

2015 자료구조 실습과제 7

1. 미로탐색 프로그램 완성

(1) 2차원 좌표를 나타내기 위한 클래스 작성

```
struct CPoint2D {  
    int    x;  
    int    y;  
    CPoint2D ( int xx=0, int yy=0 ) { x = xx; y = yy; }  
    ...  
};
```

(2) 스택 클래스 작성

- 2차원 좌표를 Push()하고 Pop() 하도록 함
- 배열을 이용한 스택을 사용해도 됨

```
typedef CPoint2D CElement;  
class CStack  
{  
    CElement m_data[MAX_STACK_SIZE]; // 요소의 배열  
    int      m_top;                  // 요소의 개수  
public:  
    CStack() { m_top = 0; }           // 스택 생성자 (ADT의 create()역할)  
    ~CStack(){}                       // 스택 소멸자  
    ...  
};
```

(3) 미로 클래스 완성

- SearchExit(): 스택을 이용하여 미로 찾기를 수행하는 함수
- Push(x,y): 좌표(x,y)를 검사하여 갈 수 있는 길이면 스택에 추가하는 함수

```
class CMaze  
{  
    int    m_w;           // 미로의 width  
    int    m_h;           // 미로의 height  
    int**  m_maze;         // 미로의 각 항목 값을 저장  
    CStack m_stack;        // 미로 탐색을 위한 스택  
    CElement m_start;      // 미로의 입구  
    CElement m_exit;       // 미로의 출구  
public:  
    CMaze() { Init( 0, 0 ); }  
    ~CMaze(){ Reset(); }  
    void Init(int w, int h) {...} // 이차원 배열을 동적으로 할당
```

```

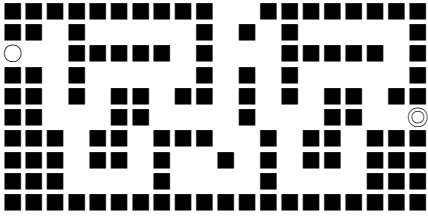
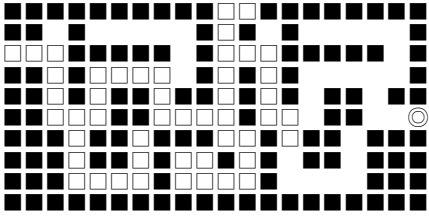
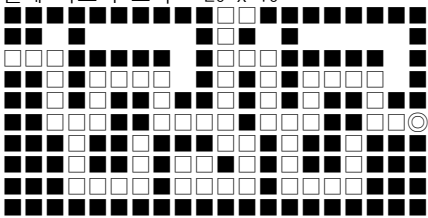
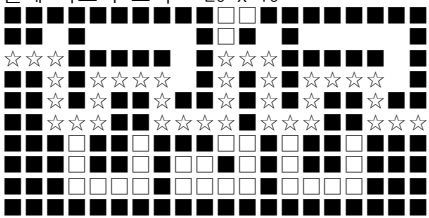
void Reset(int w, int h) {...} // 이차원 배열을 동적으로 해제
void Load( char *fname = NULL); // 파일/키보드 에서 Maze정보를 읽어옴
void Print(char *fname = NULL); // 현재 Maze를 파일/화면에 저장
void Push(int x, int y); // 현재 위치를 검사해서 스택에 추가
void SearchExit() ; // 미로찾기 시작
};

void main()
{
    CMaze maze;
    maze.Load("maze.txt");
    maze.Print();
    printf( " 입구○ --> 출구◎ 경로찾기 시작 [enter]\n");
    getchar();
    maze.searchExit();
    maze.Print();
}

```

(4) 자신만의 미로 파일을 만들어 테스트 함

- 최적경로 (다음 그림의 오른쪽 아래) 출력은 하지 않아도 됨

<p>전체 미로의 크기 = 20 x 10</p>  <p>입구○ --> 출구◎ 경로찾기 시작 [enter]</p>	<p>전체 미로의 크기 = 20 x 10</p>  <p>현재 상황: [13,6] == [19,5]</p>
<p>전체 미로의 크기 = 20 x 10</p>  <p>최적 경로 ☆ 출력 [enter]</p>	<p>전체 미로의 크기 = 20 x 10</p>  <p>종료 [enter]</p>

※ 과제의 양이 적은 학생들 (과제에 포함된 내용이 아님)

- 최적경로를 구하는 방법을 생각해 보고 구현해 볼 것
- 스택을 사용하지 않고 순환호출을 이용한 미로찾기를 구현해 볼 것