Lab 7 - Exercise – All sections
Two Dimensional (2D) Arrays
CS 262 – Spring 2022

The purpose of this activity is to practice how store strings in a 2D array (matrix) and how access each cell of the Matrix to get/modify a value.

An array of arrays is referred as a two-dimensional (2D) array, also known as **matrix**. Visually, a matrix can be represented as a table of rows and columns. Each individual item can be accesses by referencing its **`row-index`** and **`col-index`** as follows:

```
name[row_index][col_index]
```

Here is an example of a 2D array of integers size 3 X 4:

```
int numbers[3][4];
```

	0	1	2	3
0		🟨		
1	🟩			
2				🟧

Here are different ways how we can access the "marked cells"

```
sscanf(inBuf, "%i", &numbers[0][1]);
numbers[1][0] = 5;
printf("%i\n", numbers[2][3]);
```

/* Note that in C, array indices start with 0 */

In general cases, numeric 2D arrays (`int`, `float`, etc.) use a cell to store an element. However, 2D arrays of `char` are different, because one of the cells is used to store the '`\0`' character to indicate the end of the string in that row. Suppose we want to store the words "**Mon**", "**Fri**" each in a row of a 2D array. It might think that a 2x3 matrix is needed, however an additional column is required for the '`\0`' character. Visually the storage would be as follows:

```
char text[2][4];
/* ... */
```

	0	1	2	3
0	M	o	n	\0
1	F	r	i	\0

Because 2D matrices of type `char` can contain the `null char` to indicate the end of the string, when ONLY the row-index is specified, we can get all values of the row until we reach `'\0'`. Here an example using the previous definition:

```
printf("%s", text[1]);          //it prints Fri as output
strcpy(text[0],"Sun");          //it assigns Sun in the first row of text
sscanf(pt,"%s", text[0]);       //it copies the string pointed by pt to the 1st row of text
```

In the above example the **`col-index`** is omitted. In a 2D array of `char`, it gives us access to the entire row. However, when both indices are given, we have access to an individual cell.

Description of the program

Code a C program that gets two words as user input, store those words in a 2D array of characters. Output each word backwards. Printing must be done by accessing each cell in each row in "descending" order.

General steps:

- Prompts the user to enter a word and store it in the corresponding row on the 2D matrix. Perform this twice.
- In a nested loop access each char that you want to print.
  - Hint: output loop is for rows and inner loop for cols.

Requirements:

- Declare `char words[ROWS][COLS]` as the matrix to store the user input.
- Use `fgets()` and `sscanf()` to get each word, use `%s` in both functions
- Assume that the word the user enters has a maximum of 15 characters. Define an appropriate value for `ROWS` and `COLS` based on this information.

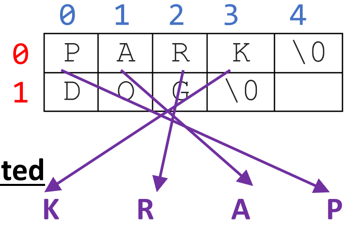
Example 1:

```
Word 1: Hello
Word 2: World
Output:
olleH
dlroW
```

Example 2:

```
Word 1: PARK
Word 2: DOG
Output:
KRAP
GOD
```

Stored:



	0	1	2	3	4
0	P	A	R	K	\0
1	D	O	G	\0	

Printed

Additional columns were intentionally omitted
This visual example is intended solely as a support to implement your program, spaces between letters are intentional