

Segunda práctica
Mas Lomo

Primer miembro: Cristian Leonel Mendoza (i0546315)

Segundo miembro: María Castellano Blanco (i0939352)

09 de Mayo de 2023

ESQUEMA DE SINCRONIZACION

SEMAFOROS	INICIAL	MAXIMO
sem_EsperarPadre	0	nTrenes
semaforos[400]	1	1
PADRE		Hijo
<pre>Captura Ctrl+c() LOMO_generar_mapa() LOMO_inicio() for(nTrenes){ creacionHijos } for(nTrenes){ W(sem_EsperarPadre) } terminar()</pre>		<pre>LOMO_trenNuevo() LOMO_peticionAvance() mientras{ if(coord a la que se quiere acceder){ W(semaforo[n]) } W(semaforo[361]) if(coordenada == libre){ coordenada = ocupada; } S(semaforo[361]) W(semaforo[361]) LOMO_avance() if(coordenada == libre){ S(semaforo[n]) } if(tren ha salido entero){ coordenada = libre; } LOMO_espera() S(semaforo[361]) W(semaforo[361]) DeteccionDelInterbloqueo() S(semaforo[361]) }fin mientras for(nTrenes){ S(sem_EsperarPadre) }</pre>
TERMINAR		
<pre>LOMO_fin()</pre>		

```

1 #include <iostream>
2 #include "windows.h"
3 #include "stdio.h"
4 #include "lomo2.h"
5 #include "stdlib.h"
6 #include "winuser.h" //para el temporizador
7 #define MAX 100
8 #define MAX2 400
9
10 int(*LOMO_generar_mapa)(const CHAR*, const CHAR*);
11 int(*LOMO_inicio)(int, int, const char*, const char*);
12 int(*LOMO_trenNuevo)(void);
13 int(*LOMO_peticionAvance)(int, int*, int*);
14 int(*LOMO_avance)(int, int*, int*);
15 char* (*LOMO_getColor)(int);
16 void(*LOMO_espera)(int, int);
17 int(*LOMO_fin)(void);
18 void(*pon_error)(const char*);
19
20 //Globales
21 int mapa[17][75];
22 int vectorIdTrenes[MAX];
23 int nTrenes;
24
25 HANDLE vectorTrenes[MAX];
26 HANDLE trenPadre;
27 HANDLE sem_EsperarPadre;
28 HANDLE semaforos[MAX2];
29
30 HINSTANCE libreria;
31
32
33 /*##### PROTOTIPOS #####*/
34 BOOL WINAPI CtrlHandler(DWORD fdwCtrlType);
35 void terminarPrograma(int signal);
36 DWORD WINAPI funcionTrenesHijos(LPVOID param);
37
38
39 int main(int argc, char* argv[]) {
40     int i;
41     int ret = 0;
42     int tamMaximo = 0;
43
44     trenPadre = GetCurrentThread(); //identificamos el padre
45
46     /////////////////////////////////
47     /// CAMBIO DE LA SENIAL SIGINT A LA MANEJADORA
48     /////////////////////////////////
49     if (!SetConsoleCtrlHandler(CtrlHandler, TRUE)) {
50         printf("\nERROR: no se ha podido cambiar el funcionamiento de SIGINT.\n\n");
51         terminarPrograma(1);
52         exit(1);
53     }
54
55
56     /////////////////////////////////
57     /// VINCULACION DE LA DLL
58     /////////////////////////////////
59     libreria = LoadLibrary(TEXT("lomo2.dll"));

```

```

60     if (libreria == NULL) {
61         fprintf(stderr, "No se ha podido cargar la biblioteca\n");
62         terminarPrograma(1);
63         return 100;
64     }
65
66     LOMO_generar_mapa = (int (*) (const char*, const char*)) GetProcAddress(libreria, "LOMO_generar_mapa");
67     if (LOMO_generar_mapa == NULL) {
68         fprintf(stderr, "No se ha podido cargar LOMO_generar_mapa\n");
69         terminarPrograma(1);
70         return 100;
71     }
72     LOMO_inicio = (int (*) (int, int, const char*, const char*)) GetProcAddress(libreria, "LOMO_inicio");
73     if (LOMO_inicio == NULL) {
74         fprintf(stderr, "No se ha podido cargar LOMO_inicio\n");
75         terminarPrograma(1);
76         return 100;
77     }
78     LOMO_trenNuevo = (int(*)(void)) GetProcAddress(libreria, "LOMO_trenNuevo");
79     if (LOMO_trenNuevo == NULL) {
80         fprintf(stderr, "No se ha podido cargar LOMO_trenNuevo\n");
81         terminarPrograma(1);
82         return 100;
83     }
84     LOMO_peticionAvance = (int (*) (int, int*, int*)) GetProcAddress(libreria, "LOMO_peticionAvance");
85     if (LOMO_peticionAvance == NULL) {
86         fprintf(stderr, "No se ha podido cargar LOMO_peticionAvance\n");
87         terminarPrograma(1);
88         return 100;
89     }
90     LOMO_avance = (int (*) (int, int*, int*)) GetProcAddress(libreria, "LOMO_avance");
91     if (LOMO_avance == NULL) {
92         fprintf(stderr, "No se ha podido cargar LOMO_avance\n");
93         terminarPrograma(1);
94         return 100;
95     }
96     LOMO_getColor = (char* (*) (int)) GetProcAddress(libreria, "LOMO_getColor");
//no estoy seguro
97     if (LOMO_getColor == NULL) {
98         fprintf(stderr, "No se ha podido cargar LOMO_getColor\n");
99         terminarPrograma(1);
100        return 100;
101    }
102    LOMO_espera = (void (*) (int, int)) GetProcAddress(libreria, "LOMO_espera");
103    if (LOMO_espera == NULL) {
104        fprintf(stderr, "No se ha podido cargar LOMO_espera\n");
105        terminarPrograma(1);
106        return 100;
107    }
108    LOMO_fin = (int (*) (void)) GetProcAddress(libreria, "LOMO_fin");
109    if (LOMO_fin == NULL) {
110        fprintf(stderr, "No se ha podido cargar LOMO_fin\n");
111        terminarPrograma(1);
112        return 100;
113    }
114    pon_error = (void (*) (const char*)) GetProcAddress(libreria, "pon_error");

```

```

115     if (pon_error == NULL) {
116         fprintf(stderr, "No se ha podido cargar pon_error\n");
117         terminarPrograma(1);
118         return 100;
119     }
120
121
122     ///////////////////////////////////////////////////
123     ///          COMPROBACIONES INICIALES
124     ///////////////////////////////////////////////////
125     if (argc != 2 && argc != 4) {
126         fprintf(stderr, "Debe introducir:\n lomo2 --mapa: si quiere ver el mapa\n"
127             "lomo2 retardo tamMaximo nTrenes: para ver el funcionamiento de la practica\n");
128         exit(1);
129     }
130     //primer caso, si solo introduce un argumento despues de ./lomo
131     else if (argc == 2) {
132         if (strcmp(argv[1], "--mapa") == 0) { //generamos mapa
133             if (LOMO_generar_mapa("i0546315", "i0939352") == -1) {
134                 perror("ERROR, problema al llamar LOMO_generar_Mapa\n");
135                 terminarPrograma(1);
136                 return 100;
137             }
138         }
139         //si no introduce ni uno ni dos argumentos despues de ./lomo
140         fprintf(stderr, "Debe introducir:\n lomo2 --mapa: si quiere ver el
141             mapa\n lomo2 retardo tamMaximo nTrenes: para ver el funcionamiento de la
142             practica\n");
143         terminarPrograma(1);
144         exit(1);
145     }
146     //segundo caso, si introduce tres argumentos despues de ./lomo
147     //lomo2 retardo tamMaximo nTrenes
148     else if (argc == 4) {
149         ret = atoi(argv[1]);
150         if (ret < 0) {
151             fprintf(stderr, "El primer parametro, retardo, debe ser mayor o igual
152             que 0.\n");
153             terminarPrograma(1);
154             exit(1);
155         }
156         tamMaximo = atoi(argv[2]);
157         if (tamMaximo < 3 || tamMaximo>20) {
158             fprintf(stderr, "El segundo parametro, tamMaximo, debe ser mayor que
159             tres y menor que 20.\n");
160             terminarPrograma(1);
161             exit(1);
162         }
163         nTrenes = atoi(argv[3]);
164         if (nTrenes < 0 || nTrenes>100) {
165             fprintf(stderr, "El tercer parametro, numero de trenes, debe ser mayor
166             que cero y menor que 100.\n");
167             terminarPrograma(1);
168             exit(1);
169         }
170     }

```

```

169 ///////////////////////////////////////////////////////////////////
170 /**
171 * INICIO DE PANTALLA PRINCIPAL
172 */
173 if (LOMO_inicio(ret, tamMaximo, "i0546315", "i0939352") == -1) {
174     fprintf(stderr, "Error al llamar LOMO_inicio\n");
175     pon_error("ERROR, Problema al llamar LOMO_inicio\n");
176     terminarPrograma(1);
177     return 100;
178 }
179 ///////////////////////////////////////////////////////////////////
180 /**
181 * MEDIDAS DE COMUNICACION
182 */
183 sem_EsperarPadre = CreateSemaphore(NULL, 0, nTrenes, NULL);
184 if (sem_EsperarPadre == NULL) {
185     fprintf(stderr, "Error al crear Semaforo EsperarPadre\n");
186     terminarPrograma(1);
187     return 100;
188 }
189 for (int s = 0; s < 392; s++) {
190     semaforos[s] = CreateSemaphore(NULL, 1, 1, NULL);
191     if (semaforos[s] == NULL) {
192         fprintf(stderr, "Error al crear Semaforos\n");
193         terminarPrograma(1);
194         return 100;
195     }
196 }
197
198 //rellenamos un mapa de 0
199 int d, f;
200 for (d = 0; d < 17; d++) {
201     for (f = 0; f < 75; f++) {
202         mapa[d][f] = 0;
203     }
204 }
205
206 ///////////////////////////////////////////////////////////////////
207 /**
208 * CREACION DE TRENES
209 */
210 for (i = 0; i < nTrenes; i++) {
211     vectorTrenes[i] = CreateThread(NULL, 0, funcionTrenesHijos, LPVOID(i), 0,
NULL);
212     if (vectorTrenes[i] == NULL) {
213         fprintf(stderr, "Error al crear trenes\n");
214         terminarPrograma(1);
215         return 100;
216     }
217 }
218
219
220 /**
221 * padre coge tantos recursos como trenes
222 */
223 for (i = 0; i < nTrenes; i++) {
224     if (WaitForSingleObject(sem_EsperarPadre, INFINITE) != WAIT_OBJECT_0) {
225         pon_error("Error wait esperarPadre\n");
226         terminarPrograma(1);
227         return 100;
228     }

```

```

228
229     terminarPrograma(1);
230     return 0;
231 }
232
233 ///////////////////////////////////////////////////
234 ///          FUNCIONES
235 ///////////////////////////////////////////////////
236 DWORD WINAPI funcionTrenesHijos(LPVOID param) {
237     int coordY = 0;
238     int xcab = 0, ycab = 0;
239     int ycola = 0, xcola = 0;
240     int i = (int)param;
241
242     const int tiempoMax = 60000; //1 min en miliseg 60000
243     DWORD tiempoComienzo = GetTickCount(); //nos da el tiempo inicial
244
245     /*##### PETICION DE IDENTIFICADOR #####*/
246     vectorIdTrenes[i] = LOMO_trenNuevo();
247     if (vectorIdTrenes[i] == -1) {
248         pon_error("Error en al asigna id LOMO_trenNuevo\n");
249         terminarPrograma(1);
250         return 100;
251     }
252
253     while (1) {
254
255         ///////////////////////////////////////////////////
256         ///          TEMPORIZADOR DE UN MINUTO
257         ///////////////////////////////////////////////////
258         if ((GetTickCount() - tiempoComienzo) >= tiempoMax) {
259             pon_error("SE TERMINO EL TIEMPO DE EJECUCION MAXIMA");
260             terminarPrograma(1);
261             return 100;
262         }
263
264         /*##### PETICION DE AVANCE #####*/
265         if ((LOMO_peticiOnAvance(vectorIdTrenes[i], &xcab, &ycab)) == -1) {
266             pon_error("Error en la peticionAvance\n");
267             terminarPrograma(1);
268             return 100;
269         }
270
271
272         /*##### WAIT CRUCES #####*/
273         if (xcab == 16 && ycab == 0) {
274             WaitForSingleObject(semaforos[363], INFINITE);
275         }
276         if (xcab == 36 && ycab == 0) {
277             WaitForSingleObject(semaforos[364], INFINITE);
278         }
279         if (xcab == 54 && ycab == 0) {
280             WaitForSingleObject(semaforos[365], INFINITE);
281         }
282         if (xcab == 68 && ycab == 0) {
283             WaitForSingleObject(semaforos[366], INFINITE);
284         }
285         if (xcab == 74 && ycab == 0) {
286             WaitForSingleObject(semaforos[367], INFINITE);
287         }

