|  |
| --- |
| 得分 |
|  |

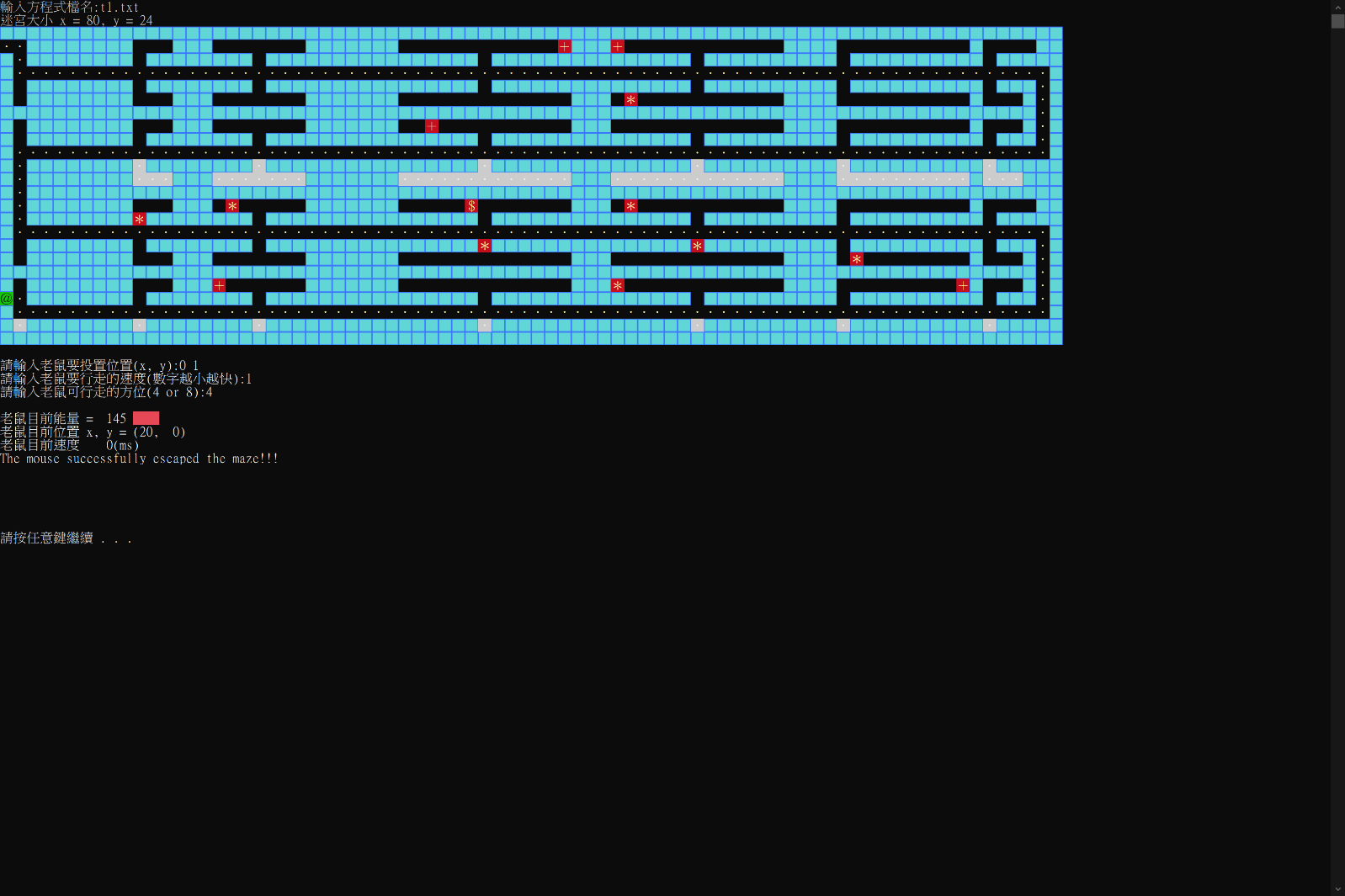
HWK4

老鼠走迷宮

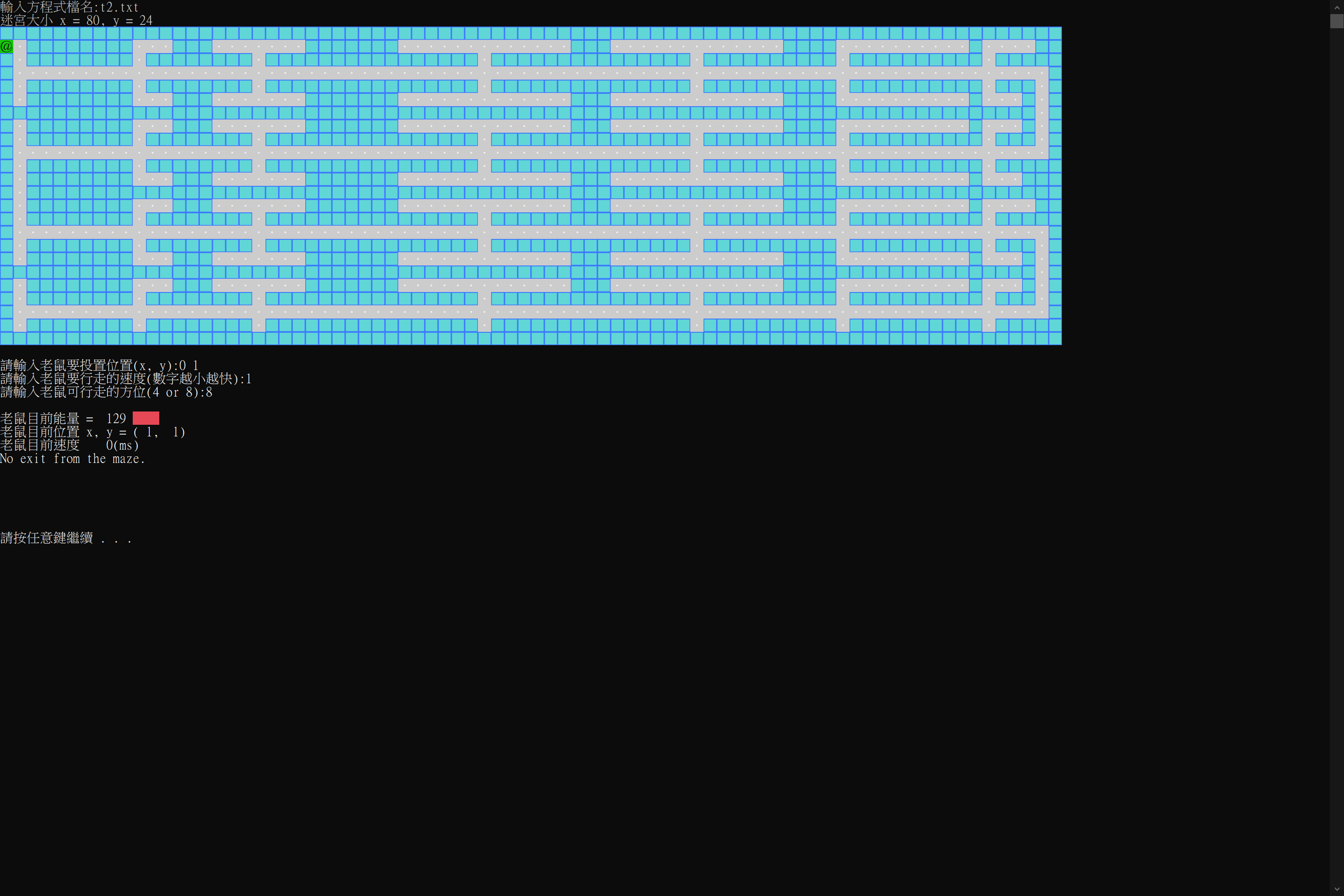
B10702057 盧昱達

1. #include <iostream>
2. #include <fstream>
3. #include <windows.h>
4. #include <dos.h>
5. #include <conio.h>
6. #define x\_max 200
7. #define y\_max 200
8. **using** **namespace** std;
10. **typedef** **struct**{
11. **int** type;
12. }note;
14. **typedef** **struct**{
15. **int** x, y;
16. }point;
18. point nextpoint(point now, note a[][y\_max]);
19. **void** clearline(**int** n);
20. **void** pause(**int** n);
21. **void** gotoxy(**int** x, **int** y);
22. **void** ConsoleFullScreen();
23. **void** setColor(**int** color);
24. **void** char\_to\_notearray(note a[][y\_max], **char** in[], **int** n, **int** \*x, **int** \*y);
25. **void** printSwitch(**int** a);
26. **void** print\_map(note a[][y\_max], **int** x, **int** y);
27. **void** foodpoint(point p, note a[][y\_max]);
28. **void** mouse\_stack(note a[][y\_max], **int** x, **int** y, **int** gox, **int** goy);
29. **void** print\_mouse\_point(**int** x);
30. **void** runtime();
31. **bool** CheakAll(point p, note a);
32. **bool** not1or3(point p, note a[][y\_max]);
33. **int** InDate(**char** in[]);
35. //全域變數
36. **int** mouse\_point = 500;
37. **int** op = 1;
38. **int** op4or8 = 1;
39. **int** time = 10;
40. **int** timeop = 10;
42. **int** main ( **void** )
43. {
44. **int** t;
45. **int** x = 0, y = 0, n = 0;
46. **int** gox, goy;
47. **char** in[x\_max \* y\_max];
48. note a[x\_max][y\_max];
50. ConsoleFullScreen();
52. //讀入迷宮地圖
53. n = InDate(in);
54. char\_to\_notearray(a, in, n, &x, &y);
55. //列印地圖
56. print\_map(a, x, y);
58. //輸入老鼠投放位置
59. **bool** run = **true**;
