**Стрельба по движущемуся противнику**

Штирлиц получил из Центра задание разведать планы фашистов по переброске войск. Советская армия планирует в определённые моменты нанести по фашистам артиллерийские удары, и Штирлиц должен предоставить информацию о том, где будут в эти моменты времени находиться фашисты, чтобы стрелять именно туда. Помогите Штирлицу выполнить свой долг, написав программу, которая будет рассчитывать координаты местонахождения фашистских войск.

**Входные данные:**

На первой строке вводится вещественное число v - скорость перемещения фашистских войск в км/ч с точностью до 2 знаков после запятой (0 < v ≤ 103).

На второй строке вводится натуральное число n - количество точек, образующих полилинию, по которой будут двигаться фашистские войска (0 < n ≤ 103).

Далее на n строках вводится по два числа - координаты каждой точки полилинии движения войск в км. Каждая координата - это вещественное число от -40000 до 40000 с точностью до двух знаков после запятой.

Далее вводится время, через сколько часов после начала маршрута движения фашистских войск планируется артиллерийский удар.

**Выходные данные:**

На одной строке выводится два числа - координаты **x**и **y**, где будут находиться фашистские войска в момент артиллерийского удара с точностью до двух знаков после запятой. Если в момент артиллерийского удара фашисты уже закончат свой маршрут, программа должна вывести прочерк.

**Sample Input 1:**

5.00

3

0.55 -0.45

10.55 -0.45

10.55 9.55

3

**Sample Output 1:**

10.55 4.55

**Sample Input 2:**

11.50

4

2.00 19.00

6.00 16.00

11.00 4.00

17.00 12.00

2

**Sample Output 2:**

14.00 8.00

**Sample Input 3:**

5.00

3

0.55 -0.45

10.55 -0.45

10.55 9.55

5

**Sample Output 3:**

-

**Перехват вражеских планов**

Выполняя задание Центра, Штирлиц остался один в кабинете Мюллера. В бумагах группенфюрера разведчик обнаружил описание шифровки, используемой фашистами для описания перемещений своих войск. Они шифруют цифры, пробелы, точки и дефисы в виде двухэтажных сообщений, в которых каждый символ занимает две соседние строки. В ширину каждый символ может кодироваться количеством символов от 1 до 3. Таблица, которую нашёл Штирлиц, показана ниже.

| **Шифр** | **Символ** | **Примечание** |
| --- | --- | --- |
| | | | 1 |  |
| ^| /\_ | 2 |  |
| \_) \_) | 3 |  |
| |\_|   | | 4 | Шифр в 3 символа в ширину |
| |- \_) | 5 |  |
| / () | 6 |  |
| ~/ / | 7 |  |
| () () | 8 |  |
| ()  / | 9 |  |
| /\ \/ | 0 |  |
| . | . | Вверху пробел, внизу точка |
| \_ | - | Вверху знак подчёркивания, внизу пробел - кодирует минус |
|  |  | Пробел сверху и снизу кодирует пробел |

 Напишите программу, которая будет преобразовывать фашистский шифр в обычные строки.

**Входные данные:**

Вводится две строки, в первой из которых хранится верхняя часть зашифрованного сообщения, а во второй - нижняя часть. Если строки имеют разную длину, то считается, что длина всей шифровки - это длина более короткой строки, т.е. лишний суффикс одной из строк игнорируется. Длина каждой из строк не превышает 106.

**Выходные данные:**

Расшифрованная строка. Если строку расшифровать не удалось, программа должна вывести NICHT.

