Homework 3 Maze Problem DUE: 4/22 09:00

문제: 주어진 matrix안에서, stack을 이용하여 길을 찾는 문제 (시작점과 도착점이 제공).

조건:

- 데이터: 주어진 데이터 6x6 matrix 이용. 시작점 (0,0), 도착점(5,5)은 고정되어있음.
- 출력(output): 1) PATH 2) Marked Matrix

알고리즘 (참조):

```
typedef struct {   int row;     int col;     int dir;   } element
typedef struct {   int vert;     int horiz;}   offsets;              // moving direction
offsets move[8];
move[0].vert=-1; move[0].horiz=0;     // N        move[1].vert=-1; move[1].horiz=1;     // NE
move[2].vert=0; move[2].horiz=1;      // E        move[3].vert=1; move[3].horiz=1;      // SE
move[4].vert=1; move[4].horiz=0;      // S        move[5].vert=1; move[5].horiz=-1;     // SW
move[6].vert=0; move[6].horiz=-1;     // W        move[7].vert=-1; move[7].horiz=-1;   // NW

int maze[6][6] = { 0,1,1,1,1,1,   1,0,1,1,1,1,   1,0,0,0,0,1,   1,1,0,1,1,1,   1,0,1,0,0,1,   1,1,1,1,1,0 };
int mark[6][6] = { 0,0,0,0,0,0,   0,0,0,0,0,0,   0,0,0,0,0,0,   0,0,0,0,0,0,   0,0,0,0,0,0,   0,0,0,0,0,0 };
```

```
1. Start at 0,0:    mark[0][0] =1;    stack[0].row=1; stack[0].col=1;    stack[0].dir=NORTH;

2. while (!stack_empty&& !found) {
     temp = POP(top);       row = temp.row;      col = temp.col;      dir = temp.dir; //

     while (dir<8 && !found) { //계속 이동가능하다면
          next_row = row + move[dir].vert;   next_col = col + move[dir].horiz;
          if(next_row == 5 && next_col == 5)   // reached exit point?
                found = 1;
          else if (!maze[next_row][next_col]   &&   !mark[next_row][next_col]) {   // new position
                    mark[next_row][next_col]=1;                              // and not been here before
                    temp.row = row; temp.col=col; temp.dir=++dir;     // store current position
                    PUSH(top, temp);                                        // 현재위치 저장
                    row=next_row;     col = next_col;     dir = NORTH;   }// new position
        else   dir++;    // position change     } }

3. Print maze
  1) Print Stack (The path)
  2) Print Maked maze
```

***** OUTPUT**

```
The Path is

row    col
 0      0
 1      1
 2      2
 2      3
 3      2
 4      3
 4      4
 5      5

Marked Matrix
 1 0 0 0 0 0
 0 1 0 0 0 0
 0 0 1 1 1 0
 0 0 1 0 0 0
 0 0 0 1 1 0
 0 0 0 0 0 1
```