60. cout << "\n請輸入老鼠要投置位置(x, y):";
61. **do**
62. {
63. cin >> goy >> gox;
64. **if**(a[gox][goy].type == '1') clearline(x + 3), gotoxy(x + 3, 0), cout << "牆壁無法投放，再輸入一次。";
65. **else** **if**(gox >= x || goy >= y || (gox >= x && goy >= y)) clearline(x + 3), gotoxy(x + 3, 0), cout << "超出範圍，再輸入一次。";
66. **else** run = **false**;
67. }
68. **while**(run);
70. cout << "請輸入老鼠要行走的速度(數字越小越快):";
71. **do**
72. {
73. cin >> time;
74. **if**(time <= 0) clearline(x + 4), gotoxy(x + 4, 0), cout << "時間輸入錯誤";
75. }
76. **while**(time <= 0);
77. timeop = time;
79. cout << "請輸入老鼠可行走的方位(4 or 8):";
80. **do**
81. {
82. cin >> t;
83. **if** (t == 4 || t == 8) op4or8 = t;
84. **else** clearline(x + 5), gotoxy(x + 5, 0), cout << "方位輸入錯誤";
85. }
86. **while**(t != 8 && t != 4);
88. //老鼠投放至起始位置
89. gotoxy(2 + gox, goy \* 2);
90. printSwitch('@');
91. print\_mouse\_point(x);
92. pause(x + 16);
94. //老鼠開跑囉!!!!
95. mouse\_stack(a, x, y, gox, goy);
97. //結束!!
98. pause(x + 16);
99. }
101. **void** clearline(**int** n)
102. {
103. gotoxy(n, 0);
104. cout << "                                                                                   ";
105. }
107. **void** ConsoleFullScreen()
108. {
109. keybd\_event(VK\_MENU,0x38,0,0);
110. keybd\_event(VK\_RETURN,0x1c,0,0);
111. keybd\_event(VK\_MENU,0xb8,KEYEVENTF\_KEYUP,0);
112. keybd\_event(VK\_RETURN,0x9c,KEYEVENTF\_KEYUP,0);
113. }
115. **void** pause(**int** n)
116. {
117. gotoxy(n, 0);
118. system("pause");
119. gotoxy(n, 0);
120. cout << "                                         ";
121. }
123. **void** gotoxy(**int** x, **int** y)
124. {
125. **static** **HANDLE** o = GetStdHandle (STD\_OUTPUT\_HANDLE);
126. COORD c = {y, x};
127. SetConsoleCursorPosition (o, c);
128. }