**Sample Input 1:**

|||\_|||-/\\_) / ~/ ()\_)\_()|\_|!!

|| ||\_)\/\_).()/ ()\_) / |

**Sample Output 1:**

1141503.67 83-94

**Sample Input 2:**

|||\_|||-/\\_)$ / ~/ ()\_)\_()|\_|

|| ||\_)\\ff/\_).()/ ()\_) / |

**Sample Output 2:**

NICHT

## Шифровки из Центра

Оперативную информацию из Центра Штирлиц получает с помощью шифровок. Принцип шифрования такой: с помощью последовательности цифр кодируются фигуры - полилинии, эллипсы и точки. Если эти фигуры нарисовать, то полученная картинка визуально отображает необходимую информацию. Чтобы не рисовать всё вручную, напишите программу, которая будет преобразовывать шифровку в html-файл с svg-изображением того, что написано в шифровке.

Шифровка устроена следующим образом. В начале каждой фигуры идёт число, обозначающее её тип

* 0 - полигон. Следующее число - это количество вершин, дальше идет по два числа на каждую вершину - её координаты.
* 1 - полилиния. Устроена так же, как и полигон.
* 2 - эллипс. Следующие два числа - x и y координаты центра эллипса, следующие два числа - его радиусы по x и по y, следующее число равно либо 0, либо 1. 0 означает, что эллипс закрашенный, 1 что это просто контур
* 3 - конец шифровки

**Входные данные:**

На первой строке через пробел вводятся натуральное число n - количество чисел в шифровке (0 < n ≤ 105) и натуральное число w - толщина линий в изображении, которое требуется получить (от 1 до 5).

Далее вводится n натуральных чисел от 0 до 1000 - сама шифровка. На всех строках, кроме самой последней, вводится по 10 чисел, последняя строка может содержать меньше, если n  не делится на 10.

**Выходные данные:**

На первой строке нужно вывести <svg width="250"  height="150">. На место значение атрибута width нужно поставить x координату правой границы самой правой фигуры, увеличенную на w, на место атрибута height нужно поставить y координату нижней границы самой нижней фигуры, увеличенную на w.

Далее каждую фигуру нужно выводить на отдельной строке. Формат каждой фигуры:

* Полигон: <polygon stroke="black" stroke-width="2px" points="20,30 88,30 88,90"/>, где на место значения атрибута points нужно подставить координаты вершин полигона из шифровки. Обратите внимание на отсутствие пробелов между координатами.
* Полилиния: <polyline stroke="black" stroke-width="2px" fill="none" points="92,30 92,90 160,30"/>, где на место значения атрибута points нужно подставить координаты точек полилинии из шифровки.
* Эллипс: <ellipse stroke="black" stroke-width="2px" fill="white" cx="40" cy="50" rx="20" ry="40"/>, где на место значений атрибутов cx, cy, rx и ry нужно подставить координаты центра и радиусы эллипса из шифровки. Если в шифровке указано, что эллипс должен быть закрашенным, атрибут fill отсутствует.

На место значения атрибута stroke-width необходимо во всех фигурах подставить значение w с суффиксом px.

На последней строке нужно вывести </svg>

Отступы в начале строк нигде не делать.

**Sample Input:**

31 2

0 3 20 30 88 30 88 90 1 4

92 30 92 90 160 30 92 30 2 40

125 20 40 0 2 135 125 20 40 1

3

**Sample Output:**

<svg width="162" height="167">

<polygon stroke="black" stroke-width="2px" points="20,30 88,30 88,90"/>

<polyline stroke="black" stroke-width="2px" fill="none" points="92,30 92,90 160,30 92,30"/>

<ellipse stroke="black" stroke-width="2px" fill="white" cx="40" cy="125" rx="20" ry="40"/>

<ellipse stroke="black" stroke-width="2px" cx="135" cy="125" rx="20" ry="40"/>

</svg>

## Помощь радистке Кэт

Радистка Кэт, прослушивая эфир, столкнулась с такой проблемой: в эфире слышно множество шифровок, но не все они могут интересовать советскую разведку. Её интересуют шифровки с перемещениями фашистских войск, а также шифровки из Центра. Для облегчения работы радистки нужна программа, которая будет определять тип шифровки: из Центра, фашистская или вообще что-то другое. Тогда ей предстоит расшифровывать только те шифровки, которые могут представлять интерес, а не всё подряд.

**Входные данные:**

На первой строке вводится число n - количество строк в шифровке (0 < n ≤ 103).

Далее вводится n строк, которые могут представлять собой либо интересующие разведку шифровки, либо что-то неизвестное. Длина каждой из строк не превышает 1000.

**Выходные данные:**

Если шифровка из Центра, программа должна вывести Blumenstrasse. Если шифровка содержит перемещение фашистских войск, программа должна вывести Leipzigestrasse. В противном случае, программа выводит Krakow

**Sample Input 1:**

2

0 50 40 1 60 80 90 60 120 80

120 80 2 90 120 10 30 1 3

**Sample Output 1:**

Blumenstrasse

**Sample Input 2:**

2

|||\_|||-/\\_) / ~/ ()\_)\_()|\_|

|| ||\_)\/\_).()/ ()\_) / |XX

**Sample Output 2:**

Leipzigestrasse

**Sample Input 3:**

2

8 33 XX 47 41 25 28 25 28

1 61 8 61 8 0 V 0 31 88 7

**Sample Output 3:**

Krakow

## Формат донесения в Центр.

Чтобы отправить донесение в Центр, Штирлиц должен пользоваться той же самой шифровкой. Но для идентификации Центр использует строго определённое начертание цифр, которые требуется изобразить с помощью шифровки. Радистка Кэт передала Штирлицу несколько шифровок от Центра. В одной из них находится эталонное изображение цифр и прочих символов, которые требуется использовать для передачи разведданных. Кроме этого изображения, на этой шифровке будет прозвище Геббельса.

Расшифруйте предложенные шифровки и напишите программу, которая будет зашифровывать данные так, как показано в найденной шифровке.

|  |  |
| --- | --- |
| 1 7 233 291 263 257 340 257 366 289 363 321 349 329 341 371 1 5 232 291 239 321 262 328 277 365 277 366 1 2 280 284 298 357 1 2 330 285 322 351 2 270 300 12 6 1 2 341 300 10 4 1 2 307 384 43 24 0 2 296 380 3 7 1 2 323 381 2 7 1 2 313 427 32 32 0 2 312 428 27 24 0 1 7 350 269 380 250 388 216 377 183 396 215 391 254 362 281 1 7 253 268 227 250 216 212 222 187 211 213 215 253 242 279 2 417 298 51 8 0 2 187 299 47 11 0 1 2 260 276 285 295 1 2 326 294 349 278 1 5 297 408 302 413 314 415 322 412 328 408 0 3 323 373 328 374 326 380 0 4 296 373 291 374 294 381 295 381 1 2 375 268 361 282 3 | 2 124 111 34 51 1 0 12 91 99 84 40 99 78 101 23 114 62 129 11 135 62 157 18 151 79 169 44 158 102 149 116 0 3 91 98 80 81 91 111 0 3 157 105 168 85 157 126 2 123 161 15 12 1 2 123 176 4 12 1 1 18 123 183 123 198 126 214 128 224 130 235 131 253 130 268 127 289 123 303 114 318 108 336 102 352 100 368 100 393 102 411 104 423 107 435 108 436 0 3 122 185 124 202 125 177 0 21 132 239 138 229 144 222 153 215 164 210 175 205 180 204 193 202 201 202 208 200 214 196 216 194 211 207 202 219 187 229 174 232 166 234 156 234 151 234 140 233 136 234 0 14 126 290 120 274 107 259 93 252 81 248 69 246 60 239 63 250 72 260 84 268 94 271 107 273 116 277 121 281 0 13 101 367 108 352 117 343 131 336 147 332 161 332 178 326 173 333 158 346 139 356 125 360 111 367 104 370 0 4 99 384 110 363 103 363 104 363 3 |
| 1 7 233 291 263 257 340 257 366 289 363 321 349 329 341 371 1 5 232 291 239 321 262 328 277 365 277 366 1 2 280 284 298 357 1 2 330 285 322 351 2 270 300 12 6 1 2 341 300 10 4 1 2 307 384 43 24 0 2 296 380 3 7 1 2 323 381 2 7 1 2 313 427 32 32 0 2 312 428 27 24 0 1 7 350 269 380 250 388 216 377 183 396 215 391 254 362 281 1 7 253 268 227 250 216 212 222 187 211 213 215 253 242 279 2 417 298 51 8 0 2 187 299 47 11 0 1 2 260 276 285 295 1 2 326 294 349 278 1 5 297 408 302 413 314 415 322 412 328 408 0 3 323 373 328 374 326 380 0 4 296 373 291 374 294 381 295 381 1 2 375 268 361 282 1 3 20 40 30 20 25 70 2 50 45 10 25 0 2 80 30 7 10 0 2 80 55 10 15 0 2 100 68 2 2 1 1 3 120 20 110 50 130 50 1 2 125 40 125 70 1 3 140 40 150 20 145 70 2 200 57 10 13 0 1 7 190 57 190 50 192 40 195 30 200 22 204 20 208 22 2 230 33 10 13 0 1 8 240 33 240 40 239 50 237 60 234 68 230 70 225 68 222 65 2 250 68 2 2 1 1 3 260 40 270 20 265 70 1 3 280 40 290 20 285 70 1 22 38 90 32 94 24 94 24 110 27 109 30 108 33 108 36 110 38 113 39 116 40 120 40 124 38 134 37 135 36 136 35 138 32 139 28 140 26 140 24 139 22 138 20 136 1 5 50 95 55 90 62 95 70 90 52 140 1 2 57 110 67 115 2 80 138 2 2 1 2 100 100 7 10 0 2 100 125 10 15 0 1 3 120 110 130 90 125 140 1 23 170 95 175 90 182 95 190 90 174 110 177 109 180 108 183 108 186 110 188 113 189 116 190 120 190 124 188 134 187 135 186 136 185 138 182 139 178 140 176 140 174 139 172 138 170 136 1 14 201 105 200 100 202 96 205 92 210 90 215 92 219 96 220 100 219 108 215 115 200 140 205 137 215 140 220 135 2 230 138 2 2 1 2 250 115 10 25 0 2 280 100 7 10 0 2 280 125 10 15 0 3 | 2 141 97 36 51 1 2 171 87 6 40 1 2 110 83 8 36 1 2 125 63 4 24 1 2 153 64 5 27 1 1 13 142 145 140 173 141 200 144 215 149 235 153 255 157 273 163 312 165 341 164 375 157 400 144 426 140 440 2 141 147 12 10 1 2 140 158 3 6 1 2 140 49 3 18 1 2 197 241 47 9 1 2 125 295 34 11 1 2 205 356 39 8 1 2 123 193 17 5 1 1 18 283 54 288 48 298 43 310 43 318 47 321 55 316 63 304 67 314 67 322 68 326 76 328 85 322 96 314 100 301 101 291 100 283 93 282 90 1 3 339 94 346 41 359 91 1 2 342 71 354 71 1 5 372 94 372 42 382 74 388 42 388 91 1 4 