**과제4. 편집기 지뢰찾기**



|  |  |
| --- | --- |
| **과 목 :** | **프로그래밍** |
| **제출 일자 :** | **2018. 11. 29** |
| **담당 교수 :** | **김승수** |
| **학 과 :** | **사회복지학부** |
| **학 번 :** | **20170299** |
| **이 름 :** | **김유빈** |

1. 문장 추가(함수 MEMORY)

str

hello world

nice to meet you

NULL

nice to meet you

nice to meet you

**과제(1) 편집기**

첫번째 시도: 4개의 문장

->i=4 (str[3]까지 처리)

Last = i=4

두번째 시도: 5개의 문장

->i= last(=4)

str배열 처리과정 결과 i = 4 + 5 = 9

**i = 3**

Last = i = 4 + 5 = 9

**i = 4**

I am a boy

last

you are a girl

korea

**i = 8**

good job

nice idea

**i = 9**

2. 문장 삭제(ERASE 함수)

\*메모리 삭제 : free함수

\*포인터를 NULL로 (아무것도 가리키지 않는다)

-> 만약 포인터 == NULL

PRINTF((deleted));

3. 문장 정리(ARRANGE함수)

str

hello world

nice to meet you

NULL

nice to meet you

nice to meet you

**\*\*\*\*\*\*\*중요점\*\*\*\*\*\*\*  
메모리 이동(X) -> 포인터 이동!!!**

str[i] == NULL이면

그 뒤에 있는 포인터 다 앞으로 옮기기

NULL

str

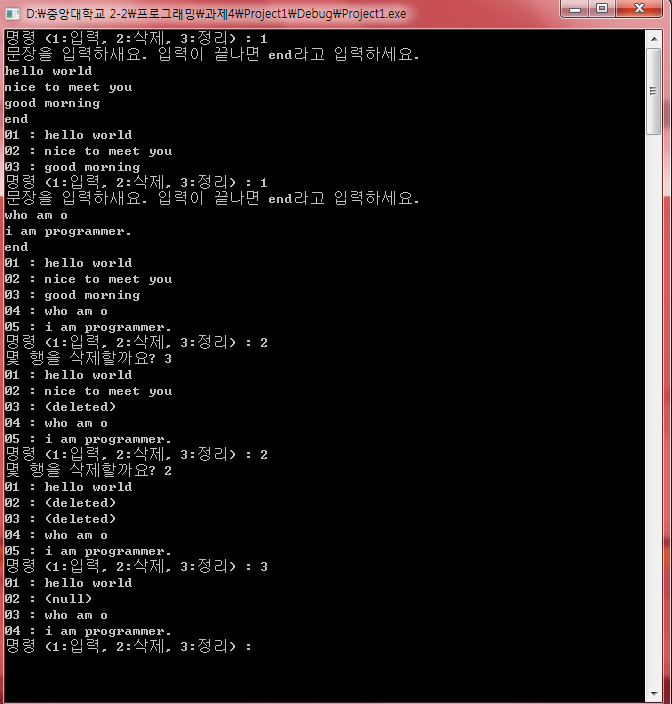
hello world

nice to meet you

nice to meet you

nice to meet you

NULL



* **좀 더 쉽게 하기 위해,** 
  + 2개의 배열을 사용한다.
  + 1개는 폭탄의 위치 또는 숫자를 저장하기 위한 것
    - 폭탄의 위치와 화면에 표시할 숫자를 미리 계산하여 저장한다.
    - 이를 **block[9][19]** 라 하자.
    - 폭탄의 위치에는 -1이, 그 외의 위치에는 주변 폭탄 개수를 나타내는 숫자가 저장되어 있다.
  + 1개는 열린 블록인지 아닌지를 저장하기 위한 것
    - 사용자가 연 블록인지 아직 열지 않은 블록인지 저장한다.
    - 이를 **mask[9][19]** 라 하자.
    - 처음은 모두 0이다.(닫혀 있음) 사용자가 열거나 처음에 자동으로 열리면(시작 화면 만드는 방법에 의해 자동으로 열림) 해당 블록을 1로 만든다.

**과제 (2) 지뢰찾기**

1. 폭탄 생성(배열 block)

row[i] = rand() % 9;

line[i] = rand() % 19;

(+) 폭탄을 생성했을 때 중복되는 것도 피해야함

|  |  |  |
| --- | --- | --- |
| ①-1 | ②1 | ③2 |
| ④2 | **2** | ⑤0 |
| ⑥0 | ⑦1 | ⑧-1 |

2. 폭탄 개수 새기(배열block)

단, block[i][j] == -1일 때 (폭탄) 폭탄 개수는 셀 필요 없다!!