```

```
288     if (xcab == 16 && ycab == 4) {
289         WaitForSingleObject(semaforos[368], INFINITE);
290     }
291     if (xcab == 16 && ycab == 7) {
292         WaitForSingleObject(semaforos[369], INFINITE);
293     }
294     if (xcab == 36 && ycab == 7) {
295         WaitForSingleObject(semaforos[370], INFINITE);
296     }
297     if (xcab == 54 && ycab == 7) {
298         WaitForSingleObject(semaforos[371], INFINITE);
299     }
300     if (xcab == 68 && ycab == 7) {
301         WaitForSingleObject(semaforos[372], INFINITE);
302     }
303     if (xcab == 74 && ycab == 7) {
304         WaitForSingleObject(semaforos[373], INFINITE);
305     }
306     if (xcab == 0 && ycab == 9) {
307         WaitForSingleObject(semaforos[374], INFINITE);
308     }
309     if (xcab == 16 && ycab == 9) {
310         WaitForSingleObject(semaforos[375], INFINITE);
311     }
312     if (xcab == 0 && ycab == 12) {
313         WaitForSingleObject(semaforos[376], INFINITE);
314     }
315     if (xcab == 16 && ycab == 12) {
316         WaitForSingleObject(semaforos[377], INFINITE);
317     }
318     if (xcab == 36 && ycab == 12) {
319         WaitForSingleObject(semaforos[378], INFINITE);
320     }
321     if (xcab == 54 && ycab == 12) {
322         WaitForSingleObject(semaforos[379], INFINITE);
323     }
324     if (xcab == 68 && ycab == 12) {
325         WaitForSingleObject(semaforos[380], INFINITE);
326     }
327     if (xcab == 74 && ycab == 12) {
328         WaitForSingleObject(semaforos[381], INFINITE);
329     }
330     if (xcab == 0 && ycab == 16) {
331         WaitForSingleObject(semaforos[382], INFINITE);
332     }
333     if (xcab == 16 && ycab == 16) {
334         WaitForSingleObject(semaforos[383], INFINITE);
335     }
336     if (xcab == 36 && ycab == 16) {
337         WaitForSingleObject(semaforos[384], INFINITE);
338     }
339     if (xcab == 54 && ycab == 16) {
340         WaitForSingleObject(semaforos[385], INFINITE);
341     }
342     if (xcab == 68 && ycab == 16) {
343         WaitForSingleObject(semaforos[386], INFINITE);
344     }
345     if (xcab == 74 && ycab == 16) {
346         WaitForSingleObject(semaforos[387], INFINITE);
347     }
```

```
348
349 //primer trayecto
350 if (xcab == 0 && ycab == 0) {
351     WaitForSingleObject(semaforos[360], INFINITE); //sem1
352 }
353 if (xcab == 1 && ycab == 0) {
354     WaitForSingleObject(semaforos[362], INFINITE); //sem3
355 }
356 if (xcab == 2 && ycab == 0) {
357     WaitForSingleObject(semaforos[0], INFINITE);
358 }
359 if (xcab == 3 && ycab == 0) {
360     WaitForSingleObject(semaforos[1], INFINITE);
361 }
362 if (xcab == 4 && ycab == 0) {
363     WaitForSingleObject(semaforos[2], INFINITE);
364 }
365 if (xcab == 5 && ycab == 0) {
366     WaitForSingleObject(semaforos[3], INFINITE);
367 }
368 if (xcab == 6 && ycab == 0) {
369     WaitForSingleObject(semaforos[4], INFINITE);
370 }
371 if (xcab == 7 && ycab == 0) {
372     WaitForSingleObject(semaforos[5], INFINITE);
373 }
374 if (xcab == 8 && ycab == 0) {
375     WaitForSingleObject(semaforos[6], INFINITE);
376 }
377 if (xcab == 9 && ycab == 0) {
378     WaitForSingleObject(semaforos[7], INFINITE);
379 }
380 if (xcab == 10 && ycab == 0) {
381     WaitForSingleObject(semaforos[8], INFINITE);
382 }
383 if (xcab == 11 && ycab == 0) {
384     WaitForSingleObject(semaforos[9], INFINITE);
385 }
386 if (xcab == 12 && ycab == 0) {
387     WaitForSingleObject(semaforos[10], INFINITE);
388 }
389 if (xcab == 13 && ycab == 0) {
390     WaitForSingleObject(semaforos[11], INFINITE);
391 }
392 if (xcab == 14 && ycab == 0) {
393     WaitForSingleObject(semaforos[12], INFINITE);
394 }
395 if (xcab == 15 && ycab == 0) {
396     WaitForSingleObject(semaforos[13], INFINITE);
397 }
398 if (xcab == 17 && ycab == 0) {
399     WaitForSingleObject(semaforos[14], INFINITE);
400 }
401 if (xcab == 18 && ycab == 0) {
402     WaitForSingleObject(semaforos[15], INFINITE);
403 }
404 if (xcab == 19 && ycab == 0) {
405     WaitForSingleObject(semaforos[16], INFINITE);
406 }
407 if (xcab == 20 && ycab == 0) {
```

```
408         WaitForSingleObject(semaforos[17], INFINITE);
409     }
410     if (xcb == 21 && ycab == 0) {
411         WaitForSingleObject(semaforos[18], INFINITE);
412     }
413     if (xcb == 22 && ycab == 0) {
414         WaitForSingleObject(semaforos[19], INFINITE);
415     }
416     if (xcb == 23 && ycab == 0) {
417         WaitForSingleObject(semaforos[20], INFINITE);
418     }
419     if (xcb == 24 && ycab == 0) {
420         WaitForSingleObject(semaforos[21], INFINITE);
421     }
422     if (xcb == 25 && ycab == 0) {
423         WaitForSingleObject(semaforos[22], INFINITE);
424     }
425     if (xcb == 26 && ycab == 0) {
426         WaitForSingleObject(semaforos[23], INFINITE);
427     }
428     if (xcb == 27 && ycab == 0) {
429         WaitForSingleObject(semaforos[24], INFINITE);
430     }
431     if (xcb == 28 && ycab == 0) {
432         WaitForSingleObject(semaforos[25], INFINITE);
433     }
434     if (xcb == 29 && ycab == 0) {
435         WaitForSingleObject(semaforos[26], INFINITE);
436     }
437     if (xcb == 30 && ycab == 0) {
438         WaitForSingleObject(semaforos[27], INFINITE);
439     }
440     if (xcb == 31 && ycab == 0) {
441         WaitForSingleObject(semaforos[28], INFINITE);
442     }
443     if (xcb == 32 && ycab == 0) {
444         WaitForSingleObject(semaforos[29], INFINITE);
445     }
446     if (xcb == 33 && ycab == 0) {
447         WaitForSingleObject(semaforos[30], INFINITE);
448     }
449     if (xcb == 34 && ycab == 0) {
450         WaitForSingleObject(semaforos[31], INFINITE);
451     }
452     if (xcb == 35 && ycab == 0) {
453         WaitForSingleObject(semaforos[32], INFINITE);
454     }
455     if (xcb == 37 && ycab == 0) {
456         WaitForSingleObject(semaforos[33], INFINITE);
457     }
458     if (xcb == 38 && ycab == 0) {
459         WaitForSingleObject(semaforos[34], INFINITE);
460     }
461     if (xcb == 39 && ycab == 0) {
462         WaitForSingleObject(semaforos[35], INFINITE);
463     }
464     if (xcb == 40 && ycab == 0) {
465         WaitForSingleObject(semaforos[36], INFINITE);
466     }
467     if (xcb == 41 && ycab == 0) {
```

```
468     WaitForSingleObject(semaforos[37], INFINITE);
469 }
470 if (xcb == 42 && ycab == 0) {
471     WaitForSingleObject(semaforos[38], INFINITE);
472 }
473 if (xcb == 43 && ycab == 0) {
474     WaitForSingleObject(semaforos[39], INFINITE);
475 }
476 if (xcb == 44 && ycab == 0) {
477     WaitForSingleObject(semaforos[40], INFINITE);
478 }
479 if (xcb == 45 && ycab == 0) {
480     WaitForSingleObject(semaforos[41], INFINITE);
481 }
482 if (xcb == 46 && ycab == 0) {
483     WaitForSingleObject(semaforos[42], INFINITE);
484 }
485 if (xcb == 47 && ycab == 0) {
486     WaitForSingleObject(semaforos[43], INFINITE);
487 }
488 if (xcb == 48 && ycab == 0) {
489     WaitForSingleObject(semaforos[44], INFINITE);
490 }
491 if (xcb == 49 && ycab == 0) {
492     WaitForSingleObject(semaforos[45], INFINITE);
493 }
494 if (xcb == 50 && ycab == 0) {
495     WaitForSingleObject(semaforos[46], INFINITE);
496 }
497 if (xcb == 51 && ycab == 0) {
498     WaitForSingleObject(semaforos[47], INFINITE);
499 }
500 if (xcb == 52 && ycab == 0) {
501     WaitForSingleObject(semaforos[48], INFINITE);
502 }
503 if (xcb == 53 && ycab == 0) {
504     WaitForSingleObject(semaforos[49], INFINITE);
505 }
506 if (xcb == 55 && ycab == 0) {
507     WaitForSingleObject(semaforos[50], INFINITE);
508 }
509 if (xcb == 56 && ycab == 0) {
510     WaitForSingleObject(semaforos[51], INFINITE);
511 }
512 if (xcb == 57 && ycab == 0) {
513     WaitForSingleObject(semaforos[52], INFINITE);
514 }
515 if (xcb == 58 && ycab == 0) {
516     WaitForSingleObject(semaforos[53], INFINITE);
517 }
518 if (xcb == 59 && ycab == 0) {
519     WaitForSingleObject(semaforos[54], INFINITE);
520 }
521 if (xcb == 60 && ycab == 0) {
522     WaitForSingleObject(semaforos[55], INFINITE);
523 }
524 if (xcb == 61 && ycab == 0) {
525     WaitForSingleObject(semaforos[56], INFINITE);
526 }
527 if (xcb == 62 && ycab == 0) {
```

```
528         WaitForSingleObject(semaforos[57], INFINITE);
529     }
530     if (xcab == 63 && ycab == 0) {
531         WaitForSingleObject(semaforos[58], INFINITE);
532     }
533     if (xcab == 64 && ycab == 0) {
534         WaitForSingleObject(semaforos[59], INFINITE);
535     }
536     if (xcab == 65 && ycab == 0) {
537         WaitForSingleObject(semaforos[60], INFINITE);
538     }
539     if (xcab == 66 && ycab == 0) {
540         WaitForSingleObject(semaforos[61], INFINITE);
541     }
542     if (xcab == 67 && ycab == 0) {
543         WaitForSingleObject(semaforos[62], INFINITE);
544     }
545     if (xcab == 69 && ycab == 0) {
546         WaitForSingleObject(semaforos[63], INFINITE);
547     }
548     if (xcab == 70 && ycab == 0) {
549         WaitForSingleObject(semaforos[64], INFINITE);
550     }
551     if (xcab == 71 && ycab == 0) {
552         WaitForSingleObject(semaforos[65], INFINITE);
553     }
554     if (xcab == 72 && ycab == 0) {
555         WaitForSingleObject(semaforos[66], INFINITE);
556     }
557     if (xcab == 73 && ycab == 0) {
558         WaitForSingleObject(semaforos[67], INFINITE);
559     }

560
561 //segunda linea horizontal
562 if (xcab == 0 && ycab == 4) {
563     WaitForSingleObject(semaforos[68], INFINITE);
564 }
565 if (xcab == 1 && ycab == 4) {
566     WaitForSingleObject(semaforos[69], INFINITE);
567 }
568 if (xcab == 2 && ycab == 4) {
569     WaitForSingleObject(semaforos[70], INFINITE);
570 }
571 if (xcab == 3 && ycab == 4) {
572     WaitForSingleObject(semaforos[71], INFINITE);
573 }
574 if (xcab == 4 && ycab == 4) {
575     WaitForSingleObject(semaforos[72], INFINITE);
576 }
577 if (xcab == 5 && ycab == 4) {
578     WaitForSingleObject(semaforos[73], INFINITE);
579 }
580 if (xcab == 6 && ycab == 4) {
581     WaitForSingleObject(semaforos[74], INFINITE);
582 }
583 if (xcab == 7 && ycab == 4) {
584     WaitForSingleObject(semaforos[75], INFINITE);
585 }
586 if (xcab == 8 && ycab == 4) {
587     WaitForSingleObject(semaforos[76], INFINITE);
```

```
588 }
589 if (xcab == 9 && ycab == 4) {
590     WaitForSingleObject(semaforos[77], INFINITE);
591 }
592 if (xcab == 10 && ycab == 4) {
593     WaitForSingleObject(semaforos[78], INFINITE);
594 }
595 if (xcab == 11 && ycab == 4) {
596     WaitForSingleObject(semaforos[79], INFINITE);
597 }
598 if (xcab == 12 && ycab == 4) {
599     WaitForSingleObject(semaforos[80], INFINITE);
600 }
601 if (xcab == 13 && ycab == 4) {
602     WaitForSingleObject(semaforos[81], INFINITE);
603 }
604 if (xcab == 14 && ycab == 4) {
605     WaitForSingleObject(semaforos[82], INFINITE);
606 }
607 if (xcab == 15 && ycab == 4) {
608     WaitForSingleObject(semaforos[83], INFINITE);
609 }
610 //tercera trayectoria horizontal
611 if (xcab == 17 && ycab == 7) {
612     WaitForSingleObject(semaforos[84], INFINITE);
613 }
614 if (xcab == 18 && ycab == 7) {
615     WaitForSingleObject(semaforos[85], INFINITE);
616 }
617 if (xcab == 19 && ycab == 7) {
618     WaitForSingleObject(semaforos[86], INFINITE);
619 }
620 if (xcab == 20 && ycab == 7) {
621     WaitForSingleObject(semaforos[87], INFINITE);
622 }
623 if (xcab == 21 && ycab == 7) {
624     WaitForSingleObject(semaforos[88], INFINITE);
625 }
626 if (xcab == 22 && ycab == 7) {
627     WaitForSingleObject(semaforos[89], INFINITE);
628 }
629 if (xcab == 23 && ycab == 7) {
630     WaitForSingleObject(semaforos[90], INFINITE);
631 }
632 if (xcab == 24 && ycab == 7) {
633     WaitForSingleObject(semaforos[91], INFINITE);
634 }
635 if (xcab == 25 && ycab == 7) {
636     WaitForSingleObject(semaforos[92], INFINITE);
637 }
638 if (xcab == 26 && ycab == 7) {
639     WaitForSingleObject(semaforos[93], INFINITE);
640 }
641 if (xcab == 27 && ycab == 7) {
642     WaitForSingleObject(semaforos[94], INFINITE);
643 }
644 if (xcab == 28 && ycab == 7) {
645     WaitForSingleObject(semaforos[95], INFINITE);
646 }
647 if (xcab == 29 && ycab == 7) {
```

```
648     WaitForSingleObject(semaforos[96], INFINITE);
649 }
650 if (xcb == 30 && ycab == 7) {
651     WaitForSingleObject(semaforos[97], INFINITE);
652 }
653 if (xcb == 31 && ycab == 7) {
654     WaitForSingleObject(semaforos[98], INFINITE);
655 }
656 if (xcb == 32 && ycab == 7) {
657     WaitForSingleObject(semaforos[99], INFINITE);
658 }
659 if (xcb == 33 && ycab == 7) {
660     WaitForSingleObject(semaforos[100], INFINITE);
661 }
662 if (xcb == 34 && ycab == 7) {
663     WaitForSingleObject(semaforos[101], INFINITE);
664 }
665 if (xcb == 35 && ycab == 7) {
666     WaitForSingleObject(semaforos[102], INFINITE);
667 }
668 if (xcb == 37 && ycab == 7) {
669     WaitForSingleObject(semaforos[103], INFINITE);
670 }
671 if (xcb == 38 && ycab == 7) {
672     WaitForSingleObject(semaforos[104], INFINITE);
673 }
674 if (xcb == 39 && ycab == 7) {
675     WaitForSingleObject(semaforos[105], INFINITE);
676 }
677 if (xcb == 40 && ycab == 7) {
678     WaitForSingleObject(semaforos[106], INFINITE);
679 }
680 if (xcb == 41 && ycab == 7) {
681     WaitForSingleObject(semaforos[107], INFINITE);
682 }
683 if (xcb == 42 && ycab == 7) {
684     WaitForSingleObject(semaforos[108], INFINITE);
685 }
686 if (xcb == 43 && ycab == 7) {
687     WaitForSingleObject(semaforos[109], INFINITE);
688 }
689 if (xcb == 44 && ycab == 7) {
690     WaitForSingleObject(semaforos[110], INFINITE);
691 }
692 if (xcb == 45 && ycab == 7) {
693     WaitForSingleObject(semaforos[111], INFINITE);
694 }
695 if (xcb == 46 && ycab == 7) {
696     WaitForSingleObject(semaforos[112], INFINITE);
697 }
698 if (xcb == 47 && ycab == 7) {
699     WaitForSingleObject(semaforos[113], INFINITE);
700 }
701 if (xcb == 48 && ycab == 7) {
702     WaitForSingleObject(semaforos[114], INFINITE);
703 }
704 if (xcb == 49 && ycab == 7) {
705     WaitForSingleObject(semaforos[115], INFINITE);
706 }
707 if (xcb == 50 && ycab == 7) {
```

```
708         WaitForSingleObject(semaforos[116], INFINITE);
709     }
710     if (xcb == 51 && ycab == 7) {
711         WaitForSingleObject(semaforos[117], INFINITE);
712     }
713     if (xcb == 52 && ycab == 7) {
714         WaitForSingleObject(semaforos[118], INFINITE);
715     }
716     if (xcb == 53 && ycab == 7) {
717         WaitForSingleObject(semaforos[119], INFINITE);
718     }
719     if (xcb == 55 && ycab == 7) {
720         WaitForSingleObject(semaforos[120], INFINITE);
721     }
722     if (xcb == 56 && ycab == 7) {
723         WaitForSingleObject(semaforos[121], INFINITE);
724     }
725     if (xcb == 57 && ycab == 7) {
726         WaitForSingleObject(semaforos[122], INFINITE);
727     }
728     if (xcb == 58 && ycab == 7) {
729         WaitForSingleObject(semaforos[123], INFINITE);
730     }
731     if (xcb == 59 && ycab == 7) {
732         WaitForSingleObject(semaforos[124], INFINITE);
733     }
734     if (xcb == 60 && ycab == 7) {
735         WaitForSingleObject(semaforos[125], INFINITE);
736     }
737     if (xcb == 61 && ycab == 7) {
738         WaitForSingleObject(semaforos[126], INFINITE);
739     }
740     if (xcb == 62 && ycab == 7) {
741         WaitForSingleObject(semaforos[127], INFINITE);
742     }
743     if (xcb == 63 && ycab == 7) {
744         WaitForSingleObject(semaforos[128], INFINITE);
745     }
746     if (xcb == 64 && ycab == 7) {
747         WaitForSingleObject(semaforos[129], INFINITE);
748     }
749     if (xcb == 65 && ycab == 7) {
750         WaitForSingleObject(semaforos[130], INFINITE);
751     }
752     if (xcb == 66 && ycab == 7) {
753         WaitForSingleObject(semaforos[131], INFINITE);
754     }
755     if (xcb == 67 && ycab == 7) {
756         WaitForSingleObject(semaforos[132], INFINITE);
757     }
758     if (xcb == 69 && ycab == 7) {
759         WaitForSingleObject(semaforos[133], INFINITE);
760     }
761     if (xcb == 70 && ycab == 7) {
762         WaitForSingleObject(semaforos[134], INFINITE);
763     }
764     if (xcb == 71 && ycab == 7) {
765         WaitForSingleObject(semaforos[135], INFINITE);
766     }
767     if (xcb == 72 && ycab == 7) {
```

```
768     WaitForSingleObject(semaforos[136], INFINITE);
769 }
770 if (xcab == 73 && ycab == 7) {
771     WaitForSingleObject(semaforos[137], INFINITE);
772 }
773 //cuarta trayectoria horizontal
774 if (xcab == 1 && ycab == 9) {
775     WaitForSingleObject(semaforos[138], INFINITE);
776 }
777 if (xcab == 2 && ycab == 9) {
778     WaitForSingleObject(semaforos[139], INFINITE);
779 }
780 if (xcab == 3 && ycab == 9) {
781     WaitForSingleObject(semaforos[140], INFINITE);
782 }
783 if (xcab == 4 && ycab == 9) {
784     WaitForSingleObject(semaforos[141], INFINITE);
785 }
786 if (xcab == 5 && ycab == 9) {
787     WaitForSingleObject(semaforos[142], INFINITE);
788 }
789 if (xcab == 6 && ycab == 9) {
790     WaitForSingleObject(semaforos[143], INFINITE);
791 }
792 if (xcab == 7 && ycab == 9) {
793     WaitForSingleObject(semaforos[144], INFINITE);
794 }
795 if (xcab == 8 && ycab == 9) {
796     WaitForSingleObject(semaforos[145], INFINITE);
797 }
798 if (xcab == 9 && ycab == 9) {
799     WaitForSingleObject(semaforos[146], INFINITE);
800 }
801 if (xcab == 10 && ycab == 9) {
802     WaitForSingleObject(semaforos[147], INFINITE);
803 }
804 if (xcab == 11 && ycab == 9) {
805     WaitForSingleObject(semaforos[148], INFINITE);
806 }
807 if (xcab == 12 && ycab == 9) {
808     WaitForSingleObject(semaforos[149], INFINITE);
809 }
810 if (xcab == 13 && ycab == 9) {
811     WaitForSingleObject(semaforos[150], INFINITE);
812 }
813 if (xcab == 14 && ycab == 9) {
814     WaitForSingleObject(semaforos[151], INFINITE);
815 }
816 if (xcab == 15 && ycab == 9) {
817     WaitForSingleObject(semaforos[152], INFINITE);
818 }
819 //quinta trayectoria horizontal
820 if (xcab == 1 && ycab == 12) {
821     WaitForSingleObject(semaforos[153], INFINITE);
822 }
823 if (xcab == 2 && ycab == 12) {
824     WaitForSingleObject(semaforos[154], INFINITE);
825 }
826 if (xcab == 3 && ycab == 12) {
827     WaitForSingleObject(semaforos[155], INFINITE);
```

```
828     }
829     if (xcab == 4 && ycab == 12) {
830         WaitForSingleObject(semaforos[156], INFINITE);
831     }
832     if (xcab == 5 && ycab == 12) {
833         WaitForSingleObject(semaforos[157], INFINITE);
834     }
835     if (xcab == 6 && ycab == 12) {
836         WaitForSingleObject(semaforos[158], INFINITE);
837     }
838     if (xcab == 7 && ycab == 12) {
839         WaitForSingleObject(semaforos[159], INFINITE);
840     }
841     if (xcab == 8 && ycab == 12) {
842         WaitForSingleObject(semaforos[160], INFINITE);
843     }
844     if (xcab == 9 && ycab == 12) {
845         WaitForSingleObject(semaforos[161], INFINITE);
846     }
847     if (xcab == 10 && ycab == 12) {
848         WaitForSingleObject(semaforos[162], INFINITE);
849     }
850     if (xcab == 11 && ycab == 12) {
851         WaitForSingleObject(semaforos[163], INFINITE);
852     }
853     if (xcab == 12 && ycab == 12) {
854         WaitForSingleObject(semaforos[164], INFINITE);
855     }
856     if (xcab == 13 && ycab == 12) {
857         WaitForSingleObject(semaforos[165], INFINITE);
858     }
859     if (xcab == 14 && ycab == 12) {
860         WaitForSingleObject(semaforos[166], INFINITE);
861     }
862     if (xcab == 15 && ycab == 12) {
863         WaitForSingleObject(semaforos[167], INFINITE);
864     }
865     if (xcab == 17 && ycab == 12) {
866         WaitForSingleObject(semaforos[168], INFINITE);
867     }
868     if (xcab == 18 && ycab == 12) {
869         WaitForSingleObject(semaforos[169], INFINITE);
870     }
871     if (xcab == 19 && ycab == 12) {
872         WaitForSingleObject(semaforos[170], INFINITE);
873     }
874     if (xcab == 20 && ycab == 12) {
875         WaitForSingleObject(semaforos[171], INFINITE);
876     }
877     if (xcab == 21 && ycab == 12) {
878         WaitForSingleObject(semaforos[172], INFINITE);
879     }
880     if (xcab == 22 && ycab == 12) {
881         WaitForSingleObject(semaforos[173], INFINITE);
882     }
883     if (xcab == 23 && ycab == 12) {
884         WaitForSingleObject(semaforos[174], INFINITE);
885     }
886     if (xcab == 24 && ycab == 12) {
887         WaitForSingleObject(semaforos[175], INFINITE);
```

```
888 }
889 if (xcb == 25 && ycab == 12) {
890     WaitForSingleObject(semaforos[176], INFINITE);
891 }
892 if (xcb == 26 && ycab == 12) {
893     WaitForSingleObject(semaforos[177], INFINITE);
894 }
895 if (xcb == 27 && ycab == 12) {
896     WaitForSingleObject(semaforos[178], INFINITE);
897 }
898 if (xcb == 28 && ycab == 12) {
899     WaitForSingleObject(semaforos[179], INFINITE);
900 }
901 if (xcb == 29 && ycab == 12) {
902     WaitForSingleObject(semaforos[180], INFINITE);
903 }
904 if (xcb == 30 && ycab == 12) {
905     WaitForSingleObject(semaforos[181], INFINITE);
906 }
907 if (xcb == 31 && ycab == 12) {
908     WaitForSingleObject(semaforos[182], INFINITE);
909 }
910 if (xcb == 32 && ycab == 12) {
911     WaitForSingleObject(semaforos[183], INFINITE);
912 }
913 if (xcb == 33 && ycab == 12) {
914     WaitForSingleObject(semaforos[184], INFINITE);
915 }
916 if (xcb == 34 && ycab == 12) {
917     WaitForSingleObject(semaforos[185], INFINITE);
918 }
919 if (xcb == 35 && ycab == 12) {
920     WaitForSingleObject(semaforos[186], INFINITE);
921 }
922 if (xcb == 37 && ycab == 12) {
923     WaitForSingleObject(semaforos[187], INFINITE);
924 }
925 if (xcb == 38 && ycab == 12) {
926     WaitForSingleObject(semaforos[188], INFINITE);
927 }
928 if (xcb == 39 && ycab == 12) {
929     WaitForSingleObject(semaforos[189], INFINITE);
930 }
931 if (xcb == 40 && ycab == 12) {
932     WaitForSingleObject(semaforos[190], INFINITE);
933 }
934 if (xcb == 41 && ycab == 12) {
935     WaitForSingleObject(semaforos[191], INFINITE);
936 }
937 if (xcb == 42 && ycab == 12) {
938     WaitForSingleObject(semaforos[192], INFINITE);
939 }
940 if (xcb == 43 && ycab == 12) {
941     WaitForSingleObject(semaforos[193], INFINITE);
942 }
943 if (xcb == 44 && ycab == 12) {
944     WaitForSingleObject(semaforos[194], INFINITE);
945 }
946 if (xcb == 45 && ycab == 12) {
947     WaitForSingleObject(semaforos[195], INFINITE);
```

```
948 }
949 if (xcab == 46 && ycab == 12) {
950     WaitForSingleObject(semaforos[196], INFINITE);
951 }
952 if (xcab == 47 && ycab == 12) {
953     WaitForSingleObject(semaforos[197], INFINITE);
954 }
955 if (xcab == 48 && ycab == 12) {
956     WaitForSingleObject(semaforos[198], INFINITE);
957 }
958 if (xcab == 49 && ycab == 12) {
959     WaitForSingleObject(semaforos[199], INFINITE);
960 }
961 if (xcab == 50 && ycab == 12) {
962     WaitForSingleObject(semaforos[200], INFINITE);
963 }
964 if (xcab == 51 && ycab == 12) {
965     WaitForSingleObject(semaforos[201], INFINITE);
966 }
967 if (xcab == 52 && ycab == 12) {
968     WaitForSingleObject(semaforos[202], INFINITE);
969 }
970 if (xcab == 53 && ycab == 12) {
971     WaitForSingleObject(semaforos[203], INFINITE);
972 }
973 if (xcab == 55 && ycab == 12) {
974     WaitForSingleObject(semaforos[204], INFINITE);
975 }
976 if (xcab == 56 && ycab == 12) {
977     WaitForSingleObject(semaforos[205], INFINITE);
978 }
979 if (xcab == 57 && ycab == 12) {
980     WaitForSingleObject(semaforos[206], INFINITE);
981 }
982 if (xcab == 58 && ycab == 12) {
983     WaitForSingleObject(semaforos[207], INFINITE);
984 }
985 if (xcab == 59 && ycab == 12) {
986     WaitForSingleObject(semaforos[208], INFINITE);
987 }
988 if (xcab == 60 && ycab == 12) {
989     WaitForSingleObject(semaforos[209], INFINITE);
990 }
991 if (xcab == 61 && ycab == 12) {
992     WaitForSingleObject(semaforos[210], INFINITE);
993 }
994 if (xcab == 62 && ycab == 12) {
995     WaitForSingleObject(semaforos[211], INFINITE);
996 }
997 if (xcab == 63 && ycab == 12) {
998     WaitForSingleObject(semaforos[212], INFINITE);
999 }
1000 if (xcab == 64 && ycab == 12) {
1001     WaitForSingleObject(semaforos[213], INFINITE);
1002 }
1003 if (xcab == 65 && ycab == 12) {
1004     WaitForSingleObject(semaforos[214], INFINITE);
1005 }
1006 if (xcab == 66 && ycab == 12) {
1007     WaitForSingleObject(semaforos[215], INFINITE);
```

```
1008 }
1009 if (xcab == 67 && ycab == 12) {
1010     WaitForSingleObject(semaforos[216], INFINITE);
1011 }
1012 if (xcab == 69 && ycab == 12) {
1013     WaitForSingleObject(semaforos[217], INFINITE);
1014 }
1015 if (xcab == 70 && ycab == 12) {
1016     WaitForSingleObject(semaforos[218], INFINITE);
1017 }
1018 if (xcab == 71 && ycab == 12) {
1019     WaitForSingleObject(semaforos[219], INFINITE);
1020 }
1021 if (xcab == 72 && ycab == 12) {
1022     WaitForSingleObject(semaforos[220], INFINITE);
1023 }
1024 if (xcab == 73 && ycab == 12) {
1025     WaitForSingleObject(semaforos[221], INFINITE);
1026 }
1027 //sexta trayectoria horizontal
1028 if (xcab == 1 && ycab == 16) {
1029     WaitForSingleObject(semaforos[222], INFINITE);
1030 }
1031 if (xcab == 2 && ycab == 16) {
1032     WaitForSingleObject(semaforos[223], INFINITE);
1033 }
1034 if (xcab == 3 && ycab == 16) {
1035     WaitForSingleObject(semaforos[224], INFINITE);
1036 }
1037 if (xcab == 4 && ycab == 16) {
1038     WaitForSingleObject(semaforos[225], INFINITE);
1039 }
1040 if (xcab == 5 && ycab == 16) {
1041     WaitForSingleObject(semaforos[226], INFINITE);
1042 }
1043 if (xcab == 6 && ycab == 16) {
1044     WaitForSingleObject(semaforos[227], INFINITE);
1045 }
1046 if (xcab == 7 && ycab == 16) {
1047     WaitForSingleObject(semaforos[228], INFINITE);
1048 }
1049 if (xcab == 8 && ycab == 16) {
1050     WaitForSingleObject(semaforos[229], INFINITE);
1051 }
1052 if (xcab == 9 && ycab == 16) {
1053     WaitForSingleObject(semaforos[230], INFINITE);
1054 }
1055 if (xcab == 10 && ycab == 16) {
1056     WaitForSingleObject(semaforos[231], INFINITE);
1057 }
1058 if (xcab == 11 && ycab == 16) {
1059     WaitForSingleObject(semaforos[232], INFINITE);
1060 }
1061 if (xcab == 12 && ycab == 16) {
1062     WaitForSingleObject(semaforos[233], INFINITE);
1063 }
1064 if (xcab == 13 && ycab == 16) {
1065     WaitForSingleObject(semaforos[234], INFINITE);
1066 }
1067 if (xcab == 14 && ycab == 16) {
```

```
1068     WaitForSingleObject(semaforos[235], INFINITE);
1069 }
1070 if (xcb == 15 && ycab == 16) {
1071     WaitForSingleObject(semaforos[236], INFINITE);
1072 }
1073 if (xcb == 17 && ycab == 16) {
1074     WaitForSingleObject(semaforos[237], INFINITE);
1075 }
1076 if (xcb == 18 && ycab == 16) {
1077     WaitForSingleObject(semaforos[238], INFINITE);
1078 }
1079 if (xcb == 19 && ycab == 16) {
1080     WaitForSingleObject(semaforos[239], INFINITE);
1081 }
1082 if (xcb == 20 && ycab == 16) {
1083     WaitForSingleObject(semaforos[240], INFINITE);
1084 }
1085 if (xcb == 21 && ycab == 16) {
1086     WaitForSingleObject(semaforos[241], INFINITE);
1087 }
1088 if (xcb == 22 && ycab == 16) {
1089     WaitForSingleObject(semaforos[242], INFINITE);
1090 }
1091 if (xcb == 23 && ycab == 16) {
1092     WaitForSingleObject(semaforos[243], INFINITE);
1093 }
1094 if (xcb == 24 && ycab == 16) {
1095     WaitForSingleObject(semaforos[244], INFINITE);
1096 }
1097 if (xcb == 25 && ycab == 16) {
1098     WaitForSingleObject(semaforos[245], INFINITE);
1099 }
1100 if (xcb == 26 && ycab == 16) {
1101     WaitForSingleObject(semaforos[246], INFINITE);
1102 }
1103 if (xcb == 27 && ycab == 16) {
1104     WaitForSingleObject(semaforos[247], INFINITE);
1105 }
1106 if (xcb == 28 && ycab == 16) {
1107     WaitForSingleObject(semaforos[248], INFINITE);
1108 }
1109 if (xcb == 29 && ycab == 16) {
1110     WaitForSingleObject(semaforos[249], INFINITE);
1111 }
1112 if (xcb == 30 && ycab == 16) {
1113     WaitForSingleObject(semaforos[250], INFINITE);
1114 }
1115 if (xcb == 31 && ycab == 16) {
1116     WaitForSingleObject(semaforos[251], INFINITE);
1117 }
1118 if (xcb == 32 && ycab == 16) {
1119     WaitForSingleObject(semaforos[252], INFINITE);
1120 }
1121 if (xcb == 33 && ycab == 16) {
1122     WaitForSingleObject(semaforos[253], INFINITE);
1123 }
1124 if (xcb == 34 && ycab == 16) {
1125     WaitForSingleObject(semaforos[254], INFINITE);
1126 }
1127 if (xcb == 35 && ycab == 16) {
```

```
1128     WaitForSingleObject(semaforos[255], INFINITE);
1129 }
1130 if (xcb == 37 && ycab == 16) {
1131     WaitForSingleObject(semaforos[256], INFINITE);
1132 }
1133 if (xcb == 38 && ycab == 16) {
1134     WaitForSingleObject(semaforos[257], INFINITE);
1135 }
1136 if (xcb == 39 && ycab == 16) {
1137     WaitForSingleObject(semaforos[258], INFINITE);
1138 }
1139 if (xcb == 40 && ycab == 16) {
1140     WaitForSingleObject(semaforos[259], INFINITE);
1141 }
1142 if (xcb == 41 && ycab == 16) {
1143     WaitForSingleObject(semaforos[260], INFINITE);
1144 }
1145 if (xcb == 42 && ycab == 16) {
1146     WaitForSingleObject(semaforos[261], INFINITE);
1147 }
1148 if (xcb == 43 && ycab == 16) {
1149     WaitForSingleObject(semaforos[262], INFINITE);
1150 }
1151 if (xcb == 44 && ycab == 16) {
1152     WaitForSingleObject(semaforos[263], INFINITE);
1153 }
1154 if (xcb == 45 && ycab == 16) {
1155     WaitForSingleObject(semaforos[264], INFINITE);
1156 }
1157 if (xcb == 46 && ycab == 16) {
1158     WaitForSingleObject(semaforos[265], INFINITE);
1159 }
1160 if (xcb == 47 && ycab == 16) {
1161     WaitForSingleObject(semaforos[266], INFINITE);
1162 }
1163 if (xcb == 48 && ycab == 16) {
1164     WaitForSingleObject(semaforos[267], INFINITE);
1165 }
1166 if (xcb == 49 && ycab == 16) {
1167     WaitForSingleObject(semaforos[268], INFINITE);
1168 }
1169 if (xcb == 50 && ycab == 16) {
1170     WaitForSingleObject(semaforos[269], INFINITE);
1171 }
1172 if (xcb == 51 && ycab == 16) {
1173     WaitForSingleObject(semaforos[270], INFINITE);
1174 }
1175 if (xcb == 52 && ycab == 16) {
1176     WaitForSingleObject(semaforos[271], INFINITE);
1177 }
1178 if (xcb == 53 && ycab == 16) {
1179     WaitForSingleObject(semaforos[272], INFINITE);
1180 }
1181 if (xcb == 55 && ycab == 16) {
1182     WaitForSingleObject(semaforos[273], INFINITE);
1183 }
1184 if (xcb == 56 && ycab == 16) {
1185     WaitForSingleObject(semaforos[274], INFINITE);
1186 }
1187 if (xcb == 57 && ycab == 16) {
```

```
1188         WaitForSingleObject(semaforos[275], INFINITE);
1189     }
1190     if (xcb == 58 && ycab == 16) {
1191         WaitForSingleObject(semaforos[276], INFINITE);
1192     }
1193     if (xcb == 59 && ycab == 16) {
1194         WaitForSingleObject(semaforos[277], INFINITE);
1195     }
1196     if (xcb == 60 && ycab == 16) {
1197         WaitForSingleObject(semaforos[278], INFINITE);
1198     }
1199     if (xcb == 61 && ycab == 16) {
1200         WaitForSingleObject(semaforos[279], INFINITE);
1201     }
1202     if (xcb == 62 && ycab == 16) {
1203         WaitForSingleObject(semaforos[280], INFINITE);
1204     }
1205     if (xcb == 63 && ycab == 16) {
1206         WaitForSingleObject(semaforos[281], INFINITE);
1207     }
1208     if (xcb == 64 && ycab == 16) {
1209         WaitForSingleObject(semaforos[282], INFINITE);
1210     }
1211     if (xcb == 65 && ycab == 16) {
1212         WaitForSingleObject(semaforos[283], INFINITE);
1213     }
1214     if (xcb == 66 && ycab == 16) {
1215         WaitForSingleObject(semaforos[284], INFINITE);
1216     }
1217     if (xcb == 67 && ycab == 16) {
1218         WaitForSingleObject(semaforos[285], INFINITE);
1219     }
1220     if (xcb == 69 && ycab == 16) {
1221         WaitForSingleObject(semaforos[286], INFINITE);
1222     }
1223     if (xcb == 70 && ycab == 16) {
1224         WaitForSingleObject(semaforos[287], INFINITE);
1225     }
1226     if (xcb == 71 && ycab == 16) {
1227         WaitForSingleObject(semaforos[288], INFINITE);
1228     }
1229     if (xcb == 72 && ycab == 16) {
1230         WaitForSingleObject(semaforos[289], INFINITE);
1231     }
1232     if (xcb == 73 && ycab == 16) {
1233         WaitForSingleObject(semaforos[290], INFINITE);
1234     }

1235 //primer trayectoria vertical
1236 if (xcb == 0 && ycab == 4) {
1237     WaitForSingleObject(semaforos[291], INFINITE);
1238 }
1239 if (xcb == 0 && ycab == 5) {
1240     WaitForSingleObject(semaforos[292], INFINITE);
1241 }
1242 if (xcb == 0 && ycab == 6) {
1243     WaitForSingleObject(semaforos[293], INFINITE);
1244 }
1245 if (xcb == 0 && ycab == 7) {
1246     WaitForSingleObject(semaforos[294], INFINITE);
1247 }
```

```
1248 }
1249 if (xcab == 0 && ycab == 8) {
1250     WaitForSingleObject(semaforos[295], INFINITE);
1251 }
1252 if (xcab == 0 && ycab == 10) {
1253     WaitForSingleObject(semaforos[296], INFINITE);
1254 }
1255 if (xcab == 0 && ycab == 11) {
1256     WaitForSingleObject(semaforos[297], INFINITE);
1257 }
1258 if (xcab == 0 && ycab == 13) {
1259     WaitForSingleObject(semaforos[298], INFINITE);
1260 }
1261 if (xcab == 0 && ycab == 14) {
1262     WaitForSingleObject(semaforos[299], INFINITE);
1263 }
1264 if (xcab == 0 && ycab == 15) {
1265     WaitForSingleObject(semaforos[300], INFINITE);
1266 }
1267 //segunda trayectoria vertical
1268 if (xcab == 16 && ycab == 1) {
1269     WaitForSingleObject(semaforos[301], INFINITE);
1270 }
1271 if (xcab == 16 && ycab == 2) {
1272     WaitForSingleObject(semaforos[302], INFINITE);
1273 }
1274 if (xcab == 16 && ycab == 3) {
1275     WaitForSingleObject(semaforos[303], INFINITE);
1276 }
1277 if (xcab == 16 && ycab == 5) {
1278     WaitForSingleObject(semaforos[304], INFINITE);
1279 }
1280 if (xcab == 16 && ycab == 6) {
1281     WaitForSingleObject(semaforos[305], INFINITE);
1282 }
1283 if (xcab == 16 && ycab == 8) {
1284     WaitForSingleObject(semaforos[306], INFINITE);
1285 }
1286 if (xcab == 16 && ycab == 10) {
1287     WaitForSingleObject(semaforos[307], INFINITE);
1288 }
1289 if (xcab == 16 && ycab == 11) {
1290     WaitForSingleObject(semaforos[308], INFINITE);
1291 }
1292 if (xcab == 16 && ycab == 13) {
1293     WaitForSingleObject(semaforos[309], INFINITE);
1294 }
1295 if (xcab == 16 && ycab == 14) {
1296     WaitForSingleObject(semaforos[310], INFINITE);
1297 }
1298 if (xcab == 16 && ycab == 15) {
1299     WaitForSingleObject(semaforos[311], INFINITE);
1300 }
1301 //tercera trayectoria vertical
1302 if (xcab == 36 && ycab == 1) {
1303     WaitForSingleObject(semaforos[312], INFINITE);
1304 }
1305 if (xcab == 36 && ycab == 2) {
1306     WaitForSingleObject(semaforos[313], INFINITE);
1307 }
```

```
1308     if (xcab == 36 && ycab == 3) {
1309         WaitForSingleObject(semaforos[314], INFINITE);
1310     }
1311     if (xcab == 36 && ycab == 4) {
1312         WaitForSingleObject(semaforos[388], INFINITE);
1313     }
1314     if (xcab == 36 && ycab == 5) {
1315         WaitForSingleObject(semaforos[315], INFINITE);
1316     }
1317     if (xcab == 36 && ycab == 6) {
1318         WaitForSingleObject(semaforos[316], INFINITE);
1319     }
1320     if (xcab == 36 && ycab == 8) {
1321         WaitForSingleObject(semaforos[317], INFINITE);
1322     }
1323     if (xcab == 36 && ycab == 9) {
1324         WaitForSingleObject(semaforos[318], INFINITE);
1325     }
1326     if (xcab == 36 && ycab == 10) {
1327         WaitForSingleObject(semaforos[319], INFINITE);
1328     }
1329     if (xcab == 36 && ycab == 11) {
1330         WaitForSingleObject(semaforos[320], INFINITE);
1331     }
1332     if (xcab == 36 && ycab == 13) {
1333         WaitForSingleObject(semaforos[321], INFINITE);
1334     }
1335     if (xcab == 36 && ycab == 14) {
1336         WaitForSingleObject(semaforos[322], INFINITE);
1337     }
1338     if (xcab == 36 && ycab == 15) {
1339         WaitForSingleObject(semaforos[323], INFINITE);
1340     }
1341 //cuarta trayectoria
1342     if (xcab == 54 && ycab == 1) {
1343         WaitForSingleObject(semaforos[324], INFINITE);
1344     }
1345     if (xcab == 54 && ycab == 2) {
1346         WaitForSingleObject(semaforos[325], INFINITE);
1347     }
1348     if (xcab == 54 && ycab == 3) {
1349         WaitForSingleObject(semaforos[326], INFINITE);
1350     }
1351     if (xcab == 54 && ycab == 4) {
1352         WaitForSingleObject(semaforos[389], INFINITE);
1353     }
1354     if (xcab == 54 && ycab == 5) {
1355         WaitForSingleObject(semaforos[327], INFINITE);
1356     }
1357     if (xcab == 54 && ycab == 6) {
1358         WaitForSingleObject(semaforos[328], INFINITE);
1359     }
1360     if (xcab == 54 && ycab == 8) {
1361         WaitForSingleObject(semaforos[329], INFINITE);
1362     }
1363     if (xcab == 54 && ycab == 9) {
1364         WaitForSingleObject(semaforos[330], INFINITE);
1365     }
1366     if (xcab == 54 && ycab == 10) {
1367         WaitForSingleObject(semaforos[331], INFINITE);
```

```

1368 }
1369 if (xcab == 54 && ycab == 11) {
1370     WaitForSingleObject(semaforos[332], INFINITE);
1371 }
1372 if (xcab == 54 && ycab == 13) {
1373     WaitForSingleObject(semaforos[333], INFINITE);
1374 }
1375 if (xcab == 54 && ycab == 14) {
1376     WaitForSingleObject(semaforos[334], INFINITE);
1377 }
1378 if (xcab == 54 && ycab == 15) {
1379     WaitForSingleObject(semaforos[335], INFINITE);
1380 }
1381 //quinta trayectoria
1382 if (xcab == 68 && ycab == 1) {
1383     WaitForSingleObject(semaforos[336], INFINITE);
1384 }
1385 if (xcab == 68 && ycab == 2) {
1386     WaitForSingleObject(semaforos[337], INFINITE);
1387 }
1388 if (xcab == 68 && ycab == 3) {
1389     WaitForSingleObject(semaforos[338], INFINITE);
1390 }
1391 if (xcab == 68 && ycab == 4) {
1392     WaitForSingleObject(semaforos[390], INFINITE);
1393 }
1394 if (xcab == 68 && ycab == 5) {
1395     WaitForSingleObject(semaforos[339], INFINITE);
1396 }
1397 if (xcab == 68 && ycab == 6) {
1398     WaitForSingleObject(semaforos[340], INFINITE);
1399 }
1400 if (xcab == 68 && ycab == 8) {
1401     WaitForSingleObject(semaforos[341], INFINITE);
1402 }
1403 if (xcab == 68 && ycab == 9) {
1404     WaitForSingleObject(semaforos[342], INFINITE);
1405 }
1406 if (xcab == 68 && ycab == 10) {
1407     WaitForSingleObject(semaforos[343], INFINITE);
1408 }
1409 if (xcab == 68 && ycab == 11) {
1410     WaitForSingleObject(semaforos[344], INFINITE);
1411 }
1412 if (xcab == 68 && ycab == 13) {
1413     WaitForSingleObject(semaforos[345], INFINITE);
1414 }
1415 if (xcab == 68 && ycab == 14) {
1416     WaitForSingleObject(semaforos[346], INFINITE);
1417 }
1418 if (xcab == 68 && ycab == 15) {
1419     WaitForSingleObject(semaforos[347], INFINITE);
1420 }
1421 //ultima trayectoria vertical
1422 if (xcab == 74 && ycab == 1) {
1423     WaitForSingleObject(semaforos[348], INFINITE);
1424 }
1425 if (xcab == 74 && ycab == 2) {
1426     WaitForSingleObject(semaforos[349], INFINITE);
1427 }