130. **void** setColor(**int** color)
131. {
132. **HANDLE** hConsole;
133. hConsole = GetStdHandle (STD\_OUTPUT\_HANDLE);
134. SetConsoleTextAttribute(hConsole, color);
135. }
137. **int** InDate(**char** in[])
138. {
139. fstream InF;
140. **int** n = 0;
141. **char** FName[20], ch;
142. cout << "輸入方程式檔名:";
143. cin >> FName;
144. InF.open(FName, ios::in);
145. **if**(!InF)
146. {
147. cout << "檔案無法開啟\n";
148. exit(1);
149. }
150. **else**
151. {
152. **while**(InF.get(ch))
153. {
154. in[n] = ch;
155. n++;
156. }
157. InF.close();
158. }
159. **return** n;
160. }
162. **void** char\_to\_notearray(note a[][y\_max], **char** in[], **int** n, **int** \*x, **int** \*y)
163. {
164. **int** xx = 0, yy = 0, maxY = 0;
165. **for** ( **int** i = 0; i < n; i++)
166. {
167. **if** (in[i] == '\n')
168. {
169. xx += 1;
170. yy = 0;
171. }
172. **else**
173. {
174. a[xx][yy++].type = in[i];
175. **if**(yy > maxY) maxY = yy;
176. }
177. }
178. **if** (in[n - 1] != '\n') xx++;
179. \*x = xx;
180. \*y = maxY;
181. }
183. **void** printSwitch(**int** a)
184. {
185. /\*
186. ## 0 = 未走過的路                -- 1 = 牆壁
187. ## 2 = 走過正確的路           -- 3 = 走過錯誤的路
188. ## + = 體力+50的食物             -- \* = 體力+100的食物
189. ## $ = 體力+200的食物            -- # = 出口
190. ## + == 43, \* == 42, $ == 36
191. \*/
192. **switch**(a)
193. {
194. **case** '0':  //未走過的路
195. **case** 8:    //沒能量
196. setColor(15);
197. cout << "　";
198. **break**;
199. **case** '1':  //牆壁
200. setColor(155);
201. cout << "■";
202. setColor(15);
203. **break**;
204. **case** 9:    //有能量
205. setColor(200);
206. cout << "　";
207. setColor(15);
208. **break**;
209. **case** '2':  //走過錯誤的路
210. setColor(127);
211. cout << "．";
212. setColor(15);