①block[i - 1][j - 1] == -1

②block[i - 1][j] == -1

③block[i - 1][j + 1] == -1

④block[i][j - 1] == -1

⑤block[i][j + 1] == -1

⑥block[i + 1][j - 1] == -1

⑦block[i + 1][j] == -1

⑧block[i + 1][j + 1] == -1

* block[i][j]++;

단, i – 1, j – 1 >= 0 // i + 1 < 9, j + 1 < 19

4. 열린 블록, 닫힌 블록 저장(배열mask)

if(block[i][j] == 0)

3번 처럼 주변 8개를 mask = 1로 넣기(1: 열림, 0: 닫힘)

mask[i][j] = 1로 하기

5. 시작 게임판 출력

* + - * if (mask[i][j] == 1)   
         printf(“%d”, block[i][j]);  
        else  
         printf(“?”);

단, block[i][j] == 0 이면 printf(“ “); -> 빈칸 출력

**6, 열 블록 선택★★★**

scanf("%c%c", &x,&y); -> 처음 입력 받을 때

**scanf(" %c%c", &x,&y); -> 그 이후**

처음 x,y좌표 입력할 때 입력하고 나서 ‘엔터키’를 누른다

따라서 다음 입력 받을 때 x 값에 엔터키가 들어가는 문제가 발생!!!!!!

* 이를 해결하기 위해 %c앞에 빈칸(공백)

입력 받는 변수: char형 x,y

실제 배열에 쓰이는 변수: int형 r,w

r = x - '0';

if (y >= 65)

w = y - 'A' + 10;

else

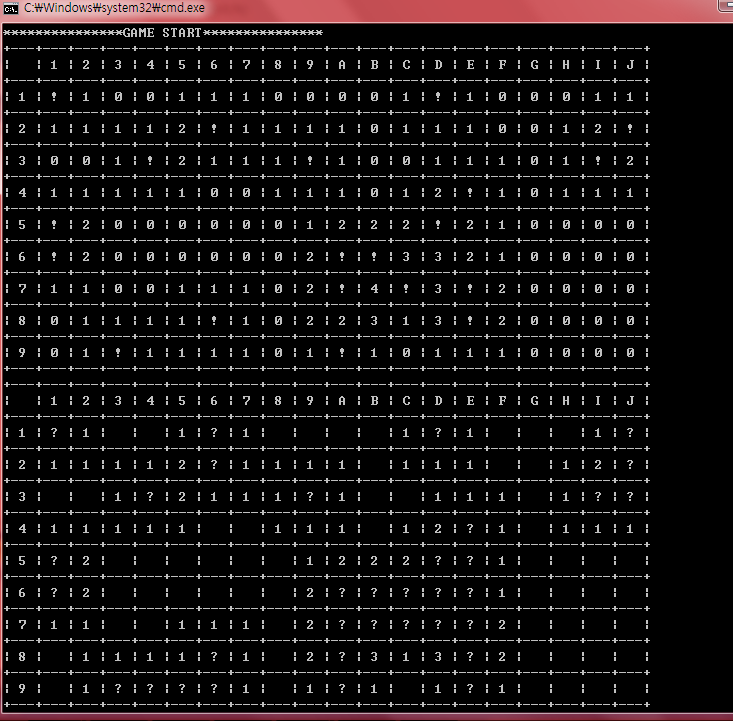
w = y - '0';

7. 게임 진행

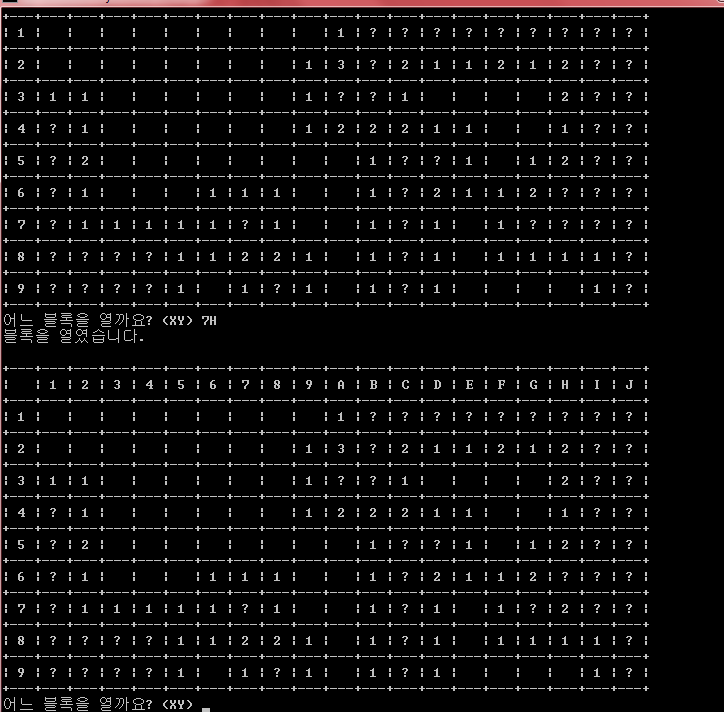
block[r-1][w-1] != -1 이거나 mask의 값이 1인 개수 = 19\*9 – 20인 경우

: 게임 끝!

열고 싶은 칸을 mask = 1로 해서 출력



* 첫 화면이 올바르게 나오는지 테스트



->게임 실행 모습

