

3장. 응용예제 답

응용예제 3-1. 게이트볼 공은 어디에?

```
#define _CRT_SECURE_NO_WARNINGS
#pragma warning(disable : 6031) //scanf의 반환값없음warning C6031을 해제하기 위해 추가.
#include<stdio.h>
#define MAX 50

int findBall(int boardSize, int* board, char* swings);

int main(void)
{
    int boardSize, board[MAX][MAX] = { 0 }, success=0;
    char swings[MAX];

    board[0][0] = 1; //시작 위치
    printf("게이트볼 훈련장 크기와 스윙 기록을 입력하세요 :Wn");
    scanf("%d %s", &boardSize, swings);
    success = findBall(boardSize, board, swings);

    if (success != 1) printf("Wn공 위치확인에 실패했습니다. WnWn");
    else printf("Wn공 위치를 확인하였습니다. WnWn");

    return 0;
}

int findBall(int boardSize, int *board, char* swings)
{
    int i=0, j=0, count=1;

    printf("Wnstart : (%d, %d)", i + 1, j + 1); //게이트볼이 있는 시작 셀 위치 출력
    (게이트볼의 이동 경로 출력용)

    while (*swings) {
        switch (*swings) {
            case 'L': //왼쪽 스윙인 경우 :
                j--; //현재 셀 위치를 왼쪽으로 이동
                if (j < 0) j++; //왼쪽 벽에 부딪힌 헛스윙인 경우, 열번호 j 수정 안함.

                break;

            case 'R': //오른쪽 스윙인 경우
                j++; //현재 셀 위치를 오른쪽으로 이동
                if (j >= boardSize) j--; //오른쪽 벽에 부딪힌 헛스윙인 경우, 열번호
j 수정 안함.
```

```

        break;

    case 'U': //위쪽 스윙인 경우
        i--; //현재 셀 위치를 위로 이동
        if (i < 0) i++; //위쪽 벽에 부딪힌 헛스윙인 경우, 행번호 i 수정 안함.
        break;

    case 'D': //아래쪽 스윙인 경우
        i++; //현재 셀 위치를 아래로 이동
        if (i >= boardSize) i--; //아래쪽 벽에 부딪힌 헛스윙인 경우, 행번호 i
수정 안함.
        break;

    default:
        printf("WnWn스윙기록이 잘못된 값입니다. Wn");
        return 0; //함수 실행 중단
    }
    *(board + i*boardSize + j) = count; //행우선 순서에 따라 2차원 배열의 원소
위치를 지정하여 스윙count 저장
    printf("Wnswing%2d: (%d, %d)", count++, i+1, j+1); //현재 셀 위치 출력
(게이트볼의 이동 경로 출력용)
    swings++;
}

printf("WnWn >> 공 마지막 위치 : (%d, %d) WnWn", i+1, j+1);

/* //공의 이동 순서를 저장한 board 출력하기 : 재미를 위해 추가함.
printf("< 공의 이동 위치 >Wn");
for (i = 0; i < boardSize; i++) {
    for (j = 0; j < boardSize; j++) {
        if (i == 0 && j == 0) printf("%3c", 'S'); //시작위치 셀은 'S'로 표시
        else printf("%3d", *(board + i * boardSize + j)); //행우선 순서에
따라 2차원 배열의 원소 위치 지정
    }
    printf("Wn");
}
*/

return 1;
}

```

[실행 예]

```
C:\WINDOWS\system32\cmd.exe
게이트볼 훈련장 크기와 스윙 기록을 입력하세요 :
4
DLRRDL
start : (1, 1)
swing 1: (2, 1)
swing 2: (2, 1)
swing 3: (2, 2)
swing 4: (2, 3)
swing 5: (3, 3)
swing 6: (3, 2)

>> 공 마지막 위치 : (3, 2)

공 위치를 확인하였습니다.
계속하려면 아무 키나 누르십시오 . . .
```

재미를 위해 출력문 추가

```
C:\WINDOWS\system32\cmd.exe
게이트볼 훈련장 크기와 스윙 기록을 입력하세요 :
5
RRDDDDDDRRRRRRRRR
start : (1, 1)
swing 1: (1, 2)
swing 2: (1, 3)
swing 3: (2, 3)
swing 4: (3, 3)
swing 5: (4, 3)
swing 6: (5, 3)
swing 7: (5, 3)
swing 8: (5, 3)
swing 9: (5, 3)
swing10: (5, 3)
swing11: (5, 4)
swing12: (5, 5)
swing13: (5, 5)
swing14: (5, 5)
swing15: (5, 5)
swing16: (5, 5)
swing17: (5, 5)
swing18: (5, 5)
swing19: (5, 5)

>> 공 마지막 위치 : (5, 5)

공 위치를 확인하였습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\WINDOWS\system32\cmd.exe
게이트볼 훈련장 크기와 스윙 기록을 입력하세요 :
4
DLRRDAAAAA
start : (1, 1)
swing 1: (2, 1)
swing 2: (2, 1)
swing 3: (2, 2)
swing 4: (2, 3)
swing 5: (3, 3)

스윙기록이 잘못된 값입니다.
공 위치확인에 실패했습니다.
계속하려면 아무 키나 누르십시오 . . .
```

잘못 입력한 경우

[실행 예 : /* //공의 이동 순서를 저장한 board 출력하기 ~ */ 주석(/*, */)을 해제하고 실행한 경우]

```
C:\WINDOWS\system32\cmd.exe
게이트볼 훈련장 크기와 스윙 기록을 입력하세요 :
4
DLRRDL
start : (1, 1)
swing 1: (2, 1)
swing 2: (2, 1)
swing 3: (2, 2)
swing 4: (2, 3)
swing 5: (3, 3)
swing 6: (3, 2)

>> 공 마지막 위치 : (3, 2)
< 공의 이동 위치 >
S 0 0 0
2 3 4 0
0 6 5 0
0 0 0 0

공 위치를 확인하였습니다.
계속하려면 아무 키나 누르십시오 . . .
```