213. **break**;
214. **case** '3':  //走過正確的路
215. setColor(14);
216. cout << "．";
217. setColor(15);
218. **break**;
219. **case** '+':  //體力+50的食物
220. setColor(78);
221. cout << "＋";
222. setColor(15);
223. **break**;
224. **case** '\*':  //體力+80的食物
225. setColor(78);
226. cout << "＊";
227. setColor(15);
228. **break**;
229. **case** '$':  //體力+100的食物
230. setColor(78);
231. cout << "＄";
232. setColor(15);
233. **break**;
234. **case** '#':  //出口
235. setColor(117);
236. cout << "＃";
237. setColor(15);
238. **break**;
239. **case** '\n': //換行切換
240. setColor(15);
241. cout << "　";
242. cout << "\n";
243. **break**;
244. **case** '@':  //老鼠
245. setColor(160);
246. cout << "＠";
247. setColor(15);
248. **break**;
249. }
250. }
252. **void** print\_map(note a[][y\_max], **int** x, **int** y)
253. {
254. cout << "迷宮大小 x = " << y << ", y = " << x << endl;
255. **for** (**int** i = 0; i < x; i++)
256. {
257. **for** (**int** l = 0; l < y; l++)
258. {
259. printSwitch(a[i][l].type);
260. }
261. setColor(15);
262. printSwitch('\n');
263. }
264. }
266. **void** print\_mouse\_point(**int** x)
267. {
268. gotoxy(7 + x, 0);
269. **if** (mouse\_point > 0)
270. {
271. printf("老鼠目前能量 = %4d ", mouse\_point);
272. **int** hundreds, tens, n\_total;
273. hundreds = mouse\_point / 100;
274. tens = mouse\_point % 100;
275. **if** (tens > 50) n\_total = hundreds \* 2 + 1;
276. **else** n\_total = hundreds \* 2;
277. **for** (**int** i = 0; i < n\_total; i++)  printSwitch(9);
278. **for** (**int** i = 0; i < 11 - n\_total; i++) printSwitch(8);
279. }
280. **else**
281. {
282. cout << "                                              ";
283. cout << "老鼠能量用盡死亡!!!";
284. }
285. }
287. **bool** not1or3(point p, note a[][y\_max])
288. {
289. **int** k = a[p.x][p.y].type;
290. **if**(k == '0' || k == '+' || k == '\*' || k == '$' || k == '#') **return** **true**;
291. **else** **return** **false**;
292. }
294. **void** foodpoint(point p, note a[][y\_max])
295. {
296. **switch**(a[p.x][p.y].type)
297. {
298. **case** '+':
299. mouse\_point += 50;
300. a[p.x][p.y].type = '3';
301. **break**;
302. **case** '\*':
303. mouse\_point += 80;
304. a[p.x][p.y].type = '3';
305. **break**;
306. **case** '$':
307. mouse\_point += 100;
308. a[p.x][p.y].type = '3';
309. **break**;
310. **case** '#':
311. op = 0;
312. **break**;
313. **default**:
314. a[p.x][p.y].type = '3';
315. }
316. }
318. point nextpoint(point now, note a[][y\_max])
319. {
320. /\*
321. // 老鼠收尋順序  = 東 > 南 > 西 > 北
322. // 老鼠先判斷是否有下一步路  bool can\_go\_road
323. // 如果下一步是食物導入  int food\_road return food\_point
324. // 如果往下個方向不能前進，給定 out(-1, -1)
325. \*/
326. point out, p;
328. **if**(not1or3(p = {now.x, now.y + 1}, a) == **true**)
329. {
330. out.x = now.x;
331. out.y = now.y + 1;
332. foodpoint(out, a);
333. }
334. **else** **if** (not1or3(p = {now.x + 1, now.y + 1}, a) == **true** && op4or8 == 8)
335. {
336. out.x = now.x + 1;
337. out.y = now.y + 1;
338. foodpoint(out, a);
339. }
340. **else** **if** (not1or3(p = {now.x + 1, now.y}, a) == **true**)
341. {
342. out.x = now.x + 1;
343. out.y = now.y;
344. foodpoint(out, a);
345. }
346. **else** **if** (not1or3(p = {now.x + 1, now.y - 1}, a) == **true** && op4or8 == 8)
347. {
348. out.x = now.x + 1;
349. out.y = now.y - 1;
350. foodpoint(out, a);
351. }
352. **else** **if** (not1or3(p = {now.x, now.y - 1}, a) == **true**)
353. {
354. out.x = now.x;
355. out.y = now.y - 1;
356. foodpoint(out, a);
357. }
358. **else** **if** (not1or3(p = {now.x - 1, now.y - 1}, a) == **true** && op4or8 == 8)
359. {
360. out.x = now.x - 1;
361. out.y = now.y - 1;
362. foodpoint(out, a);
363. }
364. **else** **if** (not1or3(p = {now.x - 1, now.y}, a) == **true**)
365. {
366. out.x = now.x - 1;
367. out.y = now.y;
368. foodpoint(out, a);
369. }
370. **else** **if** (not1or3(p = {now.x - 1, now.y + 1}, a) == **true** && op4or8 == 8)
371. {
372. out.x = now.x - 1;
373. out.y = now.y + 1;
374. foodpoint(out, a);
375. }
376. **else**
377. {
378. out.x = -1;
379. out.y = -1;
380. }
381. **return** out;
382. }
384. **void** runtime()
385. {
386. **switch**(mouse\_point)
387. {
388. **case** 1 ... 200:
389. time = timeop \* 0.5;
390. **break**;
391. **case** 201 ... 300:
392. time = timeop \* 0.75;
393. **break**;
394. **case** 301 ... 400:
395. time = timeop \* 0.8;
396. **break**;
397. **case** 401 ... 500:
398. time = timeop;
399. **break**;
400. **case** 501 ... 1000:
401. time = timeop \* 1.5;
402. **break**;
403. **default**:
404. time = timeop \* 0.3;
405. }
406. }
408. **bool** CheakAll(point p, note a[][y\_max])
409. {
410. **int** t;
411. t = a[p.x][p.y].type;
412. **if** (t == '0' || t == '+' || t == '\*' || t == '$') **return** **true**;
413. **else** **return** **false**;
414. }
416. **void** mouse\_stack(note a[][y\_max], **int** x, **int** y, **int** gox, **int** goy)
417. {
418. point next, stack\_p[1000];
419. **int** top = 0, Exit = 0;
420. stack\_p[top] = {gox, goy};
421. a[stack\_p[top].x][stack\_p[top].y].type = '3';
422. **while** (Exit == 0 && mouse\_point > 0)
423. {
424. **do**
425. {
426. print\_mouse\_point(x);
427. next = nextpoint(stack\_p[top], a);