400 42 400 89 418 89 419 86 1 2 400 63 415 63 1 2 399 42 415 42 1 3 429 40 429 88 429 90 1 2 428 63 444 63 1 2 444 40 444 89 1 3 453 90 460 42 469 90 1 2 456 69 466 69 1 5 498 35 498 86 518 86 524 94 520 101 1 2 517 85 517 35 1 4 528 40 528 81 536 41 536 79 2 557 60 15 15 0 1 2 557 30 557 95 1 9 583 87 583 30 593 31 604 32 607 38 608 45 605 50 599 53 583 53 1 3 341 162 349 146 349 186 1 3 365 164 417 163 407 167 1 2 408 158 416 163 1 3 439 137 433 166 460 166 1 2 448 151 449 181 1 3 355 203 345 237 367 237 1 2 360 222 360 257 1 2 381 230 424 230 1 2 411 225 423 229 1 2 413 234 422 228 1 11 471 207 469 209 448 209 448 224 464 223 470 225 474 232 474 239 470 244 459 246 448 241 1 14 379 268 376 274 354 274 354 292 363 289 370 289 377 292 381 300 382 308 381 314 375 320 362 322 355 320 351 315 1 3 392 290 428 290 420 293 1 2 418 285 427 289 2 453 271 10 14 0 2 454 305 12 19 0 2 365 341 12 14 0 2 368 374 13 19 0 1 3 391 367 428 367 416 361 1 2 418 372 426 366 1 3 450 370 463 346 463 388 3 |
| 1 4 89 65 125 42 105 115 104 115 1 17 174 59 177 53 186 46 198 46 205 49 205 59 203 81 194 90 190 95 178 102 178 105 175 114 187 107 196 112 200 116 202 121 209 103 1 9 239 46 273 34 256 62 276 66 290 84 284 106 269 117 257 116 243 100 1 4 324 30 309 76 354 70 355 70 1 2 342 48 334 126 1 10 97 149 94 182 113 181 115 187 119 194 120 205 119 218 115 222 110 227 108 227 1 4 98 229 79 223 79 219 80 210 1 2 98 230 109 226 1 3 98 150 117 156 129 148 1 16 205 156 189 143 173 146 166 162 163 175 165 209 173 216 190 217 201 204 199 191 195 189 184 182 177 183 170 185 168 192 165 199 1 7 229 172 234 139 241 149 246 142 252 141 264 136 247 213 1 2 249 180 259 180 2 315 156 11 19 0 2 317 200 14 24 0 2 368 158 15 23 0 1 6 381 167 381 199 377 210 366 222 362 217 354 206 2 447 177 18 45 0 2 137 217 3 2 1 2 228 214 2 3 1 0 6 537 272 578 53 635 270 495 107 685 118 538 270 0 5 569 114 557 175 588 212 612 190 593 116 0 44 51 326 216 262 173 299 340 264 306 303 478 237 446 302 509 245 503 315 582 259 562 314 699 256 647 333 777 274 713 399 785 363 721 408 789 428 715 420 725 439 691 419 687 442 635 387 643 441 545 338 543 369 472 346 501 404 445 347 463 410 373 347 393 406 332 335 358 404 286 346 306 396 240 340 268 401 161 341 193 396 109 334 128 385 34 335 35 335 2 465 22 7 7 0 2 496 26 7 8 0 2 529 25 8 7 0 2 568 25 7 6 0 2 610 25 8 5 0 2 653 34 8 8 0 2 627 50 6 6 0 2 619 82 7 7 0 2 538 66 6 7 0 2 483 55 7 6 0 2 503 84 5 8 0 2 511 52 6 6 0 2 661 199 7 6 0 2 681 166 6 5 0 2 700 137 6 5 0 2 670 81 7 7 0 2 715 87 6 6 0 3 | 2 332 134 2 2 1 2 348 146 2 2 1 2 258 157 2 2 1 2 285 170 2 2 1 2 294 114 2 2 1 2 382 91 2 2 1 2 349 110 2 2 1 2 380 154 2 2 1 2 368 221 2 2 1 2 332 202 2 2 1 2 512 147 2 2 1 2 491 219 2 2 1 2 224 281 2 2 1 2 181 185 2 2 1 2 257 39 2 2 1 2 454 61 2 2 1 2 594 80 2 2 1 2 441 155 2 2 1 2 471 266 2 2 1 2 379 324 2 2 1 0 4 178 347 522 384 571 358 566 389 0 7 112 150 148 156 102 243 115 227 118 171 130 176 140 157 0 3 629 185 636 273 630 275 0 4 708 222 701 305 712 304 713 303 0 3 15 295 103 366 103 365 0 3 62 408 152 393 138 432 0 6 90 101 126 54 115 118 96 58 151 83 91 101 0 6 108 67 121 69 127 77 118 94 106 94 103 77 0 6 681 130 689 188 679 193 677 261 677 182 684 175 2 730 83 23 26 0 2 672 360 21 30 0 2 332 411 93 15 0 2 393 57 40 32 0 2 440 273 13 35 0 2 311 290 31 31 0 3 138 182 139 178 140 176 140 174 139 172 138 170 136 1 14 201 105 200 100 202 96 205 92 210 90 215 92 219 96 220 100 219 108 215 115 200 140 205 137 215 140 220 135 2 230 138 2 2 1 2 250 115 10 25 0 2 280 100 7 10 0 2 280 125 10 15 0 3 |

