**자료구조 과제 4**

(미로찾기 프로그램 작성)

logo.gif

|  |  |
| --- | --- |
| 과목명 | 자료구조 |
| 담당교수 | 김계영 |
| 학과 | 소프트웨어학부 |
| 학년 | 3 |
| 학번 | 20152994 |
| 성명 | 이진영 |
| 제출일 | 2017.5.15 |



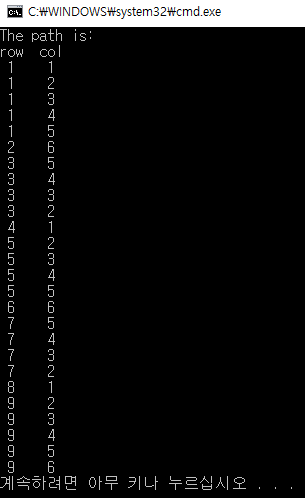
**과제4 - 미로찾기 프로그램 작성**

20152994 이진영

**1. 원시 코드**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define TRUE 1  #define FALSE 0  #define MAX\_STACK\_SIZE 100  typedef struct{  short int vert;  short int horiz;  } offsets;  offsets move[8]={{1,0},{1,1},{0,1},{-1,1},{-1,0},{-1,-1},{0,-1},{1,-1}};  typedef struct{  short int row;  short int col;  short int dir;  } element;  element stack[MAX\_STACK\_SIZE];  int top = -1;  int maze[11][8] = {{1,1,1,1,1,1,1,1},  {1,0,0,0,0,0,1,1},  {1,1,1,1,1,1,0,1},  {1,1,0,0,0,0,1,1},  {1,0,1,1,1,1,1,1},  {1,1,0,0,0,0,1,1},  {1,1,1,1,1,1,0,1},  {1,1,0,0,0,0,1,1},  {1,0,1,1,1,1,1,1},  {1,1,0,0,0,0,0,1},  {1,1,1,1,1,1,1,1}};  int EXIT\_ROW=9, EXIT\_COL=6;  int mark[11][8] = {0};  void push(element);  element pop();  void stackFull();  void stackEmpty();  void path(int\*,int,int);  void main(){  path((int\*)maze,11,8);  }  void push(element item)  {  if(top>=MAX\_STACK\_SIZE-1)  stackFull();  stack[++top] = item;  }  element pop()  {  if(top == -1)  stackEmpty();  return stack[top--];  }  void stackFull()  {  fprintf(stderr,"Stack is full, cannot add element");  exit(EXIT\_FAILURE);  }  void stackEmpty()  {  fprintf(stderr,"Stack is empty, cannot remove element");  exit(EXIT\_FAILURE);  }  void path(int\* maze, int row\_num, int col\_num)  {  int i, row, col, nextRow, nextCol, dir, found=FALSE;  element position;  position.row=1; position.col=1; position.dir=0;  mark[1][1]=1; top=0;  stack[0].row=1; stack[0].col=1; stack[0].dir=1;  while(top>-1 && !found){  position = pop();  row = position.row; col = position.col;  dir = position.dir;  while(dir<8 && !found){  nextRow = row + move[dir].vert;  nextCol = col + move[dir].horiz;  if(nextRow==EXIT\_ROW && nextCol==EXIT\_COL)  found = TRUE;  else if(!maze[(nextRow \* col\_num) + nextCol] && !mark[nextRow][nextCol]){  mark[nextRow][nextCol] = 1;  position.row = row; position.col = col;  position.dir = ++dir;  push(position);  row = nextRow; col = nextCol; dir = 0;  }  else ++dir;  }  }  if(found){  printf("The path is:\n");  printf("row col\n");  for(i=0; i<=top; i++)  printf("%2d%5d\n",stack[i].row,stack[i].col);  printf("%2d%5d\n",row,col);  printf("%2d%5d\n",EXIT\_ROW, EXIT\_COL);  }  else printf("The maze does not have a path\n");  } |

**2. 실행 결과**

****