

The Galactic Chessmaster's Challenge

Refer to the problem document link [here](#) for details of the problem.

Note: In case of a tie in the closest pair 2D, resolve it as follows:

1. By ascending squared distance between points
2. If distances are equal, by ascending x-coordinate of the first point
3. If first x-coordinates match, by ascending y-coordinate of the first point
4. If first points are identical, by ascending x-coordinate of the second point
5. If second x-coordinates match, by ascending y-coordinate of the second point
6. If all coordinates are equal, maintain the original input order using point indices

Input Format

Every test case will have one or more of the following commands:

CREATE_2D size

Followed by size lines each having size integers each denoting the quantum states.

SORT_2D ascending

SORT_2D descending

INVERSION_2D

DISPLAY_2D

CLOSEST_2D num_points

Followed by num_points lines each having two space separated integers denoting the coordinates of the players

The input will end with the END command.

Constraints

$1 \leq \text{dimension_of_matrix} \leq 10^3$

$-10^{18} \leq \text{quantam_values} \leq 10^{18} - 1$

$-10^9 \leq \text{coordinate_values_of_points} \leq 10^9$

We will be working with only one matrix per test case (i.e. CREATE_2D command will appear only once).

There can be any number of CLOSEST_2D commands in a test case (followed by their respective number of points). But the total number of pairs including all the CLOSEST_2D commands in a test case will be $\leq 10^5$

There will be no test case with a tie for the CLOSEST_2D command (i.e. even if the Euclidean distance and the x coordinates of two different pair of points are same then the y coordinates will be different)

The DISPLAY_2D command will be called at max 3 times within a test case.

Output Format

The DISPLAY_2D command should output the matrix in a row-wise manner. Each row will be printed on a new line with each element of a row being separated by a space " ".

The INVERSION_2D command will output a single integer on a newline indicating the number of inversions in the matrix.

The CLOSEST_2D command will output four integers on a single line separated by a space " ". The first two integers denote the coordinates of the first point and the next two integers denote the coordinates of the second point. The order of the points (which point to be printed first and which to be printed second) will be the same as they appear in the input.

Sample Input 0

```
CREATE_2D
3
1 -5 0
-5 -5 7
5 9 -1
INVERSION_2D
SORT_2D descending
DISPLAY_2D
CLOSEST_2D
4
9 3
9 6
3 10
-7 8
END
```

Sample Output 0

```
12
1 0 -5
7 -5 -5
9 5 -1
9 3 9 6
```

Sample Input 1

```
CREATE_2D
3
9 8 2
10 -10 9
8 8 6
SORT_2D ascending
INVERSION_2D
SORT_2D ascending
```

```
DISPLAY_2D
END
```

Sample Output 1

```
13
2 8 9
-10 9 10
6 8 8
```

Sample Input 2

```
CLOSEST_2D
4
-8 -2
7 8
6 -1
4 10
CLOSEST_2D
4
9 10
-10 0
8 -1
-2 -3
END
```

Sample Output 2

```
7 8 4 10
-10 0 -2 -3
```

Sample Input 3

```
CREATE_2D
10
59 -25 -34 13 57 -38 -31 91 -96 91
-89 61 -82 48 54 76 -35 -72 -69 -89
89 85 -52 -97 -46 76 50 -56 23 -79
98 -39 37 53 -91 99 70 46 6 -27
-22 99 52 -10 98 62 -5 42 33 6
-64 -6 75 -29 -34 -12 -71 77 87 -82
-18 80 29 10 44 -28 83 -89 23 -65
76 -26 47 83 -92 49 66 53 -89 65
100 31 55 67 -85 2 85 -84 -21 70
-4 7 92 97 -23 98 -63 13 -64 6
INVERSION_2D
SORT_2D ascending
SORT_2D descending
SORT_2D ascending
INVERSION_2D
SORT_2D descending
DISPLAY_2D
END
```

Sample Output 3

```
2304
2050
91 91 59 57 13 -25 -31 -34 -38 -96
```

```
76 61 54 48 -35 -69 -72 -82 -89 -89
89 85 76 50 23 -46 -52 -56 -79 -97
99 98 70 53 46 37 6 -27 -39 -91
99 98 62 52 42 33 6 -5 -10 -22
87 77 75 -6 -12 -29 -34 -64 -71 -82
83 80 44 29 23 10 -18 -28 -65 -89
83 76 66 65 53 49 47 -26 -89 -92
100 85 70 67 55 31 2 -21 -84 -85
98 97 92 13 7 6 -4 -23 -63 -64
```

Sample Input 4

```
CREATE_2D
9
9 -72 49 -37 11 -72 25 -32 -54
-56 50 24 -39 39 -73 70 5 21
20 -46 79 -10 -32 -44 -94 51 -53
-48 60 -44 42 -23 23 26 -35 -49
61 -37 -36 52 25 18 -87 -63 36
87 39 -14 -14 -95 47 -68 -18 51
70 -30 80 38 -96 31 22 -79 -49
-86 23 25 -74 -33 -31 -29 -83 -91
-5 -31 -76 40 -81 -95 50 -61 75
INVERSION_2D
CLOSEST_2D
6
-316 253
-833 -768
-834 -70
-648 -112
665 270
272 291
CLOSEST_2D
6
-560 400
-514 -105
-804 276
-216 21
-676 8
-646 680
SORT_2D ascending
CLOSEST_2D
6
-675 -378
-958 -917
754 513
-862 83
90 -787
590 101
INVERSION_2D
DISPLAY_2D
END
```

Sample Output 4

```
1714
-834 -70 -648 -112
-514 -105 -676 8
754 513 590 101
1526
-72 -72 -54 -37 -32 9 11 25 49
-73 -56 -39 5 21 24 39 50 70
-94 -53 -46 -44 -32 -10 20 51 79
-49 -48 -44 -35 -23 23 26 42 60
-87 -63 -37 -36 18 25 36 52 61
-95 -68 -18 -14 -14 39 47 51 87
```

-96 -79 -49 -30 22 31 3870 80
-91 -86 -83 -74 -33 -31 -29 23 25
-95 -81 -76 -61 -31 -5 40 50 75