**Escape**

You are playing a video game that involves escaping from a dangerous area. Within the area there are DEADLY regions you can't enter, HARMFUL regions that take 1 life for every step you make in them, and NORMAL regions that don't affect you in any way. You will start from (0,0) and have to make it to (500,500) using only Up, Left, Right, and Down steps.

The map will be given as a String[] deadly listing the DEADLY regions and a String[] harmful listing the HARMFUL regions. The elements in each of these parameters will be formatted as follows:

Input format (quotes for clarity): "X1 Y1 X2 Y2" where

(X1, Y1) is one corner of the region and

(X2, Y2) is the other corner of the region

The corners of the region are inclusive bounds (i.e. (4, 1) and (2, 2) include x-values between 4 and 2 inclusive and y-values between 1 and 2 inclusive). All unspecified regions are considered NORMAL. If regions overlap for a particular square, then whichever region is worst takes effect (e.g. DEADLY+HARMFUL = DEADLY, HARMFUL+NORMAL = HARMFUL, HARMFUL+HARMFUL = HARMFUL, DEADLY+NORMAL=DEADLY).

Damage taken at each step occurs based on the destination square and not on the starting square (e.g. if the square (500,500) is HARMFUL you WILL take a point of damage stepping onto it; if the square (0,0) is HARMFUL you WON'T take a point of damage stepping off of it; this works analogously for DEADLY squares).

Return the least amount of life you will have to lose in order to reach the destination. Return -1 if there is no path to the destination. Your character is not allowed to leave the map (i.e. have X or Y less than 0 or greater than 500). You should create the following:

Class: Escape

Method: lowest

Parameters: String[], String[]

Returns: int

Method signature: int lowest(String[] harmful, String[] deadly)

**Notes**

- If two harmful regions overlap, the area where they overlap is exactly the same as non-overlapping harmful regions (i.e. the effect is NOT cumulative, and the overlapping region still takes exactly 1 life)

**Constraints**

- deadly will contain between 0 and 50 elements inclusive

- harmful will contain between 0 and 50 elements inclusive

- Each element of deadly and harmful will be of the form (quotes for clarity): "X1 Y1 X2 Y2"

where X1,Y1,X2, and Y2 are integers between 0 and 500 inclusive and contain no leading zeros

- Each element of deadly and harfmul will contain no leading, trailing or extra whitespace

**Examples**

1)

{"500 0 0 500"}

{"0 0 0 0"}

Returns: 1000

(0,0) is DEADLY but that doesn't affect our path since we never step onto it (only from it). The rest of the map is NORMAL.

2)

{"0 0 250 250","250 250 500 500"}

{"0 251 249 500","251 0 500 249"}

Returns: 1000

Just enough space to get around the DEADLY regions.

3)

{"0 0 250 250","250 250 500 500"}

{"0 250 250 500","250 0 500 250"}

Returns: -1

No way around the DEADLY regions.

4)

{"468 209 456 32",

"71 260 306 427",

"420 90 424 492",

"374 253 54 253",

"319 334 152 431",

"38 93 204 84",

"246 0 434 263",

"12 18 118 461",

"215 462 44 317",

"447 214 28 475",

"3 89 38 125",

"157 108 138 264",

"363 17 333 387",

"457 362 396 324",

"95 27 374 175",

"381 196 265 302",

"105 255 253 134",

"0 308 453 55",

"169 28 313 498",

"103 247 165 376",

"264 287 363 407",

"185 255 110 415",

"475 126 293 112",

"285 200 66 484",

"60 178 461 301",

"347 352 470 479",

"433 130 383 370",

"405 378 117 377",

"403 324 369 133",

"12 63 174 309",

"181 0 356 56",

"473 380 315 378"}

{"250 384 355 234",

"28 155 470 4",

"333 405 12 456",

"329 221 239 215",

"334 20 429 338",

"85 42 188 388",

"219 187 12 111",

"467 453 358 133",

"472 172 257 288",

"412 246 431 86",

"335 22 448 47",

"150 14 149 11",

"224 136 466 328",

"369 209 184 262",

"274 488 425 195",

"55 82 279 253",

"153 201 65 228",

"208 230 132 223",

"369 305 397 267",

"200 145 98 198",

"422 67 252 479",

"231 252 401 190",

"312 20 0 350",

"406 72 207 294",

"488 329 338 326",

"117 264 497 447",

"491 341 139 438",

"40 413 329 290",

"148 245 53 386",

"147 70 186 131",

"300 407 71 183",

"300 186 251 198",

"178 67 487 77",

"98 158 55 433",

"167 231 253 90",

"268 406 81 271",

"312 161 387 153",

"33 442 25 412",

"56 69 177 428",

"5 92 61 247"}

Returns: 254

*From* [*https://community.topcoder.com/tc?module=ProblemDetail&rd=4371&pm=1170*](https://community.topcoder.com/tc?module=ProblemDetail&rd=4371&pm=1170)