**Входные данные**.

На первой строке подаётся число n - количество строк в шифровке

На следующих n строках подаётся текст, который требуется зашифровать

**Выходные данные**.

Зашифрованный текст тем способом, который нужно извлечь из предложенных шифровок

**Sample Input:**

2

108.41 69.11

57.81 32.08

**Sample Output:**

1 3 20 40 30 20 25 70 2 50

45 10 25 0 2 80 30 7 10 0

2 80 55 10 15 0 2 100 68 2

2 1 1 3 120 20 110 50 130 50

1 2 125 40 125 70 1 3 140 40

150 20 145 70 2 200 57 10 13 0

1 7 190 57 190 50 192 40 195 30

200 22 204 20 208 22 2 230 33 10

13 0 1 8 240 33 240 40 239 50

237 60 234 68 230 70 225 68 222 65

2 250 68 2 2 1 1 3 260 40

270 20 265 70 1 3 280 40 290 20

285 70 1 22 38 90 32 94 24 94

24 110 27 109 30 108 33 108 36 110

38 113 39 116 40 120 40 124 38 134

37 135 36 136 35 138 32 139 28 140

26 140 24 139 22 138 20 136 1 5

50 95 55 90 62 95 70 90 52 140

1 2 57 110 67 115 2 80 138 2

2 1 2 100 100 7 10 0 2 100

125 10 15 0 1 3 120 110 130 90

125 140 1 23 170 95 175 90 182 95

190 90 174 110 177 109 180 108 183 108

186 110 188 113 189 116 190 120 190 124

188 134 187 135 186 136 185 138 182 139

178 140 176 140 174 139 172 138 170 136

1 14 201 105 200 100 202 96 205 92

210 90 215 92 219 96 220 100 219 108

215 115 200 140 205 137 215 140 220 135

2 230 138 2 2 1 2 250 115 10

25 0 2 280 100 7 10 0 2 280

125 10 15 0 3

## Сообщение для Центра

У Штирлица всё практически готово для того, чтобы отправить в Центр необходимые разведданные. Осталось лишь собрать все компоненты воедино: расшифровать сведения о перемещениях фашистов, рассчитать, где они будут находиться в указанный момент, и отправить в Центр пароль и координаты противника, зашифрованные так, как указано в шифровке из Центра.

**Входные данные:**

Первые две строки - это зашифрованная фашистским шифром скорость перемещения фашистов.

На следующих двух строках вводится так же зашифрованное число n - количество точек в полилинии, по которой будут двигаться фашисты (0 < n ≤ 103).

На следующих n \* 2 строках - зашифрованные координаты точек этих полилиний через пробел (по два числа на каждой строке, но и на каждую строку, в соответствии с фашистским шифром, две строки)

На следующей строке вводится интервал с момента выхода фашистских войск, через который должны быть получены разведданные - вещественное число с точностью до двух знаков после запятой.

При этом в ряде случаев фашистский шифр расшифровать не удастся - это означает, что он содержит какую-то иную информацию.

**Выходные данные:**

Несколько строк должны содержать координаты фашистских войск через указанный интервал на всём протяжении пути. Эти строки должны быть зашифрованы способом из предыдущей задачи. Если шифр не удалось расшифровать, программа должна вместо всего этого вывести слово Schwitz.