```cpp
#include <iostream>
#include <fstream>
#include <windows.h>
#include <dos.h>
#include <conio.h>
#define x_max 100
#define y_max 100
using namespace std;

typedef struct{
    int type;
}note;

typedef struct{
    int x, y;
}point;

void gotoxy(int x, int y);
void setColor(int color);
void char_to_notearray(note a[][y_max], char in[], int n, int *x, int *y);
void printSwitch(int a);
void print_map(note a[][y_max], int x, int y);
void mouse_stack(note a[][y_max], int x, int y, int gox, int goy);
bool notlor3(point p, note a[][y_max]);
point nextpoint(point now, note a[][y_max]);
int foodpoint(point p, note a[][y_max]);
int InDate(char in[]);
int mouse_point = 500;
int op = 1;

int main ( void )
{
    int x = 0, y = 0, n = 0;
    int gox, goy;
    char in[x_max * y_max];
    note a[x_max][y_max];
    //讀入迷宮地圖
    n = InDate(in);
    char_to_notearray(a, in, n, &x, &y);
    //列印地圖
    print_map(a, x, y);

    //輸入老鼠投放位置
    cout << "\n請輸入老鼠要投置位置(x, y):";
    do
    {
        cin >> gox >> goy;
        if(a[gox][goy].type == '1') cout << "牆壁無法投放，再輸入一次。";
    }
    while(a[gox][goy].type == '1');

    //老鼠投放至起始位置
    gotoxy(2 + gox, goy * 2);
```

```
54      printSwitch('@');
55      gotoxy(10+x, 0);
56      system("pause");
57
58      //老鼠開跑囉!!!!
59      mouse_stack(a, x, y, gox, goy);
60
61      //結束!!
62      gotoxy(10+x, 0);
63      system("pause");
64  }
65
66  void gotoxy(int x, int y)
67  {
68      static HANDLE o = GetStdHandle (STD_OUTPUT_HANDLE);
69      COORD c = {y, x};
70      SetConsoleCursorPosition (o, c);
71  }
72
73  void setColor(int color)
74  {
75      HANDLE hConsole;
76      hConsole = GetStdHandle (STD_OUTPUT_HANDLE);
77      SetConsoleTextAttribute(hConsole, color);
78  }
79
80  int InDate(char in[])
81  {
82      fstream InF;
83      int n = 0;
84      char FName[20], ch;
85      cout << "輸入方程式檔名:";
86      cin >> FName;
87      InF.open(FName, ios::in);
88      if(!InF)
89      {
90          cout << "檔案無法開啟\n";
91      }
92      else
93      {
94          while(InF.get(ch))
95          {
96              in[n] = ch;
97              n++;
98          }
99          InF.close();
100     }
101     return n;
102 }
103
104 void char_to_notearray(note a[][y_max], char in[], int n, int *x, int *y)
105 {
```

```
106     int xx = 0, yy = 0, maxY = 0;
107     for ( int i = 0; i < n; i++)
108     {
109         if (in[i] == '\n')
110         {
111             xx += 1;
112             yy = 0;
113         }
114         else
115         {
116             a[xx][yy++].type = in[i];
117             if(yy > maxY) maxY = yy;
118         }
119     }
120     *x = xx;
121     *y = maxY;
122 }
123
124 void printSwitch(int a)
125 {
126     /*
127     ## 0 = 未走過的路                -- 1 = 牆壁
128     ## 2 = 走過正確的路              -- 3 = 走過錯誤的路
129     ## + = 體力+50的食物              -- * = 體力+100的食物
130     ## $ = 體力+200的食物             -- # = 出口
131     ## + == 43, * == 42, $ == 36
132     */
133     switch(a)
134     {
135         case '0'://未走過的路
136             setColor(15);
137             cout << "  ";
138             break;
139         case '1'://牆壁
140             setColor(155);
141             cout << "  ";
142             setColor(15);
143             break;
144         case '2'://走過錯誤的路
145             setColor(127);
146             cout << "・";
147             setColor(15);
148             break;
149         case '3'://走過正確的路
150             setColor(14);
151             cout << "・";
152             setColor(15);
153             break;
154         case '+'://體力+50的食物
155             setColor(78);
156             cout << "＋";
157             setColor(15);
158             break;
```

```
159         case '*'://體力+100的食物
160             setColor(78);
161             cout << "＊";
162             setColor(15);
163             break;
164         case '$'://體力+200的食物
165             setColor(78);
166             cout << "＄";
167             setColor(15);
168             break;
169         case '#'://出口
170             setColor(117);
171             cout << "＃";
172             setColor(15);
173             break;
174         case '\n'://換行切換
175             setColor(15);
176             cout << "  ";
177             cout << "\n";
178             break;
179         case '@'://老鼠
180             setColor(160);
181             cout << "@";
182             setColor(15);
183             break;
184     }
185 }
186
187 void print_map(note a[][y_max], int x, int y)
188 {
189     cout << "x = " << x;
190     cout << "y = " << y << endl;
191     for (int i = 0; i < x; i++)
192     {
193         for (int l = 0; l < y; l++)
194         {
195             printSwitch(a[i][l].type);
196         }
197         setColor(15);
198         printSwitch('\n');
199     }
200 }
201
202 void print_mouse_point(int x)
203 {
204     gotoxy(5 + x, 0);
205     if (mouse_point > 0) printf("老鼠目前能量 = %4d", mouse_point);
206     else cout << "老鼠能量用盡死亡!!!!!!!!!!!!";
207 }
208
209 bool notlor3(point p, note a[][y_max])
210 {
211     if(a[p.x][p.y].type == '1' || a[p.x][p.y].type == '2' || a[p.x][p.y].type == '3')
```

```
        return false;
212     else return true;
213 }
214
215 int foodpoint(point p, note a[][y_max])
216 {
