//프로그램 코드

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#define MAX\_STACK\_SIZE 100

#define FALSE 0

#define TRUE 1

int row;

int col;

typedef struct {

short int vert;

short int horiz;

} offsets;

offsets move[8];

typedef struct {

short int e\_row;

short int e\_col;

short int dir;

}element;

element stack[MAX\_STACK\_SIZE];

int maze[MAX\_STACK\_SIZE][MAX\_STACK\_SIZE];

void push(element item,int \* top)

{ if (\* top >= MAX\_STACK\_SIZE-1)

printf("Stack is Full\n");

stack[++\*top] = item;

}

element pop(int \*top)

{ if(\*top == -1) printf("stack is empty.\n");

return stack[(\*top)--];

}

void path(int(\*maze)[MAX\_STACK\_SIZE])

{ int i,j,top,next\_row,next\_col,mark[MAX\_STACK\_SIZE][MAX\_STACK\_SIZE],p\_row,p\_col,dir=1,found=FALSE;

element position;

int EXIT\_ROW=row;

int EXIT\_COL=col;

for(i=0;i<row+2;i++)

{ for(j=0;j<col+2;j++)

{mark[i][j]=0;

}

}

mark[1][1]=1; top=0;

stack[0].e\_row=1; stack[0].e\_col=1; stack[0].dir=1;

while(top>-1 && !found)

{

position=pop(&top);

p\_row=position.e\_row; p\_col=position.e\_col; dir=position.dir;

while(dir<8 && !found)

{

next\_row=p\_row+move[dir-1].vert;

next\_col=p\_col+move[dir-1].horiz;

if(next\_row==EXIT\_ROW && next\_col==EXIT\_COL)

found=TRUE;

else if(!maze[next\_row][next\_col] && !mark[next\_row][next\_col])

{

mark[next\_row][next\_col]=1;

position.e\_row=p\_row; position.e\_col=p\_col; position.dir=++dir;

push(position,&top);

p\_row=next\_row; p\_col=next\_col; dir=0;

}

else ++dir;

}

}

if (found){

printf("\n\*\* The maze dose have a path \*\*\n");

printf("(%d, %d)",row, col);

printf("<-(%d, %d)", p\_row, p\_col);

for(i=top; i>=0; i--) {

printf("<-(%d, %d)",stack[i].e\_row,stack[i].e\_col);

}

printf("\n\n");

}

else printf("\n The maze does not have path\n");

}

int main() {

int i,j;

int N=100;

//FILE \* maze\_fp;

int maze[100][100];

FILE \* maze\_fp=NULL;

maze\_fp=fopen("miro.txt","rt");

if(maze\_fp==NULL) {

printf("실패-종료\n");

exit(1);

}

fscanf(maze\_fp,"%d",&row);

fscanf(maze\_fp,"%d",&col);

printf(" %d %d \n",row,col);

for(i = 1;i<=row;i++) {

for(j = 1;j<=col;j++) {

fscanf(maze\_fp,"%d",&maze[i][j]);

printf("%d",maze[i][j]);

}

printf("\n");

}

if(maze\_fp!=NULL)

fclose(maze\_fp);

for(i=0;i<=col+2;i++)

{

maze[0][i]=1;

maze[row+1][i]=1;

}

for(i=0;i<=row+2;i++)

{

maze[i][col+1]=1;

maze[i][0]=1;

}

printf("미로 출력\n");

for(i=0;i<=row+1;i++)

{ for(j=0;j<=col+1;j++)

printf("%2d",maze[i][j]);

printf("\n");

}

move[0].vert=-1;

move[0].horiz=0;

move[1].vert=-1;

move[1].horiz=1;

move[2].vert=0;

move[2].horiz=1;

move[3].vert=1;

move[3].horiz=1;

move[4].vert=1;

move[4].horiz=0;

move[5].vert=1;

move[5].horiz=-1;

move[6].vert=0;

move[6].horiz=-1;

move[7].vert=-1;

move[7].horiz=-1;

path(maze);

return 0;

}

출력결과:

