**1. 사용한 각 함수에 대한 설명 (Input, Output, 알고리즘 설명)**

Element

[row,col,dir]로 이루어진 스택과 top을 표시할 position을 생성.

Offset

[Vert,horiz]로 이루어진 이동방향 생성

Offset Move[4]

북동서남으로 탐색방향을 전역변수로 지정.

void Go(int(\*maze)[MAX], int row, int col)

미로 탐색하는 함수. Mark값을 초기화 한 후 시작점인 1,1을 mark한다. Top을 0으로 초기화한다. 스택에 [1,1,1]을 입력하여 출발점으로 두고 북쪽부터 탐색하도록 한다.

출구에 도착하기 전까지 잘못간 구역은 pop함수로 스택에서 제거한 뒤 찾을 때까지 북동서남으로 dir을 +해가면서 탐색을 시도한다. 이미 갔던곳 (mark에 있다면)이면 dir을 증가시켜 다시 탐색한다. 이동 시 ma가에 입력하고 position에 top주소값을 push한다. 출구 도착 시 found=true(1)로 입력하여 탈출한 미로경로를 출력하도록 한다.

element pop(int\* top)

스택을 pop하는 함수

void push(int\* top, element item)

스택에 push하는 함수.

Main()

0은 갈 수 있는 길이고 1은 갈 수 없는 길이다.

미로크기를 입력받아서 그 크기만큼 Srand를 사용하여 랜덤으로 미로를 만든다. 랜덤수를 2로 나눈 나머지값을 쓰도록 하여 0과 1로만 구성한다. 9x9배열을 위해 11x11을 만들어야하므로 미로크기가 커서 미로의 탈출로가 좀 더 자주 만들어 질 수 있도록 대각선에 0을 넣어주는 for문 작성.

미로의 테두리를 모두 1로 입력하여 벽을 만든다. 입구는 1,1. 출구는 9.9 이므로 해당 위치를 0으로 설정해준다.

For문을 이용하여 미로의 형태를 출력한다.

탐색함수를 호출하여 탐색 시작한다.

**2. 실행 성공 결과 화면 단계별 Capture (작은 화면으로 Capture할 것)**