217     switch(a[p.x][p.y].type)
218     {
219         case '+':
220             mouse_point += 50;
221             a[p.x][p.y].type = '3';
222             return 1;
223             break;
224         case '*':
225             mouse_point += 100;
226             a[p.x][p.y].type = '3';
227             return 1;
228             break;
229         case '$':
230             mouse_point += 200;
231             a[p.x][p.y].type = '3';
232             return 1;
233             break;
234         case '#':
235             op = 0;
236             return 2;
237             break;
238         default:
239             a[p.x][p.y].type = '3';
240             return 0;
241     }
242 }
243
244 point nextpoint(point now, note a[][y_max])
245 {
246 /*
247 // 老鼠收尋順序  = 東 > 南 > 西 > 北
248 // 老鼠先判斷是否有下一步路  bool can_go_road
249 // 如果下一步是食物導入  int food_road return food_point
250 // 如果往下個方向不能前進，給定 out(-1, -1)
251 */
252     int op;
253     point out, p;
254
255     if(notlor3(p = {now.x, now.y + 1}, a) == true)
256     {
257         out.x = now.x;
258         out.y = now.y + 1;
259         foodpoint(out, a);
260     }
261     else if (notlor3(p = {now.x + 1, now.y + 1}, a) == true)
262     {
263         out.x = now.x + 1;
```

```
264             out.y = now.y + 1;
265             foodpoint(out, a);
266         }
267         else if (notlor3(p = {now.x + 1, now.y}, a) == true)
268         {
269             out.x = now.x + 1;
270             out.y = now.y;
271             foodpoint(out, a);
272         }
273         else if (notlor3(p = {now.x + 1, now.y - 1}, a) == true)
274         {
275             out.x = now.x + 1;
276             out.y = now.y - 1;
277             foodpoint(out, a);
278         }
279         else if (notlor3(p = {now.x, now.y - 1}, a) == true)
280         {
281             out.x = now.x;
282             out.y = now.y - 1;
283             foodpoint(out, a);
284         }
285         else if (notlor3(p = {now.x + 1, now.y - 1}, a) == true)
286         {
287             out.x = now.x + 1;
288             out.y = now.y - 1;
289             foodpoint(out, a);
290         }
291         else if (notlor3(p = {now.x - 1, now.y}, a) == true)
292         {
293             out.x = now.x - 1;
294             out.y = now.y;
295             foodpoint(out, a);
296         }
297         else if (notlor3(p = {now.x - 1, now.y - 1}, a) == true)
298         {
299             out.x = now.x - 1;
300             out.y = now.y - 1;
301             foodpoint(out, a);
302         }
303         else
304         {
305             out.x = -1;
306             out.y = -1;
307         }
308         return out;
309     }
310
311     void mouse_stack(note a[][y_max], int x, int y, int gox, int goy)
312     {
313         point stack_p[1000];
314         point next;
315         int top = 0;
316         stack_p[top].x = gox;
```

```
317     stack_p[top].y = goy;
318     a[stack_p[top].x][stack_p[top].y].type = '3';
319     print_mouse_point(x);
320
321     //判斷老鼠能量
322     while(mouse_point > 0)
323     {
324         print_mouse_point(x);
325         next = nextpoint(stack_p[top], a);
326         //先讓上一步路重新顯示
327         gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
328         printSwitch(a[stack_p[top].x][stack_p[top].y].type);
329
330         gotoxy(6 + x, 0);
331         printf("老鼠目前位置 x, y = (%2d, %2d)", stack_p[top].x, stack_p[top].y);
332
333         if (next.x != -1 && next.y != -1)
334         {
335             //老鼠成功找到下一步，再前往下一步。
336
337             top++;
338             stack_p[top].x = next.x;
339             stack_p[top].y = next.y;
340
341             gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
342             printSwitch('@');
343
344             if(op == 0)
345             {
346                 //gotoxy(42, 0);
347                 //cout << "win!!";
348                 break;
349             }
350         }
351         else if(next.x == -1 && next.y == -1 && top >= 0)
352         {
353             //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 。
354             a[stack_p[top].x][stack_p[top].y].type = '2';
355             gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
356             printSwitch(a[stack_p[top].x][stack_p[top].y].type);
357
358             top--;
359
360             a[stack_p[top].x][stack_p[top].y].type = '2';
361             gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
362             printSwitch('@');
363         }
364         else
365         {
366             break;
367         }
368         _sleep(10);
369         mouse_point--;
```

```
370     }
371
372     if(op == 0 && mouse_point != 0)
373     {
374         gotoxy(2 + stack_p[top].x, stack_p[top].y * 2);
375         printSwitch(a[stack_p[top].x][stack_p[top].y].type);
376         gotoxy(6 + x, 0);
377         printf("老鼠目前位置 x, y = (%2d, %2d)\n", stack_p[top].x, stack_p[top].y);
378         gotoxy(7 + x, 0);
379         cout << "The mouse successfully escaped the maze!!!";
380     }
381     else if (mouse_point == 0)
382     {
383         print_mouse_point(x);
384     }
385     else
386     {
387         gotoxy(7 + x, 0);
388         cout << "No exit from the maze.";
389     }
390 }
391
```