

迷路を表示するだけのコードの主な処理を算譜1に示す。print\_map()では、2重の繰り返し構文を用いることで、盤面を表現している。また、添字の変数にはrow,colとすることで、縦と横が混同しにくいようにした。それぞれの繰り返し条件は、0から始まり、MAP\_SIZE (5) になると処理をせず終わる。増加は1ずつである。列挙型のchipに対応した文字列を、2次元配列mapから参照する。図1に実行結果を示す。

|    |  |
|----|--|
| 1  | void print_map(){                          |
| 2  | int col,row;                               |
| 3  | for(row = 0;row < MAP_SIZE;row++){         |
| 4  | for(col = 0;col < MAP_SIZE;col++){         |
| 5  | printf("%s",chip_disp[map[row][col]]);     |
| 6  | }  |
| 7  | puts("");                                  |
| 8  | }  |
| 9  | }  |
| 10 |  |
| 11 | int main(void){                            |
| 12 | print_map();                               |
| 13 | int input;                                 |
| 14 | // input = get_input(INPUT_MIN,INPUT_MAX); |
| 15 | return 0;                                  |
| 16 | }  |

算譜1 迷路を表示するコード



図1 実行結果のキャプチャ