 41 43 69 52 30 78 71 45 12 96 44 32 32 67 70 1 7 40 52 7 40 17 91 32 50 63 69 13 26 58 56 52 37 98 88 52 18 67 41 45 2 54 26 44 42 57 88 52 17 101 55 38 37 43 54 36 36 70 64 6 5 47 10 35 35 78 48
- https://www.hackerrank.com/contests/aplmidsem2017/challenges/letstangle

37 36 37 65

46 51 53 84

31 22 42 29

Sample Output 16

17302

Sample Input 17

48 5 58 5

> 34 3 55 12 36 6 66 17

29 38 80 67

45 8 62 16 17 16 40 21

16 48 50 60

38 9 68 53

Sample Output 17

38801

f in
Submissions: 59
Max Score: 20
Difficulty: Medium
Rate This Challenge:
☆☆☆☆☆
More

```
Current Buffer (saved locally, editable) & 🗘
                                                                                      C++
                                                                                                                        Ö
27 ▼ #include<iostream>
28
29
    using namespace std;
30
31 ▼ class point {
            public:
32
33
            int x_coord;
34
            int y_coord;
35
    };
36 ▼ class Rectangle {
37
38
            point lower_left; // Lower Left point.
39
            point upper_right; // Upper Right point.
40
            int area;
41 ▼
            void readRectangle() {
42
                    cin >> lower_left.x_coord;
43
                    cin >> lower_left.y_coord;
44
                    cin >> upper_right.x_coord;
45
                    cin >> upper_right.y_coord;
46
            }
```

```
47 ▼
             void printRectangle() {
                      cout << "(" << lower_left.x_coord << ", " << lower_left.y_coord << ")";
cout << "("<< upper_right.x_coord << ", " << upper_right.y_coord << ")</pre>
48
49
50
51 };
52 ₹
    int main() {
             int N, M, i;
53
             cin >> N; // number of rectangles.
54
55 ₹
             for (i=0; i < N; i++) {
56
                      // add code to read N input rectangles.
57
             cin >> M; // number of queries.
58
59
             char type;
             for (i=0; i< M; i++) {
60 ₹
                      cin >> type;
if (type == 'I') {
61
62 1
                               int index1, index2;
63
64
                               cin >> index1;
65
                               cin >> index2;
66
                               // add code to process Intersection query.
67
                               continue;
68
                      }
                      if (type == 'A') {
69 ₹
70
                               int index;
71
                               cin >> index;
                               // add code to process Area query.
72
73
                               continue;
74
75
                      if (type == 'C') {
                               int index1, index2;
76
77
                               cin >> index1;
78
                               cin >> index2;
79
                               // add code to process co-ordinates query.
80
                               continue;
81
82 🔻
                      if (type == 'S') {
                               // add code to process sort-query.
83
84
                               continue;
85
                      if (type == 'P') {
86 ▼
87
                               int index1, index2, numHops;
88
                               cin >> index1;
89
                               cin >> index2;
90
                               cin >> numHops;
91
                               // add code to process path query.
92
                      }
93
             }
94
             return 0;
95
    }
96
                                                                                                                         Line: 1 Col: 1
```

1 Upload Code as File ☐ Test against custom input

Run Code

Submit Code

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature