Programming C - HW3

This exercise is based on **arrays** and **strings**. We will explore these topics using a few examples. Hope you have fun solving it! :)

- Throughout the whole exercise, there is no need to use external libraries (other than those that are already included), and it is forbidden!
- Notes for submission: (replace 123456789 by your ID number) You need to upload a single .zip file named: ex3_123456789.zip. This file will include all of the 3 attached .c files, one file for the solution of each section.
- All the files must be renamed in the following fashion: You should add “ex3_” before each filename, and add “_123456789” at the end, before the .c. For example, upon submission, the file of question 1, should look like this: ex3_Q1_123456789.c
- Follow the function signatures if given. If they are not given, follow the best practices when coming up with your own signatures. But make sure they are readable and are defined in the right fashion.
- **Please do not change the template and don’t do any printing other than the one already included.**

Question 1:

A logical matrix is a matrix in which all its elements are either 0 or 1. We define logical multiplication of matrices A and B by the operation defined below, where "·" is the logical AND operation, and "+" is the logical OR operation.
In this assignment, you will create two 5x5 logical matrices and find the corresponding matrix which will be created from "multiply" these 2 matrices

$$c[i][j] = \sum_{k=0}^{SIZE-1} a[i][k] \cdot b[k][j]$$

1) Define global SIZE equals to 5 (Already defined in the template)

2) Write a function that gets a matrix reference and reads the input to the matrix from the user. If the input is non-zero replace it by 1. If the user did not enter enough values before the end of the line, the remaining cells in the matrix will be populated with zeros. Also make sure if the user inputs too many characters, you only take what's needed and discard the remaining input. (Eg: 1509 is a 2x2 matrix with values 1101, and '1 5' is also a 2x2 matrix with values 1111, the highlighted whitespace is taken as a 1 as discussed above.)

Function signature: void read_mat(int mat[][SIZE])

3) Write a function that multiplies, as defined above, two matrices and enters the results into a third matrix with suitable dimensions.

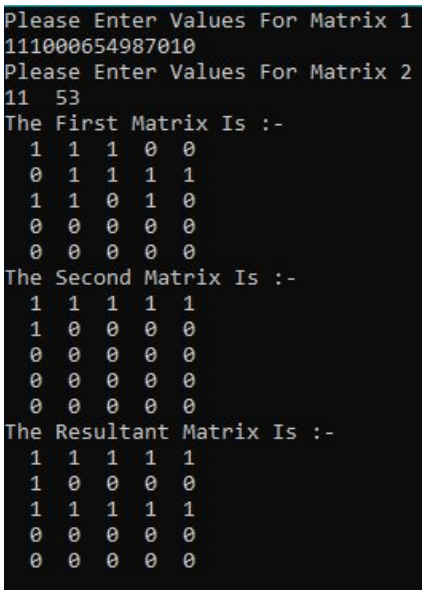
Function signature: void mult_mat(int mat1[][SIZE],int mat2[][SIZE], int result_mat[][SIZE])

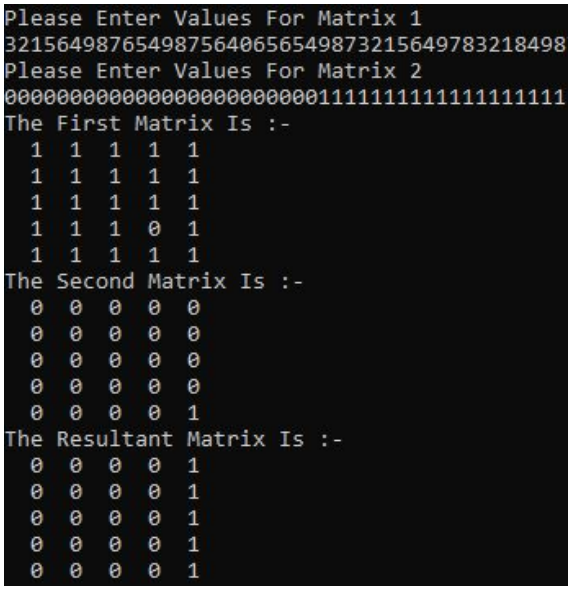
4) Write a function that prints a matrix into the screen. Please use "%3d" for printing format to make it look nice as shown below.

Function signature: void print_mat(int mat[][SIZE])

5) Write the main program which uses the functions above. The program reads the matrices values from the user, multiplies them and prints the result matrix on the screen.

