

Stack Function

[code] DS2001-HW06

Problem

수업 및 실습시간에 배운 Maze 문제에 대해서 모든 가능한 경로 중 이동비용이 최소인 경로의 개수를 리턴하는 move() 함수를 구현하시오.

[제약조건]

- 함수의 prototyp은 다음과 같다.
Int move(int **maze, int **mark, int row, int col, Infor *stack, int *top);
maze : 미로찾기 문제 2차원 배열 (수업시간 배운 개념과 같이 0:진행가능, 1:진행불가를 뜻하며 테두리 행과 열은 모두 1로 채워진다.)
mark : markin을 위한 2차원 배열 (수업시간 배운 개념과 동일하며 mark[1][1]만 1이고 나머지는 0으로 초기화되어 전달된다.)
row, col : maze와 mark 배열의 행과 열의 크기
stack : 좌표들을 저장하는 스택이며 (1,1)이 저장되어 전달된다.
Top : stack의 top(음이 아닌 정수)을 저장하는 포인터 변수이며 *top=0으로 전달
리턴값 : 주어진 미로찾기에서 이동비용이 최소인 경로의 갯수
- 출발지는 (1,1), 목적지는 (row-2, col-2)가 된다.
- 오른쪽 자료구조와 이동비용식을 이용한다.
- **주의) 제출하는 소스파일에는 main 함수는 포함하지 않는다.**
(다음 페이지 예처럼 mycode.c만 제출한다.)

```
#define N 0
#define NE 1
#define E 2
#define SE 3
#define S 4
#define SW 5
#define W 6
#define NW 7
```

```
typedef struct _Infor {
    int x;
    int y;
}Infor;
```

이동비용(per move)

- 수직/수평방향 : 1
- 대각선방향 : 2

Submission

Due: 5월 31일 (일) 23시 59분 59초

19950001@ubuntu:~/DS\$ submit DS2001-HW06 mycode.c

Self Test

```
maze = {1, 1, 1, 1, 1, 1, 1},
```

```
        {1, 0, 0, 0, 0, 1, 1},
```

```
        {1, 1, 1, 0, 1, 1, 1},
```

```
        {1, 1, 0, 0, 0, 0, 1},
```

```
        {1, 1, 0, 1, 1, 1, 1},
```

```
        {1, 1, 0, 0, 0, 0, 1},
```

{1, 1, 1, 1, 1, 1, 1} 인 경우 36가지 경로가 존재하는 데 그 중 최소이동비용 경로는 1개이며 그 때 이동비용은 10이다 .

```
19950001@ubuntu:~/DS$ sftest DS2001-HW06 mycode.c
```

```
MSG> Your source file was successfully compiled.
```

```
1 th. Testing 4 4 0.7 777 ---
```

```
Your answer is 2 : success.
```

```
2 th. Testing 4 5 0.6 1234 ---
```

```
Your answer is 5 : success.
```

```
3 th. Testing 5 4 0.65 321 ---
```

```
Your answer is 4 : success.
```