```

```

1428     if (xcab == 74 && ycab == 3) {
1429         WaitForSingleObject(semaforos[350], INFINITE);
1430     }
1431     if (xcab == 74 && ycab == 4) {
1432         WaitForSingleObject(semaforos[391], INFINITE);
1433     }
1434     if (xcab == 74 && ycab == 5) {
1435         WaitForSingleObject(semaforos[351], INFINITE);
1436     }
1437     if (xcab == 74 && ycab == 6) {
1438         WaitForSingleObject(semaforos[352], INFINITE);
1439     }
1440     if (xcab == 74 && ycab == 8) {
1441         WaitForSingleObject(semaforos[353], INFINITE);
1442     }
1443     if (xcab == 74 && ycab == 9) {
1444         WaitForSingleObject(semaforos[354], INFINITE);
1445     }
1446     if (xcab == 74 && ycab == 10) {
1447         WaitForSingleObject(semaforos[355], INFINITE);
1448     }
1449     if (xcab == 74 && ycab == 11) {
1450         WaitForSingleObject(semaforos[356], INFINITE);
1451     }
1452     if (xcab == 74 && ycab == 13) {
1453         WaitForSingleObject(semaforos[357], INFINITE);
1454     }
1455     if (xcab == 74 && ycab == 14) {
1456         WaitForSingleObject(semaforos[358], INFINITE);
1457     }
1458     if (xcab == 74 && ycab == 15) {
1459         WaitForSingleObject(semaforos[359], INFINITE);
1460     }

1461
1462
1463
1464 /*##### VERIFICACION DE AVANCE ####*/
1465
1466 WaitForSingleObject(semaforos[361], INFINITE);
1467 if (mapa[ycab][xcab] == 0) { //si no esta ocupado, lo ponemos ocupado
1468     mapa[ycab][xcab] = 1; //ponemos ocupado
1469 }
1470 ReleaseSemaphore(semaforos[361], 1, NULL);

1471 int coordsig = ycab;

1472 /*##### DECIDE AVANZAR ####*/
1473 WaitForSingleObject(semaforos[361], INFINITE);
1474 if (LOMO_avance(vectorIdTrenes[i], &xcola, &ycola) == -1) {
1475     pon_error("Error en la peticionAvance\n");
1476     terminarPrograma(1);
1477     return 100;
1478 }
1479
1480
1481
1482
1483 /*##### SIGNAL CRUCES ####*/
1484 if (xcola == 16 && ycola == 0) {
1485     ReleaseSemaphore(semaforos[363], 1, NULL); //sem4
1486 }
```

```
1487     if (xcola == 36 && ycola == 0) {
1488         ReleaseSemaphore(semaforos[364], 1, NULL);
1489     }
1490     if (xcola == 54 && ycola == 0) {
1491         ReleaseSemaphore(semaforos[365], 1, NULL);
1492     }
1493     if (xcola == 68 && ycola == 0) {
1494         ReleaseSemaphore(semaforos[366], 1, NULL);
1495     }
1496     if (xcola == 74 && ycola == 0) {
1497         ReleaseSemaphore(semaforos[367], 1, NULL);
1498     }
1499     if (xcola == 16 && ycola == 4) {
1500         ReleaseSemaphore(semaforos[368], 1, NULL);
1501     }
1502     if (xcola == 16 && ycola == 7) {
1503         ReleaseSemaphore(semaforos[369], 1, NULL);
1504     }
1505     if (xcola == 36 && ycola == 7) {
1506         ReleaseSemaphore(semaforos[370], 1, NULL);
1507     }
1508     if (xcola == 54 && ycola == 7) {
1509         ReleaseSemaphore(semaforos[371], 1, NULL);
1510     }
1511     if (xcola == 68 && ycola == 7) {
1512         ReleaseSemaphore(semaforos[372], 1, NULL);
1513     }
1514     if (xcola == 74 && ycola == 7) {
1515         ReleaseSemaphore(semaforos[373], 1, NULL);
1516     }
1517     if (xcola == 0 && ycola == 9) {
1518         ReleaseSemaphore(semaforos[374], 1, NULL);
1519     }
1520     if (xcola == 16 && ycola == 9) {
1521         ReleaseSemaphore(semaforos[375], 1, NULL);
1522     }
1523     if (xcola == 0 && ycola == 12) {
1524         ReleaseSemaphore(semaforos[376], 1, NULL);
1525     }
1526     if (xcola == 16 && ycola == 12) {
1527         ReleaseSemaphore(semaforos[377], 1, NULL);
1528     }
1529     if (xcola == 36 && ycola == 12) {
1530         ReleaseSemaphore(semaforos[378], 1, NULL);
1531     }
1532     if (xcola == 54 && ycola == 12) {
1533         ReleaseSemaphore(semaforos[379], 1, NULL);
1534     }
1535     if (xcola == 68 && ycola == 12) {
1536         ReleaseSemaphore(semaforos[380], 1, NULL);
1537     }
1538     if (xcola == 74 && ycola == 12) {
1539         ReleaseSemaphore(semaforos[381], 1, NULL);
1540     }
1541     if (xcola == 0 && ycola == 16) {
1542         ReleaseSemaphore(semaforos[382], 1, NULL);
1543     }
1544     if (xcola == 16 && ycola == 16) {
1545         ReleaseSemaphore(semaforos[383], 1, NULL);
1546     }
```

```
1547     if (xcola == 36 && ycola == 16) {
1548         ReleaseSemaphore(semaforos[384], 1, NULL);
1549     }
1550     if (xcola == 54 && ycola == 16) {
1551         ReleaseSemaphore(semaforos[385], 1, NULL);
1552     }
1553     if (xcola == 68 && ycola == 16) {
1554         ReleaseSemaphore(semaforos[386], 1, NULL);
1555     }
1556     if (xcola == 74 && ycola == 16) {
1557         ReleaseSemaphore(semaforos[387], 1, NULL);
1558     }
1559
1560 //trayectos
1561 //primer trayecto
1562 if (xcola == 0 && ycola == 0) {
1563     ReleaseSemaphore(semaforos[360], 1, NULL);
1564 }
1565 if (xcola == 1 && ycola == 0) {
1566     ReleaseSemaphore(semaforos[362], 1, NULL);
1567 }
1568 if (xcola == 2 && ycola == 0) {
1569     ReleaseSemaphore(semaforos[0], 1, NULL);
1570 }
1571 if (xcola == 3 && ycola == 0) {
1572     ReleaseSemaphore(semaforos[1], 1, NULL);
1573 }
1574 if (xcola == 4 && ycola == 0) {
1575     ReleaseSemaphore(semaforos[2], 1, NULL);
1576 }
1577 if (xcola == 5 && ycola == 0) {
1578     ReleaseSemaphore(semaforos[3], 1, NULL);
1579 }
1580 if (xcola == 6 && ycola == 0) {
1581     ReleaseSemaphore(semaforos[4], 1, NULL);
1582 }
1583 if (xcola == 7 && ycola == 0) {
1584     ReleaseSemaphore(semaforos[5], 1, NULL);
1585 }
1586 if (xcola == 8 && ycola == 0) {
1587     ReleaseSemaphore(semaforos[6], 1, NULL);
1588 }
1589 if (xcola == 9 && ycola == 0) {
1590     ReleaseSemaphore(semaforos[7], 1, NULL);
1591 }
1592 if (xcola == 10 && ycola == 0) {
1593     ReleaseSemaphore(semaforos[8], 1, NULL);
1594 }
1595 if (xcola == 11 && ycola == 0) {
1596     ReleaseSemaphore(semaforos[9], 1, NULL);
1597 }
1598 if (xcola == 12 && ycola == 0) {
1599     ReleaseSemaphore(semaforos[10], 1, NULL);
1600 }
1601 if (xcola == 13 && ycola == 0) {
1602     ReleaseSemaphore(semaforos[11], 1, NULL);
1603 }
1604 if (xcola == 14 && ycola == 0) {
1605     ReleaseSemaphore(semaforos[12], 1, NULL);
1606 }
```

```
1607     if (xcola == 15 && ycola == 0) {
1608         ReleaseSemaphore(semaforos[13], 1, NULL);
1609     }
1610     if (xcola == 17 && ycola == 0) {
1611         ReleaseSemaphore(semaforos[14], 1, NULL);
1612     }
1613     if (xcola == 18 && ycola == 0) {
1614         ReleaseSemaphore(semaforos[15], 1, NULL);
1615     }
1616     if (xcola == 19 && ycola == 0) {
1617         ReleaseSemaphore(semaforos[16], 1, NULL);
1618     }
1619     if (xcola == 20 && ycola == 0) {
1620         ReleaseSemaphore(semaforos[17], 1, NULL);
1621     }
1622     if (xcola == 21 && ycola == 0) {
1623         ReleaseSemaphore(semaforos[18], 1, NULL);
1624     }
1625     if (xcola == 22 && ycola == 0) {
1626         ReleaseSemaphore(semaforos[19], 1, NULL);
1627     }
1628     if (xcola == 23 && ycola == 0) {
1629         ReleaseSemaphore(semaforos[20], 1, NULL);
1630     }
1631     if (xcola == 24 && ycola == 0) {
1632         ReleaseSemaphore(semaforos[21], 1, NULL);
1633     }
1634     if (xcola == 25 && ycola == 0) {
1635         ReleaseSemaphore(semaforos[22], 1, NULL);
1636     }
1637     if (xcola == 26 && ycola == 0) {
1638         ReleaseSemaphore(semaforos[23], 1, NULL);
1639     }
1640     if (xcola == 27 && ycola == 0) {
1641         ReleaseSemaphore(semaforos[24], 1, NULL);
1642     }
1643     if (xcola == 28 && ycola == 0) {
1644         ReleaseSemaphore(semaforos[25], 1, NULL);
1645     }
1646     if (xcola == 29 && ycola == 0) {
1647         ReleaseSemaphore(semaforos[26], 1, NULL);
1648     }
1649     if (xcola == 30 && ycola == 0) {
1650         ReleaseSemaphore(semaforos[27], 1, NULL);
1651     }
1652     if (xcola == 31 && ycola == 0) {
1653         ReleaseSemaphore(semaforos[28], 1, NULL);
1654     }
1655     if (xcola == 32 && ycola == 0) {
1656         ReleaseSemaphore(semaforos[29], 1, NULL);
1657     }
1658     if (xcola == 33 && ycola == 0) {
1659         ReleaseSemaphore(semaforos[30], 1, NULL);
1660     }
1661     if (xcola == 34 && ycola == 0) {
1662         ReleaseSemaphore(semaforos[31], 1, NULL);
1663     }
1664     if (xcola == 35 && ycola == 0) {
1665         ReleaseSemaphore(semaforos[32], 1, NULL);
1666 }
```

```
1667     if (xcola == 37 && ycola == 0) {
1668         ReleaseSemaphore(semaforos[33], 1, NULL);
1669     }
1670     if (xcola == 38 && ycola == 0) {
1671         ReleaseSemaphore(semaforos[34], 1, NULL);
1672     }
1673     if (xcola == 39 && ycola == 0) {
1674         ReleaseSemaphore(semaforos[35], 1, NULL);
1675     }
1676     if (xcola == 40 && ycola == 0) {
1677         ReleaseSemaphore(semaforos[36], 1, NULL);
1678     }
1679     if (xcola == 41 && ycola == 0) {
1680         ReleaseSemaphore(semaforos[37], 1, NULL);
1681     }
1682     if (xcola == 42 && ycola == 0) {
1683         ReleaseSemaphore(semaforos[38], 1, NULL);
1684     }
1685     if (xcola == 43 && ycola == 0) {
1686         ReleaseSemaphore(semaforos[39], 1, NULL);
1687     }
1688     if (xcola == 44 && ycola == 0) {
1689         ReleaseSemaphore(semaforos[40], 1, NULL);
1690     }
1691     if (xcola == 45 && ycola == 0) {
1692         ReleaseSemaphore(semaforos[41], 1, NULL);
1693     }
1694     if (xcola == 46 && ycola == 0) {
1695         ReleaseSemaphore(semaforos[42], 1, NULL);
1696     }
1697     if (xcola == 47 && ycola == 0) {
1698         ReleaseSemaphore(semaforos[43], 1, NULL);
1699     }
1700     if (xcola == 48 && ycola == 0) {
1701         ReleaseSemaphore(semaforos[44], 1, NULL);
1702     }
1703     if (xcola == 49 && ycola == 0) {
1704         ReleaseSemaphore(semaforos[45], 1, NULL);
1705     }
1706     if (xcola == 50 && ycola == 0) {
1707         ReleaseSemaphore(semaforos[46], 1, NULL);
1708     }
1709     if (xcola == 51 && ycola == 0) {
1710         ReleaseSemaphore(semaforos[47], 1, NULL);
1711     }
1712     if (xcola == 52 && ycola == 0) {
1713         ReleaseSemaphore(semaforos[48], 1, NULL);
1714     }
1715     if (xcola == 53 && ycola == 0) {
1716         ReleaseSemaphore(semaforos[49], 1, NULL);
1717     }
1718     if (xcola == 55 && ycola == 0) {
1719         ReleaseSemaphore(semaforos[50], 1, NULL);
1720     }
1721     if (xcola == 56 && ycola == 0) {
1722         ReleaseSemaphore(semaforos[51], 1, NULL);
1723     }
1724     if (xcola == 57 && ycola == 0) {
1725         ReleaseSemaphore(semaforos[52], 1, NULL);
1726 }
```

```
1727     if (xcola == 58 && ycola == 0) {
1728         ReleaseSemaphore(semaforos[53], 1, NULL);
1729     }
1730     if (xcola == 59 && ycola == 0) {
1731         ReleaseSemaphore(semaforos[54], 1, NULL);
1732     }
1733     if (xcola == 60 && ycola == 0) {
1734         ReleaseSemaphore(semaforos[55], 1, NULL);
1735     }
1736     if (xcola == 61 && ycola == 0) {
1737         ReleaseSemaphore(semaforos[56], 1, NULL);
1738     }
1739     if (xcola == 62 && ycola == 0) {
1740         ReleaseSemaphore(semaforos[57], 1, NULL);
1741     }
1742     if (xcola == 63 && ycola == 0) {
1743         ReleaseSemaphore(semaforos[58], 1, NULL);
1744     }
1745     if (xcola == 64 && ycola == 0) {
1746         ReleaseSemaphore(semaforos[59], 1, NULL);
1747     }
1748     if (xcola == 65 && ycola == 0) {
1749         ReleaseSemaphore(semaforos[60], 1, NULL);
1750     }
1751     if (xcola == 66 && ycola == 0) {
1752         ReleaseSemaphore(semaforos[61], 1, NULL);
1753     }
1754     if (xcola == 67 && ycola == 0) {
1755         ReleaseSemaphore(semaforos[62], 1, NULL);
1756     }
1757     if (xcola == 69 && ycola == 0) {
1758         ReleaseSemaphore(semaforos[63], 1, NULL);
1759     }
1760     if (xcola == 70 && ycola == 0) {
1761         ReleaseSemaphore(semaforos[64], 1, NULL);
1762     }
1763     if (xcola == 71 && ycola == 0) {
1764         ReleaseSemaphore(semaforos[65], 1, NULL);
1765     }
1766     if (xcola == 72 && ycola == 0) {
1767         ReleaseSemaphore(semaforos[66], 1, NULL);
1768     }
1769     if (xcola == 73 && ycola == 0) {
1770         ReleaseSemaphore(semaforos[67], 1, NULL);
1771     }
1772 //segunda linea horizontal
1773     if (xcola == 0 && ycola == 4) {
1774         ReleaseSemaphore(semaforos[68], 1, NULL);
1775     }
1776     if (xcola == 1 && ycola == 4) {
1777         ReleaseSemaphore(semaforos[69], 1, NULL);
1778     }
1779     if (xcola == 2 && ycola == 4) {
1780         ReleaseSemaphore(semaforos[70], 1, NULL);
1781     }
1782     if (xcola == 3 && ycola == 4) {
1783         ReleaseSemaphore(semaforos[71], 1, NULL);
1784     }
1785     if (xcola == 4 && ycola == 4) {
1786         ReleaseSemaphore(semaforos[72], 1, NULL);
```

```

1787 }
1788 if (xcola == 5 && ycola == 4) {
1789     ReleaseSemaphore(semaforos[73], 1, NULL);
1790 }
1791 if (xcola == 6 && ycola == 4) {
1792     ReleaseSemaphore(semaforos[74], 1, NULL);
1793 }
1794 if (xcola == 7 && ycola == 4) {
1795     ReleaseSemaphore(semaforos[75], 1, NULL);
1796 }
1797 if (xcola == 8 && ycola == 4) {
1798     ReleaseSemaphore(semaforos[76], 1, NULL);
1799 }
1800 if (xcola == 9 && ycola == 4) {
1801     ReleaseSemaphore(semaforos[77], 1, NULL);
1802 }
1803 if (xcola == 10 && ycola == 4) {
1804     ReleaseSemaphore(semaforos[78], 1, NULL);
1805 }
1806 if (xcola == 11 && ycola == 4) {
1807     ReleaseSemaphore(semaforos[79], 1, NULL);
1808 }
1809 if (xcola == 12 && ycola == 4) {
1810     ReleaseSemaphore(semaforos[80], 1, NULL);
1811 }
1812 if (xcola == 13 && ycola == 4) {
1813     ReleaseSemaphore(semaforos[81], 1, NULL);
1814 }
1815 if (xcola == 14 && ycola == 4) {
1816     ReleaseSemaphore(semaforos[82], 1, NULL);
1817 }
1818 if (xcola == 15 && ycola == 4) {
1819     ReleaseSemaphore(semaforos[83], 1, NULL);
1820 }
//tercera trayectoria horizontal
1821 if (xcola == 17 && ycola == 7) {
1822     ReleaseSemaphore(semaforos[84], 1, NULL);
1823 }
1825 if (xcola == 18 && ycola == 7) {
1826     ReleaseSemaphore(semaforos[85], 1, NULL);
1827 }
1828 if (xcola == 19 && ycola == 7) {
1829     ReleaseSemaphore(semaforos[86], 1, NULL);
1830 }
1831 if (xcola == 20 && ycola == 7) {
1832     ReleaseSemaphore(semaforos[87], 1, NULL);
1833 }
1834 if (xcola == 21 && ycola == 7) {
1835     ReleaseSemaphore(semaforos[88], 1, NULL);
1836 }
1837 if (xcola == 22 && ycola == 7) {
1838     ReleaseSemaphore(semaforos[89], 1, NULL);
1839 }
1840 if (xcola == 23 && ycola == 7) {
1841     ReleaseSemaphore(semaforos[90], 1, NULL);
1842 }
1843 if (xcola == 24 && ycola == 7) {
1844     ReleaseSemaphore(semaforos[91], 1, NULL);
1845 }
1846 if (xcola == 25 && ycola == 7) {