The function definitions given are intentional with the return statements as void. Do not change them. Arrays are transferred between functions as references rather as primitives like variables. So the function definitions are perfectly valid.
Also, there is no limit on the input from the user. You can read only the required digits, and then stop reading, and discard the remaining input.
The examples are given below:


Please Enter Values For Matrix 1
111000654987010
Please Enter Values For Matrix 2
11 53
The First Matrix Is :-
1 1 1 0 0
0 1 1 1 1
1 1 0 1 0
0 0 0 0 0
0 0 0 0 0
The Second Matrix Is :-
1 1 1 1 1
1 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
The Resultant Matrix Is :-
1 1 1 1 1
1 0 0 0 0
1 1 1 1 1
0 0 0 0 0
0 0 0 0 0


Please Enter Values For Matrix 1
3215649876549875640656549873215649783218498
Please Enter Values For Matrix 2
00000000000000000000000011111111111111111111
The First Matrix Is :-
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 0 1
1 1 1 1 1
The Second Matrix Is :-
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 1
The Resultant Matrix Is :-
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1

Question 2:

Write a function to find the number of words in a sentence.

- I have defined a macro of 150 so we can have strings up to 150 characters in length.

1) Write a function to take in an input string and store it an array reference.
2) Write another function to take the input string and find the number of words in the sentence.

Function Signature: int get_number_of_words(char input_string[])

Make sure you're counting the number of words exactly and not finding an 'alternative' method to find them :D
In our case word is defined as any combination of alphabets or numbers. Anything else is not considered as a word, for eg: Whitespaces, tabs, punctuations or special characters.
You can make another function to read the input string or can read it in the main function itself. It's up to you. Just do not change the given function signature.

I have also added the library of ctype.h for your convenience.

```
Enter the string:
This is an example string.
Number of words in given string are: 5
```

```
Enter the string:
This is an              example.
Number of words in given string are: 4
```

```
Enter the string:
,This is,a sentence  :bla,,bla  bla@robot? and3@3banana e
Number of words in given string are: 11
```

The words in the above sentence are:
*This *is *a *sentence *bla *bla *bla *robot *and3 *3banana *e
Each * marks a new word.

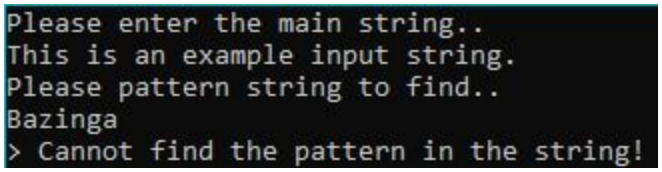
Question 3:

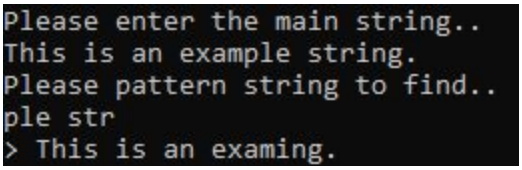
Write a function to get as input 2 strings and then delete the occurrences of the second string (the pattern string) from the first.
A pattern here is defined as any string of characters containing alphabets, special characters, numbers or even whitespaces, which may or may not be present in the original string.
It can also be present multiple times in the same input string.

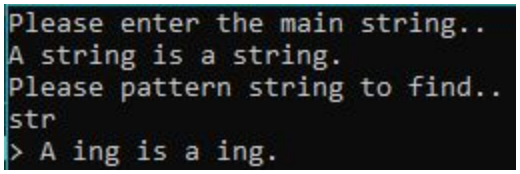
- I have defined a macro of 100 so we can have strings up to 100 characters in length.

1) Take as input both the first and second string one after the other from the user until the newline character is entered.
2) Write a function to check the occurrences of the pattern string in the first and then deletes it from the first one.
3) If we do find a string, return 1 otherwise return 0. It will print the resultant string with the characters removed if such exists.

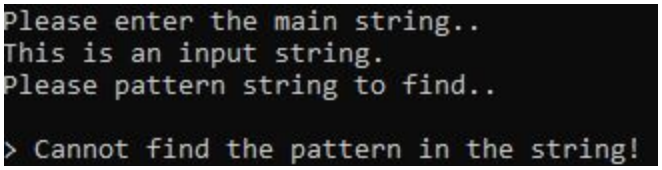
Here I am not providing any function signature. Make them as you see them fit.







In this case, the pattern string was empty:



In this case, the input was whitespace: