**뱀 게임**

장애물과 알이 있는 미로에 움직이는 뱀을 생각해보자. 뱀은 주어진 위치에서 시작한다<그림 1>. 뱀의 이동 목표는 자기 자신을 건드리지 않고, 벽에 부딪히지 않으며 모든 존재하는 알을 먹는 것이다. 뱀은 처음에는 작지만 하나의 알을 먹을 때마다 뱀의 길이가 1씩 증가하게 된다<그림 2>. (당연히, 주어진 미로보다 커지지는 않는다.)

뱀은 장애물이 없다면, 어떤 방향(상하좌우)으로든 한 칸씩 움직일 수 있다.

장애물과 알이 설정되어 있는 미로에, 뱀이 모든 알을 먹을 수 있는 **최소 이동거리**와 **그에 대한 경로**를 구하는 함수를 작성하라.

처음의 뱀 크기는 미로의 한 칸을 차지하며 그 위치는 미로의 좌상인 (0,0)이다.(그 위치에는 알이나 장애물이 없다.) 뱀이 알을 먹을 때마다, 뱀의 머리가 알이 위치하고 있는 곳에 도달하면, 그 크기는 한 단위만큼 증가하게 된다.

<그림 1> 뱀의 초기 위치

<그림 2> 알이 있는 곳으로의 이동

응용프로그램에서는, Snake라는 DLL 프로젝트에서 생성한 snake.dll파일의 GetSnakePath라는 함수를 호출하여 뱀의 최단 경로 및 거리를 구하게 된다.

GetSnakePath라는 함수의 형태는 다음과 같다.

*void GetSnakePath( int\* distance,*

 *POINT paths[],*

 *const char map[MAP\_SIZE][MAP\_SIZE] )*

응용프로그램에서 GetSnakePath라는 함수를 호출하게 된다.

Parameter :

distance : 이 곳에 뱀의 최소 이동 길이를 반환한다.

 paths : 이 배열에 뱀의 이동 경로를 paths[0]부터 차례대로 넣는다.

map : 입력되는 맵이 들어온다. 5\*5의 2차원 char 배열이 들어오며

‘-‘는 빈 공간, ‘o’는 알, ‘x’는 벽을 나타낸다.

주어진 map이라는 지형 정보를 통해 모든 알을 다 먹는 최소 이동거리와 그 경로를 구하여 distance와 paths에 결과물을 넣는 GetSnakePath 함수를 구현하시오.

<그림 3> 뱀의 이동 경로

한 예로서 위의 <그림 3>과 같은 형태로 map이 입력으로 들어올 경우 12번 이동하여 알을 다 먹는 빨간색의 경로가 최소의 경로 중 하나일 것이다. 이럴 경우의 최소 이동거리와 그에 대한 경로는 다음과 같다.

distance = 12

paths[0] = (0,1), paths[1] = (0,2), paths[2] = (0,3), paths[3] = (0,4)

paths[4] = (1,4), paths[5] = (1,3), paths[6] = (1,2), paths[7] = (1,1)

paths[8] = (1,0), paths[9] = (2,0), paths[10] = (3,0), paths[11] = (4,0)

// Snake.cpp : DLL 응용 프로그램에 대한 진입 점을 정의합니다.

//

#pragma once

#define WIN32\_LEAN\_AND\_MEAN

// Windows 헤더 파일입니다.

#include <windows.h>

#define MAP\_SIZE 5 // MAX MAP SIZE

BOOL APIENTRY DllMain( HANDLE hModule,

 DWORD ul\_reason\_for\_call,

 LPVOID lpReserved )

{

 return TRUE;

}

/\*------------------------------------------------------------------------

\* Snake Problem

\* ------------------------------------------------------------------------

\*/

/\*\* [ 문제 설명 ]

\* - 주어진 지형 정보에서 뱀이 시작 지점(0,0)에서 출발하여,

\* 지형에 존재하는 모든 알을 먹을 수 있는 최단 길이와 경로를 구한다.

\*/

/\*\* [ 참고 사항 ]

\*1) 뱀의 초기 길이는 1이며, 뱀이 알을 먹으면 뱀의 길이가 1씩 증가한다.

\*2) 뱀의 이동은 상하, 좌우이며, 한 칸씩만 가능하다.

\*3) 뱀은 벽으로 이동할 수 없으며, 자기 자신을 건드리지 못한다.

\*4) 뱀의 머리와 꼬리가 사이클을 형성했을 때, 머리와 꼬리는 교차하여 지나갈 \*수 있다.\*/

/\*\* [ 주의 사항 ]

\*1) 함수 GetSnakePath()를 제외한 사용자 정의 함수 및 구조체 등도 작성 \*가능함.\*/

/\*\* [ void GetSnakePath(int\* distance, POINT\* paths, const char map[MAP\_SIZE][MAP\_SIZE] ) ]

\* - Parameter : distance - 뱀의 이동 길이를 넣을 변수

\* paths - 뱀의 이동 경로를 넣을 구조체

\* map[][] - 지형 정보 -> 길: -, 벽: x, 알: o

\*/

/\*------------------------------------------------------------------

\* Begin - Contest Codes

\* ------------------------------------------------------------------

\*/

extern "C" \_\_declspec(dllexport) void GetSnakePath( int\* distance, POINT\* paths, const char map[MAP\_SIZE][MAP\_SIZE] )

{

}

/\*------------------------------------------------------------------

\* End - Contest Codes

\* ------------------------------------------------------------------

\*/