```

```
1847         ReleaseSemaphore(semaforos[92], 1, NULL);
1848     }
1849     if (xcola == 26 && ycola == 7) {
1850         ReleaseSemaphore(semaforos[93], 1, NULL);
1851     }
1852     if (xcola == 27 && ycola == 7) {
1853         ReleaseSemaphore(semaforos[94], 1, NULL);
1854     }
1855     if (xcola == 28 && ycola == 7) {
1856         ReleaseSemaphore(semaforos[95], 1, NULL);
1857     }
1858     if (xcola == 29 && ycola == 7) {
1859         ReleaseSemaphore(semaforos[96], 1, NULL);
1860     }
1861     if (xcola == 30 && ycola == 7) {
1862         ReleaseSemaphore(semaforos[97], 1, NULL);
1863     }
1864     if (xcola == 31 && ycola == 7) {
1865         ReleaseSemaphore(semaforos[98], 1, NULL);
1866     }
1867     if (xcola == 32 && ycola == 7) {
1868         ReleaseSemaphore(semaforos[99], 1, NULL);
1869     }
1870     if (xcola == 33 && ycola == 7) {
1871         ReleaseSemaphore(semaforos[100], 1, NULL);
1872     }
1873     if (xcola == 34 && ycola == 7) {
1874         ReleaseSemaphore(semaforos[101], 1, NULL);
1875     }
1876     if (xcola == 35 && ycola == 7) {
1877         ReleaseSemaphore(semaforos[102], 1, NULL);
1878     }
1879     if (xcola == 37 && ycola == 7) {
1880         ReleaseSemaphore(semaforos[103], 1, NULL);
1881     }
1882     if (xcola == 38 && ycola == 7) {
1883         ReleaseSemaphore(semaforos[104], 1, NULL);
1884     }
1885     if (xcola == 39 && ycola == 7) {
1886         ReleaseSemaphore(semaforos[105], 1, NULL);
1887     }
1888     if (xcola == 40 && ycola == 7) {
1889         ReleaseSemaphore(semaforos[106], 1, NULL);
1890     }
1891     if (xcola == 41 && ycola == 7) {
1892         ReleaseSemaphore(semaforos[107], 1, NULL);
1893     }
1894     if (xcola == 42 && ycola == 7) {
1895         ReleaseSemaphore(semaforos[108], 1, NULL);
1896     }
1897     if (xcola == 43 && ycola == 7) {
1898         ReleaseSemaphore(semaforos[109], 1, NULL);
1899     }
1900     if (xcola == 44 && ycola == 7) {
1901         ReleaseSemaphore(semaforos[110], 1, NULL);
1902     }
1903     if (xcola == 45 && ycola == 7) {
1904         ReleaseSemaphore(semaforos[111], 1, NULL);
1905     }
1906     if (xcola == 46 && ycola == 7) {
```

```
1907         ReleaseSemaphore(semaforos[112], 1, NULL);
1908     }
1909     if (xcola == 47 && ycola == 7) {
1910         ReleaseSemaphore(semaforos[113], 1, NULL);
1911     }
1912     if (xcola == 48 && ycola == 7) {
1913         ReleaseSemaphore(semaforos[114], 1, NULL);
1914     }
1915     if (xcola == 49 && ycola == 7) {
1916         ReleaseSemaphore(semaforos[115], 1, NULL);
1917     }
1918     if (xcola == 50 && ycola == 7) {
1919         ReleaseSemaphore(semaforos[116], 1, NULL);
1920     }
1921     if (xcola == 51 && ycola == 7) {
1922         ReleaseSemaphore(semaforos[117], 1, NULL);
1923     }
1924     if (xcola == 52 && ycola == 7) {
1925         ReleaseSemaphore(semaforos[118], 1, NULL);
1926     }
1927     if (xcola == 53 && ycola == 7) {
1928         ReleaseSemaphore(semaforos[119], 1, NULL);
1929     }
1930     if (xcola == 55 && ycola == 7) {
1931         ReleaseSemaphore(semaforos[120], 1, NULL);
1932     }
1933     if (xcola == 56 && ycola == 7) {
1934         ReleaseSemaphore(semaforos[121], 1, NULL);
1935     }
1936     if (xcola == 57 && ycola == 7) {
1937         ReleaseSemaphore(semaforos[122], 1, NULL);
1938     }
1939     if (xcola == 58 && ycola == 7) {
1940         ReleaseSemaphore(semaforos[123], 1, NULL);
1941     }
1942     if (xcola == 59 && ycola == 7) {
1943         ReleaseSemaphore(semaforos[124], 1, NULL);
1944     }
1945     if (xcola == 60 && ycola == 7) {
1946         ReleaseSemaphore(semaforos[125], 1, NULL);
1947     }
1948     if (xcola == 61 && ycola == 7) {
1949         ReleaseSemaphore(semaforos[126], 1, NULL);
1950     }
1951     if (xcola == 62 && ycola == 7) {
1952         ReleaseSemaphore(semaforos[127], 1, NULL);
1953     }
1954     if (xcola == 63 && ycola == 7) {
1955         ReleaseSemaphore(semaforos[128], 1, NULL);
1956     }
1957     if (xcola == 64 && ycola == 7) {
1958         ReleaseSemaphore(semaforos[129], 1, NULL);
1959     }
1960     if (xcola == 65 && ycola == 7) {
1961         ReleaseSemaphore(semaforos[130], 1, NULL);
1962     }
1963     if (xcola == 66 && ycola == 7) {
1964         ReleaseSemaphore(semaforos[131], 1, NULL);
1965     }
1966     if (xcola == 67 && ycola == 7) {
```

```

1967     ReleaseSemaphore(semaforos[132], 1, NULL);
1968 }
1969 if (xcola == 69 && ycola == 7) {
1970     ReleaseSemaphore(semaforos[133], 1, NULL);
1971 }
1972 if (xcola == 70 && ycola == 7) {
1973     ReleaseSemaphore(semaforos[134], 1, NULL);
1974 }
1975 if (xcola == 71 && ycola == 7) {
1976     ReleaseSemaphore(semaforos[135], 1, NULL);
1977 }
1978 if (xcola == 72 && ycola == 7) {
1979     ReleaseSemaphore(semaforos[136], 1, NULL);
1980 }
1981 if (xcola == 73 && ycola == 7) {
1982     ReleaseSemaphore(semaforos[137], 1, NULL);
1983 }
1984 //cuarta trayectoria horizontal
1985 if (xcola == 1 && ycola == 9) {
1986     ReleaseSemaphore(semaforos[138], 1, NULL);
1987 }
1988 if (xcola == 2 && ycola == 9) {
1989     ReleaseSemaphore(semaforos[139], 1, NULL);
1990 }
1991 if (xcola == 3 && ycola == 9) {
1992     ReleaseSemaphore(semaforos[140], 1, NULL);
1993 }
1994 if (xcola == 4 && ycola == 9) {
1995     ReleaseSemaphore(semaforos[141], 1, NULL);
1996 }
1997 if (xcola == 5 && ycola == 9) {
1998     ReleaseSemaphore(semaforos[142], 1, NULL);
1999 }
2000 if (xcola == 6 && ycola == 9) {
2001     ReleaseSemaphore(semaforos[143], 1, NULL);
2002 }
2003 if (xcola == 7 && ycola == 9) {
2004     ReleaseSemaphore(semaforos[144], 1, NULL);
2005 }
2006 if (xcola == 8 && ycola == 9) {
2007     ReleaseSemaphore(semaforos[145], 1, NULL);
2008 }
2009 if (xcola == 9 && ycola == 9) {
2010     ReleaseSemaphore(semaforos[146], 1, NULL);
2011 }
2012 if (xcola == 10 && ycola == 9) {
2013     ReleaseSemaphore(semaforos[147], 1, NULL);
2014 }
2015 if (xcola == 11 && ycola == 9) {
2016     ReleaseSemaphore(semaforos[148], 1, NULL);
2017 }
2018 if (xcola == 12 && ycola == 9) {
2019     ReleaseSemaphore(semaforos[149], 1, NULL);
2020 }
2021 if (xcola == 13 && ycola == 9) {
2022     ReleaseSemaphore(semaforos[150], 1, NULL);
2023 }
2024 if (xcola == 14 && ycola == 9) {
2025     ReleaseSemaphore(semaforos[151], 1, NULL);
2026 }

```

```
2027     if (xcola == 15 && ycola == 9) {
2028         ReleaseSemaphore(semáforos[152], 1, NULL);
2029     }
2030 //quinta trayectoria horizontal
2031     if (xcola == 1 && ycola == 12) {
2032         ReleaseSemaphore(semáforos[153], 1, NULL);
2033     }
2034     if (xcola == 2 && ycola == 12) {
2035         ReleaseSemaphore(semáforos[154], 1, NULL);
2036     }
2037     if (xcola == 3 && ycola == 12) {
2038         ReleaseSemaphore(semáforos[155], 1, NULL);
2039     }
2040     if (xcola == 4 && ycola == 12) {
2041         ReleaseSemaphore(semáforos[156], 1, NULL);
2042     }
2043     if (xcola == 5 && ycola == 12) {
2044         ReleaseSemaphore(semáforos[157], 1, NULL);
2045     }
2046     if (xcola == 6 && ycola == 12) {
2047         ReleaseSemaphore(semáforos[158], 1, NULL);
2048     }
2049     if (xcola == 7 && ycola == 12) {
2050         ReleaseSemaphore(semáforos[159], 1, NULL);
2051     }
2052     if (xcola == 8 && ycola == 12) {
2053         ReleaseSemaphore(semáforos[160], 1, NULL);
2054     }
2055     if (xcola == 9 && ycola == 12) {
2056         ReleaseSemaphore(semáforos[161], 1, NULL);
2057     }
2058     if (xcola == 10 && ycola == 12) {
2059         ReleaseSemaphore(semáforos[162], 1, NULL);
2060     }
2061     if (xcola == 11 && ycola == 12) {
2062         ReleaseSemaphore(semáforos[163], 1, NULL);
2063     }
2064     if (xcola == 12 && ycola == 12) {
2065         ReleaseSemaphore(semáforos[164], 1, NULL);
2066     }
2067     if (xcola == 13 && ycola == 12) {
2068         ReleaseSemaphore(semáforos[165], 1, NULL);
2069     }
2070     if (xcola == 14 && ycola == 12) {
2071         ReleaseSemaphore(semáforos[166], 1, NULL);
2072     }
2073     if (xcola == 15 && ycola == 12) {
2074         ReleaseSemaphore(semáforos[167], 1, NULL);
2075     }
2076     if (xcola == 17 && ycola == 12) {
2077         ReleaseSemaphore(semáforos[168], 1, NULL);
2078     }
2079     if (xcola == 18 && ycola == 12) {
2080         ReleaseSemaphore(semáforos[169], 1, NULL);
2081     }
2082     if (xcola == 19 && ycola == 12) {
2083         ReleaseSemaphore(semáforos[170], 1, NULL);
2084     }
2085     if (xcola == 20 && ycola == 12) {
2086         ReleaseSemaphore(semáforos[171], 1, NULL);
```

```
2087 }
2088 if (xcola == 21 && ycola == 12) {
2089     ReleaseSemaphore(semaforos[172], 1, NULL);
2090 }
2091 if (xcola == 22 && ycola == 12) {
2092     ReleaseSemaphore(semaforos[173], 1, NULL);
2093 }
2094 if (xcola == 23 && ycola == 12) {
2095     ReleaseSemaphore(semaforos[174], 1, NULL);
2096 }
2097 if (xcola == 24 && ycola == 12) {
2098     ReleaseSemaphore(semaforos[175], 1, NULL);
2099 }
2100 if (xcola == 25 && ycola == 12) {
2101     ReleaseSemaphore(semaforos[176], 1, NULL);
2102 }
2103 if (xcola == 26 && ycola == 12) {
2104     ReleaseSemaphore(semaforos[177], 1, NULL);
2105 }
2106 if (xcola == 27 && ycola == 12) {
2107     ReleaseSemaphore(semaforos[178], 1, NULL);
2108 }
2109 if (xcola == 28 && ycola == 12) {
2110     ReleaseSemaphore(semaforos[179], 1, NULL);
2111 }
2112 if (xcola == 29 && ycola == 12) {
2113     ReleaseSemaphore(semaforos[180], 1, NULL);
2114 }
2115 if (xcola == 30 && ycola == 12) {
2116     ReleaseSemaphore(semaforos[181], 1, NULL);
2117 }
2118 if (xcola == 31 && ycola == 12) {
2119     ReleaseSemaphore(semaforos[182], 1, NULL);
2120 }
2121 if (xcola == 32 && ycola == 12) {
2122     ReleaseSemaphore(semaforos[183], 1, NULL);
2123 }
2124 if (xcola == 33 && ycola == 12) {
2125     ReleaseSemaphore(semaforos[184], 1, NULL);
2126 }
2127 if (xcola == 34 && ycola == 12) {
2128     ReleaseSemaphore(semaforos[185], 1, NULL);
2129 }
2130 if (xcola == 35 && ycola == 12) {
2131     ReleaseSemaphore(semaforos[186], 1, NULL);
2132 }
2133 if (xcola == 37 && ycola == 12) {
2134     ReleaseSemaphore(semaforos[187], 1, NULL);
2135 }
2136 if (xcola == 38 && ycola == 12) {
2137     ReleaseSemaphore(semaforos[188], 1, NULL);
2138 }
2139 if (xcola == 39 && ycola == 12) {
2140     ReleaseSemaphore(semaforos[189], 1, NULL);
2141 }
2142 if (xcola == 40 && ycola == 12) {
2143     ReleaseSemaphore(semaforos[190], 1, NULL);
2144 }
2145 if (xcola == 41 && ycola == 12) {
2146     ReleaseSemaphore(semaforos[191], 1, NULL);
```

```
2147 }
2148 if (xcola == 42 && ycola == 12) {
2149     ReleaseSemaphore(semaforos[192], 1, NULL);
2150 }
2151 if (xcola == 43 && ycola == 12) {
2152     ReleaseSemaphore(semaforos[193], 1, NULL);
2153 }
2154 if (xcola == 44 && ycola == 12) {
2155     ReleaseSemaphore(semaforos[194], 1, NULL);
2156 }
2157 if (xcola == 45 && ycola == 12) {
2158     ReleaseSemaphore(semaforos[195], 1, NULL);
2159 }
2160 if (xcola == 46 && ycola == 12) {
2161     ReleaseSemaphore(semaforos[196], 1, NULL);
2162 }
2163 if (xcola == 47 && ycola == 12) {
2164     ReleaseSemaphore(semaforos[197], 1, NULL);
2165 }
2166 if (xcola == 48 && ycola == 12) {
2167     ReleaseSemaphore(semaforos[198], 1, NULL);
2168 }
2169 if (xcola == 49 && ycola == 12) {
2170     ReleaseSemaphore(semaforos[199], 1, NULL);
2171 }
2172 if (xcola == 50 && ycola == 12) {
2173     ReleaseSemaphore(semaforos[200], 1, NULL);
2174 }
2175 if (xcola == 51 && ycola == 12) {
2176     ReleaseSemaphore(semaforos[201], 1, NULL);
2177 }
2178 if (xcola == 52 && ycola == 12) {
2179     ReleaseSemaphore(semaforos[202], 1, NULL);
2180 }
2181 if (xcola == 53 && ycola == 12) {
2182     ReleaseSemaphore(semaforos[203], 1, NULL);
2183 }
2184 if (xcola == 55 && ycola == 12) {
2185     ReleaseSemaphore(semaforos[204], 1, NULL);
2186 }
2187 if (xcola == 56 && ycola == 12) {
2188     ReleaseSemaphore(semaforos[205], 1, NULL);
2189 }
2190 if (xcola == 57 && ycola == 12) {
2191     ReleaseSemaphore(semaforos[206], 1, NULL);
2192 }
2193 if (xcola == 58 && ycola == 12) {
2194     ReleaseSemaphore(semaforos[207], 1, NULL);
2195 }
2196 if (xcola == 59 && ycola == 12) {
2197     ReleaseSemaphore(semaforos[208], 1, NULL);
2198 }
2199 if (xcola == 60 && ycola == 12) {
2200     ReleaseSemaphore(semaforos[209], 1, NULL);
2201 }
2202 if (xcola == 61 && ycola == 12) {
2203     ReleaseSemaphore(semaforos[210], 1, NULL);
2204 }
2205 if (xcola == 62 && ycola == 12) {
2206     ReleaseSemaphore(semaforos[211], 1, NULL);
```

```

2207 }
2208 if (xcola == 63 && ycola == 12) {
2209     ReleaseSemaphore(semaforos[212], 1, NULL);
2210 }
2211 if (xcola == 64 && ycola == 12) {
2212     ReleaseSemaphore(semaforos[213], 1, NULL);
2213 }
2214 if (xcola == 65 && ycola == 12) {
2215     ReleaseSemaphore(semaforos[214], 1, NULL);
2216 }
2217 if (xcola == 66 && ycola == 12) {
2218     ReleaseSemaphore(semaforos[215], 1, NULL);
2219 }
2220 if (xcola == 67 && ycola == 12) {
2221     ReleaseSemaphore(semaforos[216], 1, NULL);
2222 }
2223 if (xcola == 69 && ycola == 12) {
2224     ReleaseSemaphore(semaforos[217], 1, NULL);
2225 }
2226 if (xcola == 70 && ycola == 12) {
2227     ReleaseSemaphore(semaforos[218], 1, NULL);
2228 }
2229 if (xcola == 71 && ycola == 12) {
2230     ReleaseSemaphore(semaforos[219], 1, NULL);
2231 }
2232 if (xcola == 72 && ycola == 12) {
2233     ReleaseSemaphore(semaforos[220], 1, NULL);
2234 }
2235 if (xcola == 73 && ycola == 12) {
2236     ReleaseSemaphore(semaforos[221], 1, NULL);
2237 }
2238 //sexta trayectoria horizontal
2239 if (xcola == 1 && ycola == 16) {
2240     ReleaseSemaphore(semaforos[222], 1, NULL);
2241 }
2242 if (xcola == 2 && ycola == 16) {
2243     ReleaseSemaphore(semaforos[223], 1, NULL);
2244 }
2245 if (xcola == 3 && ycola == 16) {
2246     ReleaseSemaphore(semaforos[224], 1, NULL);
2247 }
2248 if (xcola == 4 && ycola == 16) {
2249     ReleaseSemaphore(semaforos[225], 1, NULL);
2250 }
2251 if (xcola == 5 && ycola == 16) {
2252     ReleaseSemaphore(semaforos[226], 1, NULL);
2253 }
2254 if (xcola == 6 && ycola == 16) {
2255     ReleaseSemaphore(semaforos[227], 1, NULL);
2256 }
2257 if (xcola == 7 && ycola == 16) {
2258     ReleaseSemaphore(semaforos[228], 1, NULL);
2259 }
2260 if (xcola == 8 && ycola == 16) {
2261     ReleaseSemaphore(semaforos[229], 1, NULL);
2262 }
2263 if (xcola == 9 && ycola == 16) {
2264     ReleaseSemaphore(semaforos[230], 1, NULL);
2265 }
2266 if (xcola == 10 && ycola == 16) {

```

```
2267         ReleaseSemaphore(semaforos[231], 1, NULL);
2268     }
2269     if (xcola == 11 && ycola == 16) {
2270         ReleaseSemaphore(semaforos[232], 1, NULL);
2271     }
2272     if (xcola == 12 && ycola == 16) {
2273         ReleaseSemaphore(semaforos[233], 1, NULL);
2274     }
2275     if (xcola == 13 && ycola == 16) {
2276         ReleaseSemaphore(semaforos[234], 1, NULL);
2277     }
2278     if (xcola == 14 && ycola == 16) {
2279         ReleaseSemaphore(semaforos[235], 1, NULL);
2280     }
2281     if (xcola == 15 && ycola == 16) {
2282         ReleaseSemaphore(semaforos[236], 1, NULL);
2283     }
2284     if (xcola == 17 && ycola == 16) {
2285         ReleaseSemaphore(semaforos[237], 1, NULL);
2286     }
2287     if (xcola == 18 && ycola == 16) {
2288         ReleaseSemaphore(semaforos[238], 1, NULL);
2289     }
2290     if (xcola == 19 && ycola == 16) {
2291         ReleaseSemaphore(semaforos[239], 1, NULL);
2292     }
2293     if (xcola == 20 && ycola == 16) {
2294         ReleaseSemaphore(semaforos[240], 1, NULL);
2295     }
2296     if (xcola == 21 && ycola == 16) {
2297         ReleaseSemaphore(semaforos[241], 1, NULL);
2298     }
2299     if (xcola == 22 && ycola == 16) {
2300         ReleaseSemaphore(semaforos[242], 1, NULL);
2301     }
2302     if (xcola == 23 && ycola == 16) {
2303         ReleaseSemaphore(semaforos[243], 1, NULL);
2304     }
2305     if (xcola == 24 && ycola == 16) {
2306         ReleaseSemaphore(semaforos[244], 1, NULL);
2307     }
2308     if (xcola == 25 && ycola == 16) {
2309         ReleaseSemaphore(semaforos[245], 1, NULL);
2310     }
2311     if (xcola == 26 && ycola == 16) {
2312         ReleaseSemaphore(semaforos[246], 1, NULL);
2313     }
2314     if (xcola == 27 && ycola == 16) {
2315         ReleaseSemaphore(semaforos[247], 1, NULL);
2316     }
2317     if (xcola == 28 && ycola == 16) {
2318         ReleaseSemaphore(semaforos[248], 1, NULL);
2319     }
2320     if (xcola == 29 && ycola == 16) {
2321         ReleaseSemaphore(semaforos[249], 1, NULL);
2322     }
2323     if (xcola == 30 && ycola == 16) {
2324         ReleaseSemaphore(semaforos[250], 1, NULL);
2325     }
2326     if (xcola == 31 && ycola == 16) {
```

```
2327         ReleaseSemaphore(semaforos[251], 1, NULL);
2328     }
2329     if (xcola == 32 && ycola == 16) {
2330         ReleaseSemaphore(semaforos[252], 1, NULL);
2331     }
2332     if (xcola == 33 && ycola == 16) {
2333         ReleaseSemaphore(semaforos[253], 1, NULL);
2334     }
2335     if (xcola == 34 && ycola == 16) {
2336         ReleaseSemaphore(semaforos[254], 1, NULL);
2337     }
2338     if (xcola == 35 && ycola == 16) {
2339         ReleaseSemaphore(semaforos[255], 1, NULL);
2340     }
2341     if (xcola == 37 && ycola == 16) {
2342         ReleaseSemaphore(semaforos[256], 1, NULL);
2343     }
2344     if (xcola == 38 && ycola == 16) {
2345         ReleaseSemaphore(semaforos[257], 1, NULL);
2346     }
2347     if (xcola == 39 && ycola == 16) {
2348         ReleaseSemaphore(semaforos[258], 1, NULL);
2349     }
2350     if (xcola == 40 && ycola == 16) {
2351         ReleaseSemaphore(semaforos[259], 1, NULL);
2352     }
2353     if (xcola == 41 && ycola == 16) {
2354         ReleaseSemaphore(semaforos[260], 1, NULL);
2355     }
2356     if (xcola == 42 && ycola == 16) {
2357         ReleaseSemaphore(semaforos[261], 1, NULL);
2358     }
2359     if (xcola == 43 && ycola == 16) {
2360         ReleaseSemaphore(semaforos[262], 1, NULL);
2361     }
2362     if (xcola == 44 && ycola == 16) {
2363         ReleaseSemaphore(semaforos[263], 1, NULL);
2364     }
2365     if (xcola == 45 && ycola == 16) {
2366         ReleaseSemaphore(semaforos[264], 1, NULL);
2367     }
2368     if (xcola == 46 && ycola == 16) {
2369         ReleaseSemaphore(semaforos[265], 1, NULL);
2370     }
2371     if (xcola == 47 && ycola == 16) {
2372         ReleaseSemaphore(semaforos[266], 1, NULL);
2373     }
2374     if (xcola == 48 && ycola == 16) {
2375         ReleaseSemaphore(semaforos[267], 1, NULL);
2376     }
2377     if (xcola == 49 && ycola == 16) {
2378         ReleaseSemaphore(semaforos[268], 1, NULL);
2379     }
2380     if (xcola == 50 && ycola == 16) {
2381         ReleaseSemaphore(semaforos[269], 1, NULL);
2382     }
2383     if (xcola == 51 && ycola == 16) {
2384         ReleaseSemaphore(semaforos[270], 1, NULL);
2385     }
2386     if (xcola == 52 && ycola == 16) {
```

```
2387         ReleaseSemaphore(semaforos[271], 1, NULL);
2388     }
2389     if (xcola == 53 && ycola == 16) {
2390         ReleaseSemaphore(semaforos[272], 1, NULL);
2391     }
2392     if (xcola == 55 && ycola == 16) {
2393         ReleaseSemaphore(semaforos[273], 1, NULL);
2394     }
2395     if (xcola == 56 && ycola == 16) {
2396         ReleaseSemaphore(semaforos[274], 1, NULL);
2397     }
2398     if (xcola == 57 && ycola == 16) {
2399         ReleaseSemaphore(semaforos[275], 1, NULL);
2400     }
2401     if (xcola == 58 && ycola == 16) {
2402         ReleaseSemaphore(semaforos[276], 1, NULL);
2403     }
2404     if (xcola == 59 && ycola == 16) {
2405         ReleaseSemaphore(semaforos[277], 1, NULL);
2406     }
2407     if (xcola == 60 && ycola == 16) {
2408         ReleaseSemaphore(semaforos[278], 1, NULL);
2409     }
2410     if (xcola == 61 && ycola == 16) {
2411         ReleaseSemaphore(semaforos[279], 1, NULL);
2412     }
2413     if (xcola == 62 && ycola == 16) {
2414         ReleaseSemaphore(semaforos[280], 1, NULL);
2415     }
2416     if (xcola == 63 && ycola == 16) {
2417         ReleaseSemaphore(semaforos[281], 1, NULL);
2418     }
2419     if (xcola == 64 && ycola == 16) {
2420         ReleaseSemaphore(semaforos[282], 1, NULL);
2421     }
2422     if (xcola == 65 && ycola == 16) {
2423         ReleaseSemaphore(semaforos[283], 1, NULL);
2424     }
2425     if (xcola == 66 && ycola == 16) {
2426         ReleaseSemaphore(semaforos[284], 1, NULL);
2427     }
2428     if (xcola == 67 && ycola == 16) {
2429         ReleaseSemaphore(semaforos[285], 1, NULL);
2430     }
2431     if (xcola == 69 && ycola == 16) {
2432         ReleaseSemaphore(semaforos[286], 1, NULL);
2433     }
2434     if (xcola == 70 && ycola == 16) {
2435         ReleaseSemaphore(semaforos[287], 1, NULL);
2436     }
2437     if (xcola == 71 && ycola == 16) {
2438         ReleaseSemaphore(semaforos[288], 1, NULL);
2439     }
2440     if (xcola == 72 && ycola == 16) {
2441         ReleaseSemaphore(semaforos[289], 1, NULL);
2442     }
2443     if (xcola == 73 && ycola == 16) {
2444         ReleaseSemaphore(semaforos[290], 1, NULL);
2445     }
2446 }
```

```
2447     //primer trayectoria vertical
2448     if (xcola == 0 && ycola == 4) {
2449         ReleaseSemaphore(semaforos[291], 1, NULL);
2450     }
2451     if (xcola == 0 && ycola == 5) {
2452         ReleaseSemaphore(semaforos[292], 1, NULL);
2453     }
2454     if (xcola == 0 && ycola == 6) {
2455         ReleaseSemaphore(semaforos[293], 1, NULL);
2456     }
2457     if (xcola == 0 && ycola == 7) {
2458         ReleaseSemaphore(semaforos[294], 1, NULL);
2459     }
2460     if (xcola == 0 && ycola == 8) {
2461         ReleaseSemaphore(semaforos[295], 1, NULL);
2462     }
2463     if (xcola == 0 && ycola == 10) {
2464         ReleaseSemaphore(semaforos[296], 1, NULL);
2465     }
2466     if (xcola == 0 && ycola == 11) {
2467         ReleaseSemaphore(semaforos[297], 1, NULL);
2468     }
2469     if (xcola == 0 && ycola == 13) {
2470         ReleaseSemaphore(semaforos[298], 1, NULL);
2471     }
2472     if (xcola == 0 && ycola == 14) {
2473         ReleaseSemaphore(semaforos[299], 1, NULL);
2474     }
2475     if (xcola == 0 && ycola == 15) {
2476         ReleaseSemaphore(semaforos[300], 1, NULL);
2477     }
2478     //segunda trayectoria vertical
2479     if (xcola == 16 && ycola == 1) {
2480         ReleaseSemaphore(semaforos[301], 1, NULL);
2481     }
2482     if (xcola == 16 && ycola == 2) {
2483         ReleaseSemaphore(semaforos[302], 1, NULL);
2484     }
2485     if (xcola == 16 && ycola == 3) {
2486         ReleaseSemaphore(semaforos[303], 1, NULL);
2487     }
2488     if (xcola == 16 && ycola == 5) {
2489         ReleaseSemaphore(semaforos[304], 1, NULL);
2490     }
2491     if (xcola == 16 && ycola == 6) {
2492         ReleaseSemaphore(semaforos[305], 1, NULL);
2493     }
2494     if (xcola == 16 && ycola == 8) {
2495         ReleaseSemaphore(semaforos[306], 1, NULL);
2496     }
2497     if (xcola == 16 && ycola == 10) {
2498         ReleaseSemaphore(semaforos[307], 1, NULL);
2499     }
2500     if (xcola == 16 && ycola == 11) {
2501         ReleaseSemaphore(semaforos[308], 1, NULL);
2502     }
2503     if (xcola == 16 && ycola == 13) {
2504         ReleaseSemaphore(semaforos[309], 1, NULL);
2505     }
2506     if (xcola == 16 && ycola == 14) {
```

```
2507         ReleaseSemaphore(semaforos[310], 1, NULL);
2508     }
2509     if (xcola == 16 && ycola == 15) {
2510         ReleaseSemaphore(semaforos[311], 1, NULL);
2511     }
2512     //tercera vertical
2513     if (xcola == 36 && ycola == 1) {
2514         ReleaseSemaphore(semaforos[312], 1, NULL);
2515     }
2516     if (xcola == 36 && ycola == 2) {
2517         ReleaseSemaphore(semaforos[313], 1, NULL);
2518     }
2519     if (xcola == 36 && ycola == 3) {
2520         ReleaseSemaphore(semaforos[314], 1, NULL);
2521     }
2522     if (xcola == 36 && ycola == 4) {
2523         ReleaseSemaphore(semaforos[388], 1, NULL);
2524     }
2525     if (xcola == 36 && ycola == 5) {
2526         ReleaseSemaphore(semaforos[315], 1, NULL);
2527     }
2528     if (xcola == 36 && ycola == 6) {
2529         ReleaseSemaphore(semaforos[316], 1, NULL);
2530     }
2531     if (xcola == 36 && ycola == 8) {
2532         ReleaseSemaphore(semaforos[317], 1, NULL);
2533     }
2534     if (xcola == 36 && ycola == 9) {
2535         ReleaseSemaphore(semaforos[318], 1, NULL);
2536     }
2537     if (xcola == 36 && ycola == 10) {
2538         ReleaseSemaphore(semaforos[319], 1, NULL);
2539     }
2540     if (xcola == 36 && ycola == 11) {
2541         ReleaseSemaphore(semaforos[320], 1, NULL);
2542     }
2543     if (xcola == 36 && ycola == 13) {
2544         ReleaseSemaphore(semaforos[321], 1, NULL);
2545     }
2546     if (xcola == 36 && ycola == 14) {
2547         ReleaseSemaphore(semaforos[322], 1, NULL);
2548     }
2549     if (xcola == 36 && ycola == 15) {
2550         ReleaseSemaphore(semaforos[323], 1, NULL);
2551     }
2552     //cuarta vertical
2553     if (xcola == 54 && ycola == 1) {
2554         ReleaseSemaphore(semaforos[324], 1, NULL);
2555     }
2556     if (xcola == 54 && ycola == 2) {
2557         ReleaseSemaphore(semaforos[325], 1, NULL);
2558     }
2559     if (xcola == 54 && ycola == 3) {
2560         ReleaseSemaphore(semaforos[326], 1, NULL);
2561     }
2562     if (xcola == 54 && ycola == 4) {
2563         ReleaseSemaphore(semaforos[389], 1, NULL);
2564     }
2565     if (xcola == 54 && ycola == 5) {
2566         ReleaseSemaphore(semaforos[327], 1, NULL);
```

```
2567 }
2568 if (xcola == 54 && ycola == 6) {
2569     ReleaseSemaphore(semaforos[328], 1, NULL);
2570 }
2571 if (xcola == 54 && ycola == 8) {
2572     ReleaseSemaphore(semaforos[329], 1, NULL);
2573 }
2574 if (xcola == 54 && ycola == 9) {
2575     ReleaseSemaphore(semaforos[330], 1, NULL);
2576 }
2577 if (xcola == 54 && ycola == 10) {
2578     ReleaseSemaphore(semaforos[331], 1, NULL);
2579 }
2580 if (xcola == 54 && ycola == 11) {
2581     ReleaseSemaphore(semaforos[332], 1, NULL);
2582 }
2583 if (xcola == 54 && ycola == 13) {
2584     ReleaseSemaphore(semaforos[333], 1, NULL);
2585 }
2586 if (xcola == 54 && ycola == 14) {
2587     ReleaseSemaphore(semaforos[334], 1, NULL);
2588 }
2589 if (xcola == 54 && ycola == 15) {
2590     ReleaseSemaphore(semaforos[335], 1, NULL);
2591 }
2592 //quinta vertical
2593 if (xcola == 68 && ycola == 1) {
2594     ReleaseSemaphore(semaforos[336], 1, NULL);
2595 }
2596 if (xcola == 68 && ycola == 2) {
2597     ReleaseSemaphore(semaforos[337], 1, NULL);
2598 }
2599 if (xcola == 68 && ycola == 3) {
2600     ReleaseSemaphore(semaforos[338], 1, NULL);
2601 }
2602 if (xcola == 68 && ycola == 4) {
2603     ReleaseSemaphore(semaforos[339], 1, NULL);
2604 }
2605 if (xcola == 68 && ycola == 5) {
2606     ReleaseSemaphore(semaforos[340], 1, NULL);
2607 }
2608 if (xcola == 68 && ycola == 6) {
2609     ReleaseSemaphore(semaforos[341], 1, NULL);
2610 }
2611 if (xcola == 68 && ycola == 8) {
2612     ReleaseSemaphore(semaforos[342], 1, NULL);
2613 }
2614 if (xcola == 68 && ycola == 9) {
2615     ReleaseSemaphore(semaforos[343], 1, NULL);
2616 }
2617 if (xcola == 68 && ycola == 10) {
2618     ReleaseSemaphore(semaforos[344], 1, NULL);
2619 }
2620 if (xcola == 68 && ycola == 11) {
2621     ReleaseSemaphore(semaforos[345], 1, NULL);
2622 }
2623 if (xcola == 68 && ycola == 13) {
2624     ReleaseSemaphore(semaforos[346], 1, NULL);
2625 }
2626 if (xcola == 68 && ycola == 14) {
```

```

2627         ReleaseSemaphore(semaforos[346], 1, NULL);
2628     }
2629     if (xcola == 68 && ycola == 15) {
2630         ReleaseSemaphore(semaforos[347], 1, NULL);
2631     }
2632     //ultima vertical
2633     if (xcola == 74 && ycola == 1) {
2634         ReleaseSemaphore(semaforos[348], 1, NULL);
2635     }
2636     if (xcola == 74 && ycola == 2) {
2637         ReleaseSemaphore(semaforos[349], 1, NULL);
2638     }
2639     if (xcola == 74 && ycola == 3) {
2640         ReleaseSemaphore(semaforos[350], 1, NULL);
2641     }
2642     if (xcola == 74 && ycola == 4) {
2643         ReleaseSemaphore(semaforos[391], 1, NULL);
2644     }
2645     if (xcola == 74 && ycola == 5) {
2646         ReleaseSemaphore(semaforos[351], 1, NULL);
2647     }
2648     if (xcola == 74 && ycola == 6) {
2649         ReleaseSemaphore(semaforos[352], 1, NULL);
2650     }
2651     if (xcola == 74 && ycola == 8) {
2652         ReleaseSemaphore(semaforos[353], 1, NULL);
2653     }
2654     if (xcola == 74 && ycola == 9) {
2655         ReleaseSemaphore(semaforos[354], 1, NULL);
2656     }
2657     if (xcola == 74 && ycola == 10) {
2658         ReleaseSemaphore(semaforos[355], 1, NULL);
2659     }
2660     if (xcola == 74 && ycola == 11) {
2661         ReleaseSemaphore(semaforos[356], 1, NULL);
2662     }
2663     if (xcola == 74 && ycola == 13) {
2664         ReleaseSemaphore(semaforos[357], 1, NULL);
2665     }
2666     if (xcola == 74 && ycola == 14) {
2667         ReleaseSemaphore(semaforos[358], 1, NULL);
2668     }
2669     if (xcola == 74 && ycola == 15) {
2670         ReleaseSemaphore(semaforos[359], 1, NULL);
2671     }

2673     //para evitar que entre aqui mientras esta saliendo el tren
2674     if (xcola != -1 && ycola != -1) { // sino se libera posicion,nos devuelve un
-1
2675         mapa[ycola][xcola] = 0;
2676     }

2679     LOMO_espera(coordsig - 1, coordsig);
2680     ReleaseSemaphore(semaforos[361], 1, NULL);
2681     WaitForSingleObject(semaforos[361], INFINITE);

2683     //TRAYECTORIA1
2684     if (mapa[4][0] == 1 && mapa[4][1] == 1 && mapa[4][2] == 1 && mapa[4][3] == 1
&& mapa[4][4] == 1 && mapa[4][5] == 1 && mapa[4][6] == 1 && mapa[4][7] == 1 &&