재미를 위해 출력문 추가

응용예제 3-2. TDM 축소 행렬 만들기

```
#define _CRT_SECURE_NO_WARNINGS
#pragma warning(disable : 6031) //scanf의 반환값없음warning C6031을 해제하기 위해 추가.
#include<stdio.h>
#include <malloc.h>
#include <string.h>
#define MAX_N 10
#define MAX_M 30

int reduceTDM(int(*TDM)[MAX_M], int(*rTDM)[3], int N, int M, char *terms[]);

int main(void)
{
    int N = 0, M = 0, TDM[MAX_N][MAX_M] = {0}, rTDM[MAX_N * MAX_M][3] = { 0 };
    char *terms[MAX_N] = { NULL }, temp[100];
    int i = 0, j = 0, len, count=0;

    printf("TDM을 구성할 단어 개수 N과 단어목록을 입력하세요 : ");
    scanf("%d", &N);

    while (i < N) {
        scanf("%s", temp);
        //입력받은 문자열을 동적할당으로 terms 포인터 배열에 할당하기
        len = strlen(temp);
        if (len > 0) {
            char* strptr = (char*)malloc(sizeof(char) * (len + 1));
            strcpy(strptr, temp);
            terms[i] = strptr;
            i++;
        }
    }

    printf("\nTDM을 구성할 문서 개수 M을 입력하세요 : ");
    scanf("%d", &M);
    printf("\nTDM을 입력하세요 (MxN 개) : \n");
    for(i=0; i<N; i++)
        for(j=0; j<M; j++)
            scanf("%d", &TDM[i][j]);

    count = reduceTDM(TDM, rTDM, N, M, terms);
    printf("\nWnWncount : %d Wn", count); //0이 아닌 원소 갯수(축소된TDM의 행의 갯수) 출력
=> 확인용

    return 0;
}

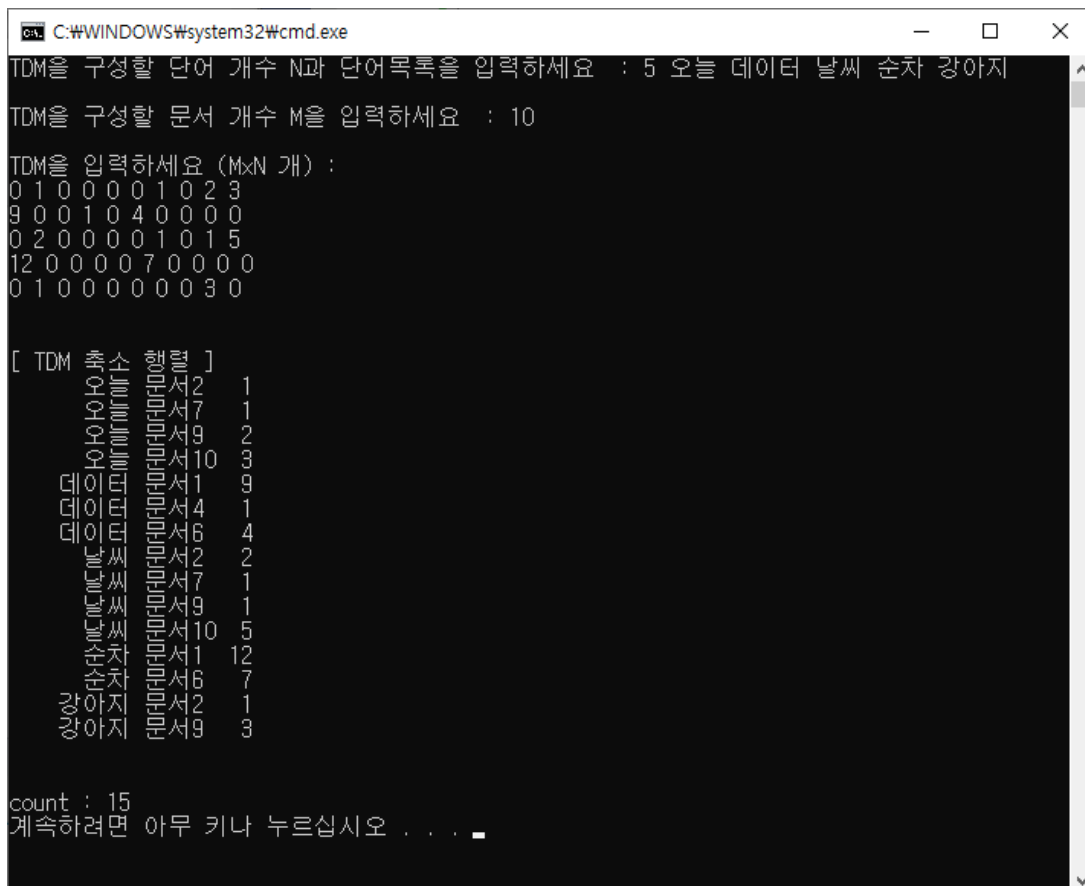
int reduceTDM(int (*TDM)[MAX_M], int (*rTDM)[3], int N, int M, char* terms[])
{
    int i = 0, j = 0, count=0;
    // TDM 축소행렬 만들기
```

```

for (i = 0; i < N; i++)
    for (j = 0; j < M; j++)
        if (TDM[i][j] != 0) {
            rTDM[count][0] = i;
            rTDM[count][1] = j;
            rTDM[count][2] = TDM[i][j];
            count++;
        }
// TDM 축소행렬 출력하기
printf("WnWn[ TDM 축소 행렬 ]Wn");
for (i = 0; i < count; i++)
    for (j = 0; j < 3; j++) {
        switch (j) {
            case 0:
                printf("%10s", terms[rTDM[i][j]]);           break;
            case 1:
                printf(" 문서%-2d", rTDM[i][j]+1);           break;
            case 2: printf("%3d Wn", rTDM[i][j]);
        }
    }

return count; //0이 아닌 원소 갯수(축소된TDM의 행의 갯수)
}

```



```

C:\WINDOWS\system32\cmd.exe
TDM을 구성할 단어 개수 N과 단어목록을 입력하세요 : 5 오늘 데이터 날씨 순차 강아지
TDM을 구성할 문서 개수 M을 입력하세요 : 10
TDM을 입력하세요 (MxN 개) :
0 1 0 0 0 0 1 0 2 3
9 0 0 1 0 4 0 0 0 0
0 2 0 0 0 0 0 1 0 1 5
12 0 0 0 0 0 7 0 0 0 0
0 1 0 0 0 0 0 0 3 0

[ TDM 축소 행렬 ]
      오늘 문서2 1
      오늘 문서7 1
      오늘 문서9 2
      오늘 문서10 3
  데이터 문서1 9
  데이터 문서4 1
  데이터 문서6 4
  날씨 문서2 2
  날씨 문서7 1
  날씨 문서9 1
  날씨 문서10 5
  순차 문서1 12
  순차 문서6 7
  강아지 문서2 1
  강아지 문서9 3

count : 15
계속하려면 아무 키나 누르시오 . . .

```