428. //先讓上一步路重新顯示
429. gotoxy(2 + stack\_p[top].x, stack\_p[top].y \* 2);
430. printSwitch(a[stack\_p[top].x][stack\_p[top].y].type);
432. gotoxy(8 + x, 0);
433. printf("老鼠目前位置 x, y = (%2d, %2d)", stack\_p[top].x, stack\_p[top].y);
435. **if** (next.x != -1 && next.y != -1)
436. {
437. //老鼠成功找到下一步，再前往下一步。
439. **if** (a[stack\_p[top].x][stack\_p[top].y].type == '2' && op4or8 == 4)
440. {
441. a[stack\_p[top].x][stack\_p[top].y].type = '3';
442. gotoxy(2 + stack\_p[top].x, stack\_p[top].y \* 2);
443. printSwitch(a[stack\_p[top].x][stack\_p[top].y].type);
444. }
446. **if** (op4or8 == 8)
447. {
448. a[stack\_p[top].x][stack\_p[top].y].type = '3';
449. gotoxy(2 + stack\_p[top].x, stack\_p[top].y \* 2);
450. printSwitch(a[stack\_p[top].x][stack\_p[top].y].type);
451. }
453. top++;
454. stack\_p[top] = next;
456. gotoxy(2 + stack\_p[top].x, stack\_p[top].y \* 2);
457. printSwitch('@');
459. **if** (op == 0)
460. {
461. mouse\_point--;
462. Exit = 1;
463. **break**;
464. }
465. }
466. **else** **if**(next.x == -1 && next.y == -1 && top >= 0)
467. {
468. //老鼠找不到下一步路，退到上一步路，並讓目前這部重新顯示錯誤的路 。
469. a[stack\_p[top].x][stack\_p[top].y].type = '2';
470. gotoxy(2 + stack\_p[top].x, stack\_p[top].y \* 2);
471. printSwitch(a[stack\_p[top].x][stack\_p[top].y].type);
473. top--;
475. a[stack\_p[top].x][stack\_p[top].y].type = '2';
476. gotoxy(2 + stack\_p[top].x, stack\_p[top].y \* 2);
477. printSwitch('@');
478. }
479. **else**
480. {
481. mouse\_point--;
482. Exit = 1;
483. **break**;
484. }
485. runtime();
486. gotoxy(9 + x, 0);
487. printf("老鼠目前速度 %4d(ms)", time);
488. \_sleep(time);
489. mouse\_point--;
490. } **while**(mouse\_point > 0 && top > 0); //判斷老鼠能量
492. point q;
493. **int** k = 0;