```

```

mapa[4][8] == 1 && mapa[4][9] == 1 && mapa[4][10] == 1 && mapa[4][11] == 1 &&
mapa[4][12] == 1 && mapa[4][13] == 1 && mapa[4][14] == 1 && mapa[4][15] == 1 &&
mapa[4][16] == 1) {
    if (mapa[9][0] == 1 && mapa[9][1] == 1 && mapa[9][2] == 1 && mapa[9][3]
== 1 && mapa[9][4] == 1 && mapa[9][5] == 1 && mapa[9][6] == 1 && mapa[9][7] == 1 &&
mapa[9][8] == 1 && mapa[9][9] == 1 && mapa[9][10] == 1 && mapa[9][11] == 1 &&
mapa[9][12] == 1 && mapa[9][13] == 1 && mapa[9][14] == 1 && mapa[9][15] == 1 &&
mapa[9][16] == 1) {
        if (mapa[5][0] == 1 && mapa[6][0] == 1 && mapa[7][0] == 1 && mapa[8]
[0] == 1) {
            if (mapa[5][16] == 1 && mapa[6][16] == 1 && mapa[7][16] == 1 &&
mapa[8][16] == 1) {
                pon_error("INTERBLOQUEO");
                terminarPrograma(1);
                return 100;
            }
        }
    }
}

//TRAYECTORIA2
if (mapa[9][0] == 1 && mapa[9][1] == 1 && mapa[9][2] == 1 && mapa[9][3] == 1
&& mapa[9][4] == 1 && mapa[9][5] == 1 && mapa[9][6] == 1 && mapa[9][7] == 1 &&
mapa[9][8] == 1 && mapa[9][9] == 1 && mapa[9][10] == 1 && mapa[9][11] == 1 &&
mapa[9][12] == 1 && mapa[9][13] == 1 && mapa[9][14] == 1 && mapa[9][15] == 1 &&
mapa[9][16] == 1) {
    if (mapa[12][0] == 1 && mapa[12][1] == 1 && mapa[12][2] == 1 && mapa[12]
[3] == 1 && mapa[12][4] == 1 && mapa[12][5] == 1 && mapa[12][6] == 1 && mapa[12][7]
== 1 && mapa[12][8] == 1 && mapa[12][9] == 1 && mapa[12][10] == 1 && mapa[12][11] ==
1 && mapa[12][12] == 1 && mapa[12][13] == 1 && mapa[12][14] == 1 && mapa[12][15] ==
1 && mapa[12][16] == 1) {
        if (mapa[10][0] == 1 && mapa[11][0] == 1) {
            if (mapa[10][16] == 1 && mapa[11][16] == 1) {
                pon_error("INTERBLOQUEO");
                terminarPrograma(1);
                return 100;
            }
        }
    }
}

//TRAYECTORIA3
if (mapa[12][0] == 1 && mapa[12][1] == 1 && mapa[12][2] == 1 && mapa[12][3]
== 1 && mapa[12][4] == 1 && mapa[12][5] == 1 && mapa[12][6] == 1 && mapa[12][7] == 1
&& mapa[12][8] == 1 && mapa[12][9] == 1 && mapa[12][10] == 1 && mapa[12][11] == 1 &&
mapa[12][12] == 1 && mapa[12][13] == 1 && mapa[12][14] == 1 && mapa[12][15] == 1 &&
mapa[12][16] == 1) {
    if (mapa[16][0] == 1 && mapa[16][1] == 1 && mapa[16][2] == 1 && mapa[16]
[3] == 1 && mapa[16][4] == 1 && mapa[16][5] == 1 && mapa[16][6] == 1 && mapa[16][7]
== 1 && mapa[16][8] == 1 && mapa[16][9] == 1 && mapa[16][10] == 1 && mapa[16][11] ==
1 && mapa[16][12] == 1 && mapa[16][13] == 1 && mapa[16][14] == 1 && mapa[16][15] ==
1 && mapa[16][16] == 1) {
        if (mapa[13][0] == 1 && mapa[14][0] == 1 && mapa[15][0] == 1) {
            if (mapa[13][16] == 1 && mapa[14][16] == 1 && mapa[15][16] == 1) {
{
                pon_error("INTERBLOQUEO");
                terminarPrograma(1);
                return 100;
            }
        }
    }
}