494. **if** (top == 0)
495. {
496. **if** (CheakAll(q = {stack\_p[top].x    , stack\_p[top].y + 1}, a)) k = 1;
497. **if** (CheakAll(q = {stack\_p[top].x + 1, stack\_p[top].y    }, a)) k = 1;
498. **if** (CheakAll(q = {stack\_p[top].x    , stack\_p[top].y - 1}, a)) k = 1;
499. **if** (CheakAll(q = {stack\_p[top].x - 1, stack\_p[top].y    }, a)) k = 1;
500. }
501. **if** (top == 0 && op == 8)
502. {
503. **if** (CheakAll(q = {stack\_p[top].x + 1, stack\_p[top].y + 1}, a)) k = 1;
504. **if** (CheakAll(q = {stack\_p[top].x + 1, stack\_p[top].y - 1}, a)) k = 1;
505. **if** (CheakAll(q = {stack\_p[top].x - 1, stack\_p[top].y - 1}, a)) k = 1;
506. **if** (CheakAll(q = {stack\_p[top].x - 1, stack\_p[top].y + 1}, a)) k = 1;
507. }
508. **if** (top == 0 && k == 0) Exit = 1;
509. }
511. //最後結算顯示
512. **if** (op == 0 && mouse\_point != 0)
513. {
514. print\_mouse\_point(x);
515. gotoxy(2 + stack\_p[top-1].x, stack\_p[top-1].y \* 2);
516. printSwitch(a[stack\_p[top-1].x][stack\_p[top-1].y].type);
517. gotoxy(8 + x, 0);
518. printf("老鼠目前位置 x, y = (%2d, %2d)\n", stack\_p[top].x, stack\_p[top].y);
519. gotoxy(10 + x, 0);
520. cout << "The mouse successfully escaped the maze!!!";
521. }
522. **if** (mouse\_point == 0)
523. {
524. print\_mouse\_point(x);
525. }
526. **if** (top <= 0)
527. {
528. gotoxy(10 + x, 0);
529. cout << "No exit from the maze.";
530. }
531. }

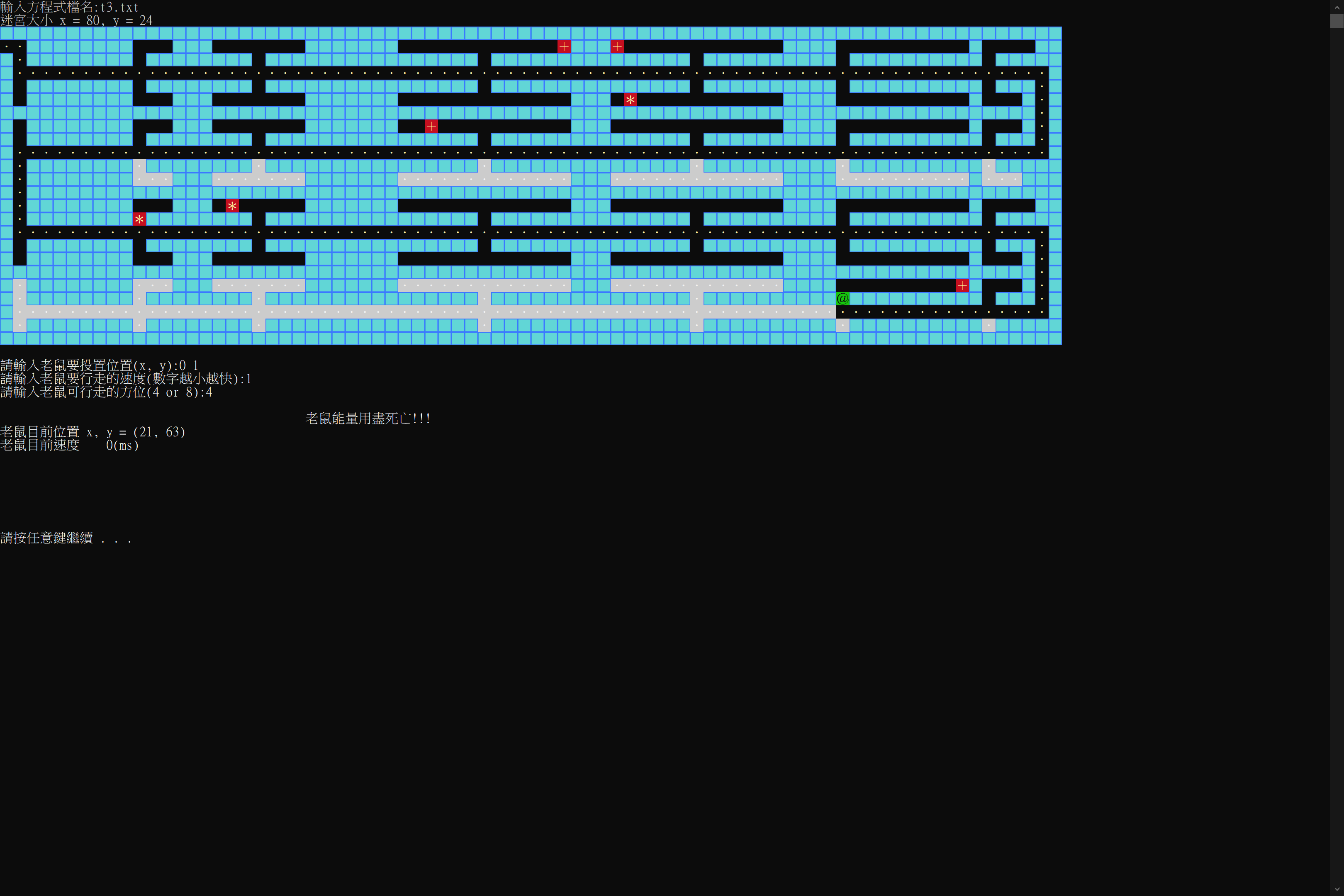
第一類 老鼠成功逃出



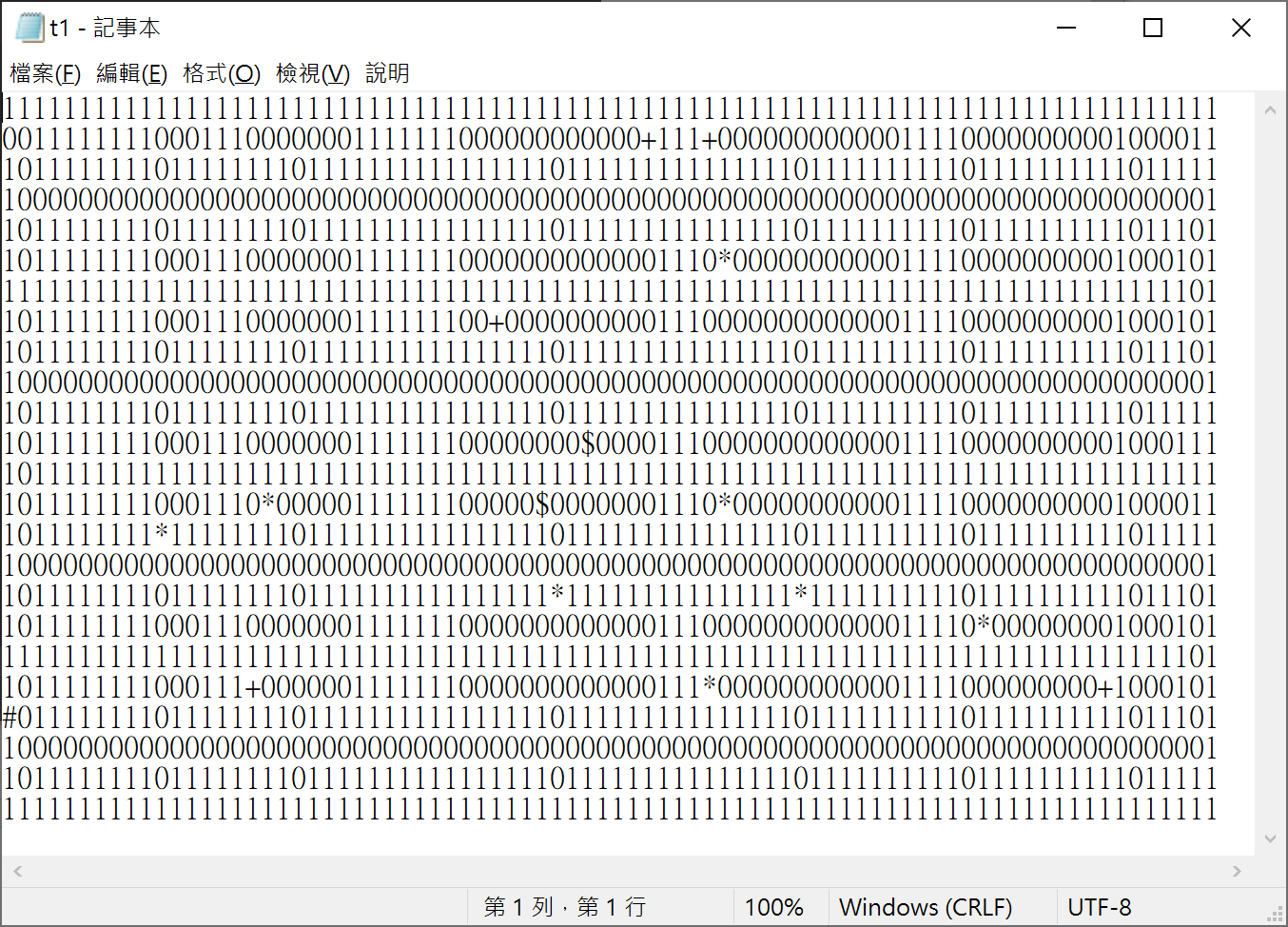
第二類 迷宮無出口



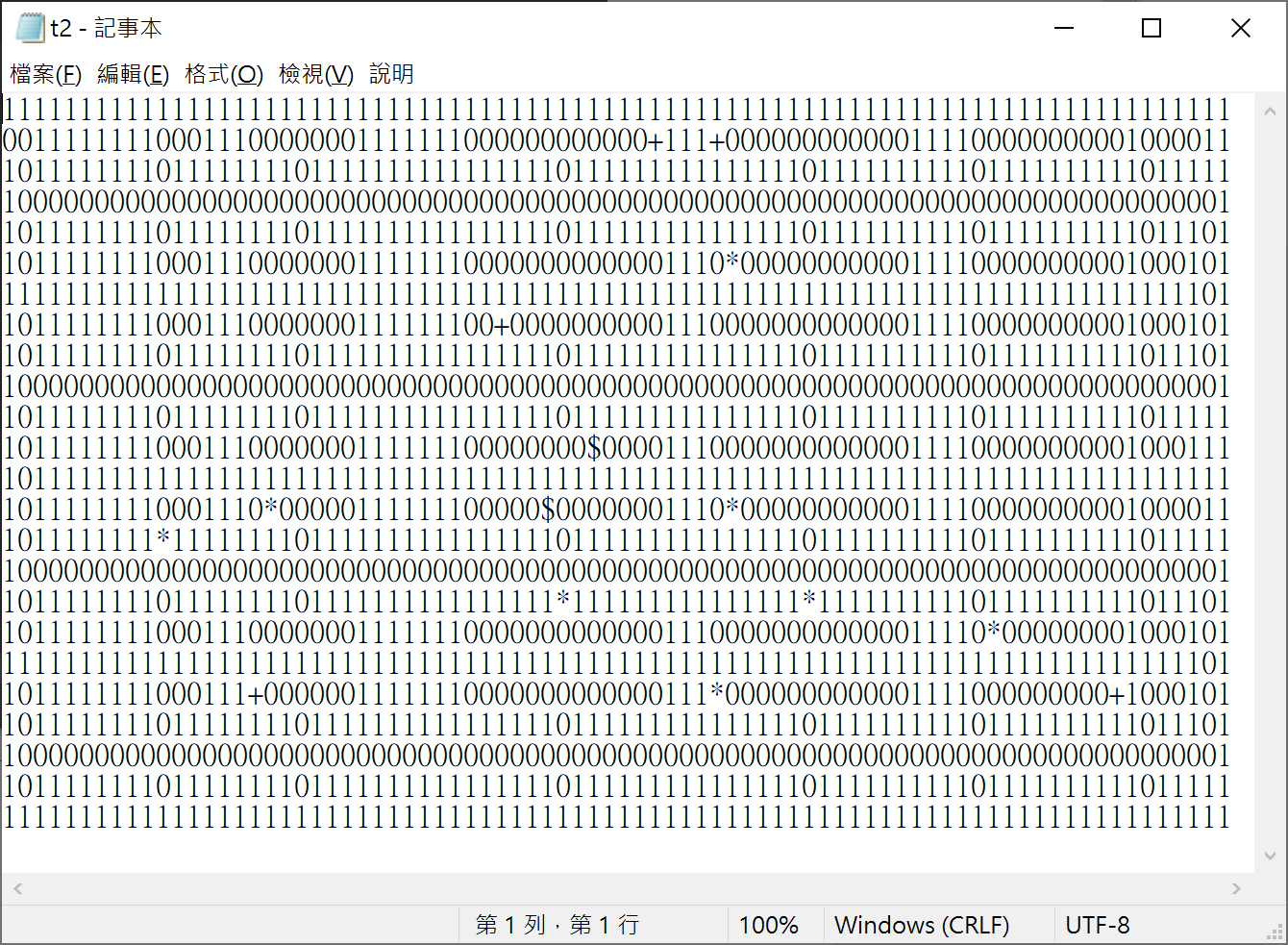
第三類 老鼠餓死



t1



t2



t3



心得

這次的功課對我來說是一種挑戰，首先是要用指定方式完成，必須跳脫以往寫程式的思維方式，便須花更多時間準備，每天打開一下，都會看到須修正的小缺陷，更正所有缺陷讓他達到更加完美，有時是讓冗長的程式縮減，有時則是加些新概念，這作業耗時將近兩個月，在之中真的學習到很多新東西，如還有時間，我期望把她轉換成GUI來實作。