```

```

2719         }
2720     }
2721     //TRAYECTORIA1_2
2722     if (mapa[4][0] == 1 && mapa[4][1] == 1 && mapa[4][2] == 1 && mapa[4][3] == 1
&& mapa[4][4] == 1 && mapa[4][5] == 1 && mapa[4][6] == 1 && mapa[4][7] == 1 &&
mapa[4][8] == 1 && mapa[4][9] == 1 && mapa[4][10] == 1 && mapa[4][11] == 1 &&
mapa[4][12] == 1 && mapa[4][13] == 1 && mapa[4][14] == 1 && mapa[4][15] == 1 &&
mapa[4][16] == 1) {
2723         if (mapa[12][0] == 1 && mapa[12][1] == 1 && mapa[12][2] == 1 && mapa[12]
[3] == 1 && mapa[12][4] == 1 && mapa[12][5] == 1 && mapa[12][6] == 1 && mapa[12][7]
== 1 && mapa[12][8] == 1 && mapa[12][9] == 1 && mapa[12][10] == 1 && mapa[12][11] ==
1 && mapa[12][12] == 1 && mapa[12][13] == 1 && mapa[12][14] == 1 && mapa[12][15] ==
1 && mapa[12][16] == 1) {
2724             if (mapa[5][0] == 1 && mapa[6][0] == 1 && mapa[7][0] == 1 && mapa[8]
[0] == 1 && mapa[10][0] == 1 && mapa[11][0] == 1) {
2725                 if (mapa[5][16] == 1 && mapa[6][16] == 1 && mapa[7][16] == 1 &&
mapa[8][16] == 1 && mapa[10][16] == 1 && mapa[11][16] == 1) {
2726                     pon_error("INTERBLOQUEO");
2727                     terminarPrograma(1);
2728                     return 100;
2729                 }
2730             }
2731         }
2732     }
2733     //TRAYECTORIA2_3
2734     if (mapa[9][0] == 1 && mapa[9][1] == 1 && mapa[9][2] == 1 && mapa[9][3] == 1
&& mapa[9][4] == 1 && mapa[9][5] == 1 && mapa[9][6] == 1 && mapa[9][7] == 1 &&
mapa[9][8] == 1 && mapa[9][9] == 1 && mapa[9][10] == 1 && mapa[9][11] == 1 &&
mapa[9][12] == 1 && mapa[9][13] == 1 && mapa[9][14] == 1 && mapa[9][15] == 1 &&
mapa[9][16] == 1) {
2735         if (mapa[16][0] == 1 && mapa[16][1] == 1 && mapa[16][2] == 1 && mapa[16]
[3] == 1 && mapa[16][4] == 1 && mapa[16][5] == 1 && mapa[16][6] == 1 && mapa[16][7]
== 1 && mapa[16][8] == 1 && mapa[16][9] == 1 && mapa[16][10] == 1 && mapa[16][11] ==
1 && mapa[16][12] == 1 && mapa[16][13] == 1 && mapa[16][14] == 1 && mapa[16][15] ==
1 && mapa[16][16] == 1) {
2736             if (mapa[10][0] == 1 && mapa[11][0] == 1 && mapa[13][0] == 1 &&
mapa[14][0] == 1 && mapa[15][0] == 1) {
2737                 if (mapa[10][16] == 1 && mapa[11][16] == 1 && mapa[13][16] == 1
&& mapa[14][16] == 1 && mapa[15][16] == 1) {
2738                     pon_error("INTERBLOQUEO");
2739                     terminarPrograma(1);
2740                     return 100;
2741                 }
2742             }
2743         }
2744     }
2745     //TRAYECTORIA13
2746     if (mapa[0][68] == 1 && mapa[0][69] == 1 && mapa[0][70] == 1 && mapa[0][71]
== 1 && mapa[0][72] == 1 && mapa[0][73] == 1 && mapa[0][74] == 1 ) {
2747         if (mapa[1][68] == 1 && mapa[2][68] == 1 && mapa[3][68] == 1 && mapa[4]
[68] == 1 && mapa[5][68] == 1 && mapa[6][68] == 1 ) {
2748             if (mapa[7][68] == 1 && mapa[7][69] == 1 && mapa[7][70] == 1 &&
mapa[7][71] == 1 && mapa[7][72] == 1 && mapa[7][73] == 1 && mapa[7][74] == 1 ) {
2749                 if (mapa[1][74] == 1 && mapa[2][74] == 1 && mapa[3][74] == 1 &&
mapa[4][74] == 1 && mapa[5][74] == 1 && mapa[6][74] == 1 ) {
2750                     pon_error("INTERBLOQUEO");
2751                     terminarPrograma(1);
2752                     return 100;
2753                 }
2754             }

```

```

2755     }
2756 }
2757
2758 //TRAYECTORIA14
2759     if (mapa[7][68] == 1 && mapa[7][69] == 1 && mapa[7][70] == 1 && mapa[7][71]
2760 == 1 && mapa[7][72] == 1 && mapa[7][73] == 1 && mapa[7][74] == 1) {
2761         if (mapa[12][68] == 1 && mapa[12][69] == 1 && mapa[12][70] == 1 &&
2762 mapa[12][71] == 1 && mapa[12][72] == 1 && mapa[12][73] == 1 && mapa[12][74] == 1) {
2763             if (mapa[8][68] == 1 && mapa[9][69] == 1 && mapa[10][68] == 1 &&
2764 mapa[11][68] == 1) {
2765                 if (mapa[8][74] == 1 && mapa[9][74] == 1 && mapa[10][74] == 1 &&
2766 mapa[11][74] == 1) {
2767                     pon_error("INTERBLOQUEO");
2768                     terminarPrograma(1);
2769                     return 100;
2770                 }
2771             }
2772         }
2773     }
2774
2775 //TRAYECTORIA15
2776     if (mapa[12][68] == 1 && mapa[12][69] == 1 && mapa[12][70] == 1 && mapa[12]
2777 [71] == 1 && mapa[12][72] == 1 && mapa[12][73] == 1 && mapa[12][74] == 1) {
2778         if (mapa[16][68] == 1 && mapa[16][69] == 1 && mapa[16][70] == 1 &&
2779 mapa[16][71] == 1 && mapa[16][72] == 1 && mapa[16][73] == 1 && mapa[16][74] == 1) {
2780             if (mapa[13][68] == 1 && mapa[14][68] == 1 && mapa[15][68] == 1) {
2781                 if (mapa[13][74] == 1 && mapa[14][74] == 1 && mapa[15][74] == 1)
2782                 {
2783                     pon_error("INTERBLOQUEO");
2784                     terminarPrograma(1);
2785                     return 100;
2786                 }
2787             }
2788         }
2789     }
2790
2791 //TRAYECTORIA13_14
2792     if (mapa[0][68] == 1 && mapa[0][69] != 0 && mapa[0][70] != 0 && mapa[0][71]
2793 != 0 && mapa[0][72] != 0 && mapa[0][73] != 0 && mapa[0][74] != 0) {
2794         if (mapa[12][68] != 0 && mapa[12][69] != 0 && mapa[12][70] != 0 &&
2795 mapa[12][71] != 0 && mapa[12][72] == 1 && mapa[12][73] == 1 && mapa[12][74] == 1) {
2796             if (mapa[1][68] == 1 && mapa[2][68] == 1 && mapa[3][68] == 1 &&
2797 mapa[4][68] == 1 && mapa[5][68] == 1 && mapa[6][68] == 1 && mapa[7][68] == 1 &&
2798 mapa[8][68] == 1 && mapa[9][68] == 1 && mapa[10][68] == 1 && mapa[11][68] == 1 &&
2799 mapa[12][68] == 1) {
2800                 if (mapa[1][74] == 1 && mapa[2][74] == 1 && mapa[3][74] == 1 &&
2801 mapa[4][74] == 1 && mapa[5][74] == 1 && mapa[6][74] == 1 && mapa[7][74] == 1 &&
2802 mapa[8][74] == 1 && mapa[9][74] == 1 && mapa[10][74] == 1 && mapa[11][74] == 1 &&
2803 mapa[12][74] == 1) {
2804                     pon_error("INTERBLOQUEO");
2805                     terminarPrograma(1);
2806                     return 100;
2807                 }
2808             }
2809         }
2810     }
2811
2812 //TRAYECTORIA14_15
2813     if (mapa[7][68] == 1 && mapa[7][69] != 0 && mapa[7][70] != 0 && mapa[7][71]
2814 != 0 && mapa[7][72] != 0 && mapa[7][73] != 0 && mapa[7][74] != 0) {

```

```

2799     if (mapa[16][68] == 1 && mapa[16][69] != 0 && mapa[16][70] != 0 &&
mapa[16][71] != 0 && mapa[16][72] != 0 && mapa[16][73] != 0 && mapa[16][74] != 0) {
2800         if (mapa[7][68] == 1 && mapa[8][68] == 1 && mapa[9][68] == 1 &&
mapa[10][68] == 1 && mapa[11][68] == 1 && mapa[12][68] == 1 && mapa[13][68] == 1 &&
mapa[14][68] == 1 && mapa[15][68] == 1 && mapa[16][68] == 1) {
2801             if (mapa[7][74] == 1 && mapa[8][74] == 1 && mapa[9][74] == 1 &&
mapa[10][74] == 1 && mapa[11][74] == 1 && mapa[12][74] == 1 && mapa[13][74] == 1 &&
mapa[14][74] == 1 && mapa[15][74] == 1 && mapa[16][74] == 1) {
2802                 pon_error("INTERBLOQUEO");
2803                 terminarPrograma(1);
2804                 return 100;
2805             }
2806         }
2807     }
2808 }
2809 }
2810
2811
2812     ReleaseSemaphore(semaforos[361], 1, NULL);
2813 } //fin bucle while
2814
2815
2816 //Cada hijo hace un signal hasta desbloquear el recurso
2817 if (ReleaseSemaphore(sem_EsperarPadre, 1, NULL) == FALSE) {
2818     pon_error("Error signal EsperarPadre\n");
2819     terminarPrograma(1);
2820     return 100;
2821 }
2822
2823 }
2824
2825 void terminarPrograma(int signal) {
2826
2827     if (GetCurrentThread() == trenPadre) {
2828         if (LOMO_fin() == -1) {
2829             fprintf(stderr, "Error al finalizar la practica\n");
2830             exit(1);
2831         }
2832         system("cls");
2833
2834
2835     //liberamos libreria
2836     if (FreeLibrary(libreria) == FALSE) {
2837         pon_error("ERROR liberacion biblioteca\n");
2838     }
2839
2840     //liberamos los handle de los trenes
2841     for (int i = 0; i < nTrenes; i++) {
2842         if (CloseHandle(vectorTrenes[i]) == FALSE) {
2843             fprintf(stderr, "Error, no se pudo cerrar vectorTrenes\n");
2844             exit(1);
2845         }
2846     }
2847
2848     if (sem_EsperarPadre != NULL) {
2849         if (CloseHandle(sem_EsperarPadre) == FALSE) {
2850             fprintf(stderr, "Error, no se pudo cerrar semEsperarPadre\n");
2851             exit(1);
2852         }
2853     }

```

```
2854
2855 //liberamos semaforos de las coordenadas
2856 for (int i = 0; i < 392; i++) {
2857     if (CloseHandle(semaforos[i]) == FALSE) {
2858         fprintf(stderr, "Error, no se pudo cerrar semaforos\n");
2859         exit(1);
2860     }
2861 }
2862
2863
2864
2865 if (CloseHandle(trenPadre) == FALSE) {
2866     fprintf(stderr, "Error, no se pudo cerrar trenPadre\n");
2867     exit(1);
2868 }
2869
2870     fprintf(stderr, "La practica ha finalizado correctamente\n");
2871     exit(0);
2872 }
2873 }
2874 else {
2875     exit(0);
2876 }
2877
2878 }
2879
2880 //control c
2881 BOOL WINAPI CtrlHandler(DWORD fdwCtrlType) {
2882     switch (fdwCtrlType) {
2883     case CTRL_C_EVENT:
2884         terminarPrograma(1);
2885         return TRUE;
2886     default:
2887         return FALSE;
2888     }
2889 }
2890 }
```