Assignment HW27 - Final Test FA-2014
Due date (on or before): Announced in class.

Version 1

Grading:
Each question worth 25 pts. There are 4 questions.
General rubric for all questions.

10 pts for results:

0-5 Working only in one case, not exactly the right results.
6-10 Work for general case, EXACTLY the right results.

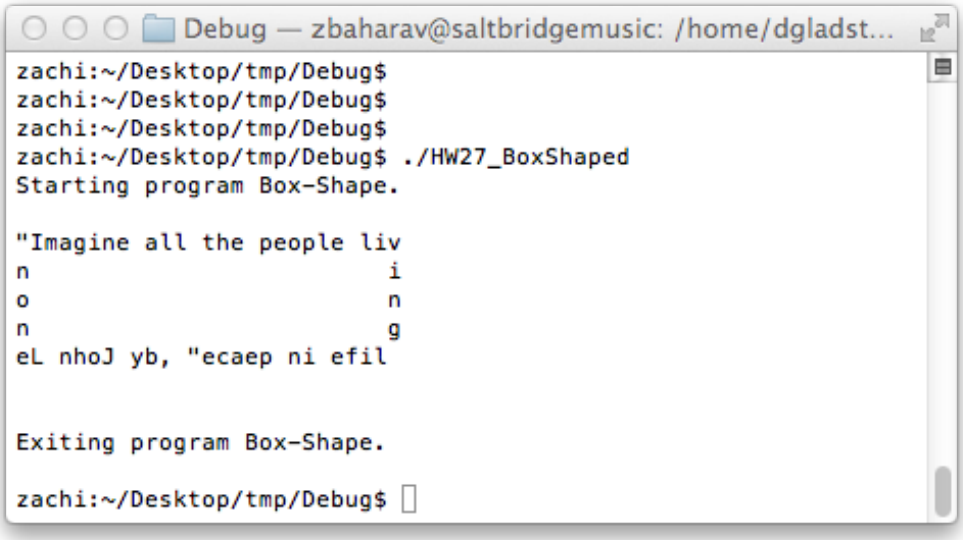
15 pts for coding: Efficency, clear, concise. Hardcoded.

0-7 Hardcoded, not clear, long.
8-15 uses constants, clear and concise, well documentd

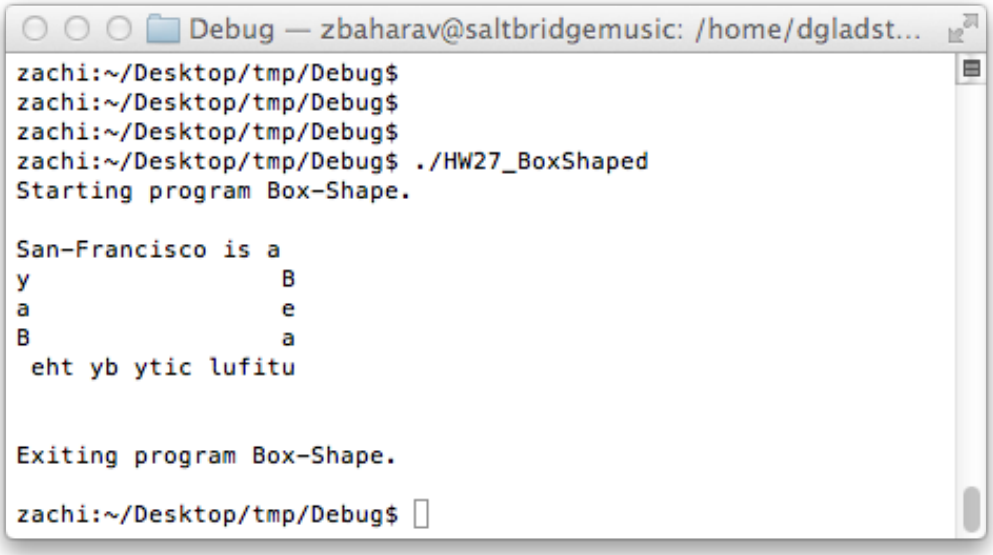
1. Box-Shape of a string (25pts)

Write a program that takes a string, and prints it in a Box-Shape, as shown below.
The Box has height of 5, and the rest printed horizontaaly.
You can aassume (as the case is for the two strings provided), that the length of the string is such that the box will be nicely wrapped around.

Example 1: String is
`"\"Imagine all the people living life in peace\" ,by John Lenon"`



Example 2: String is
`"San-Francisco is a Beautiful city by the Bay"`



You can assume in your program the String has odd-number of characters, and is defined as:

```
char* str="San-Francisco is a Beautiful city by the Bay";
```

or

```
char* str="\"Imagine all the people living life in peace\" ,by
John Lenon";
```

Specifically, you do NOT need to read the string from the user.

********** Start of code + Screen shot**********

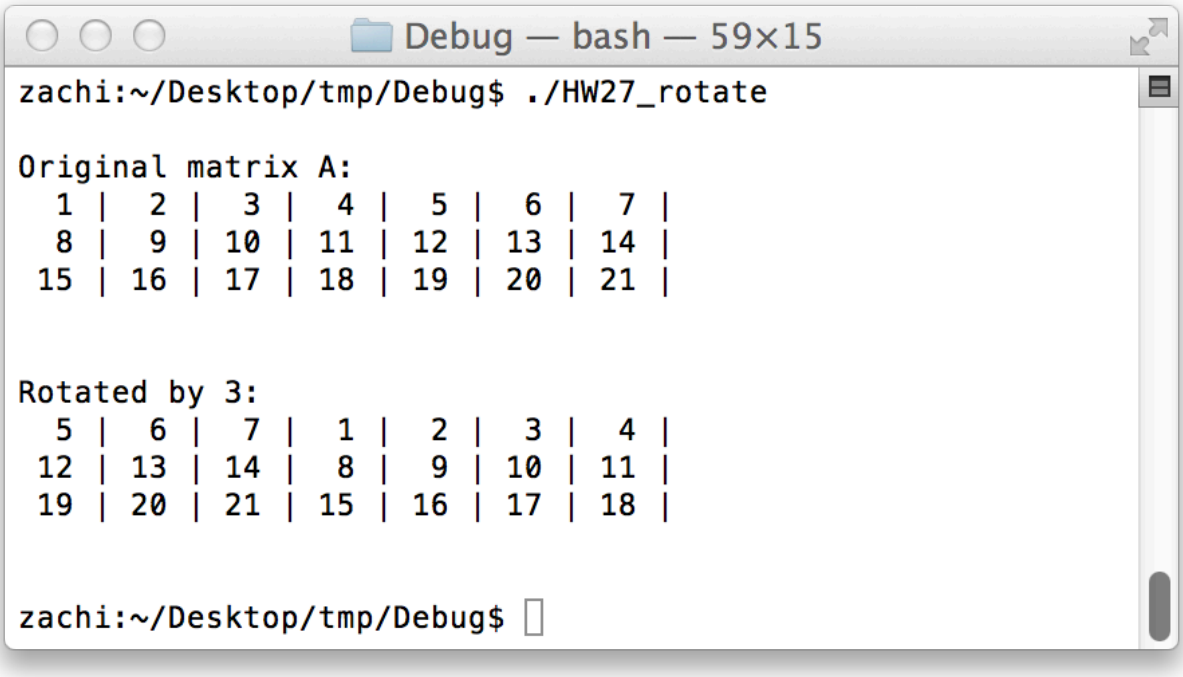
********** End of code + Screen shot**********

2. 2D array (25pts)

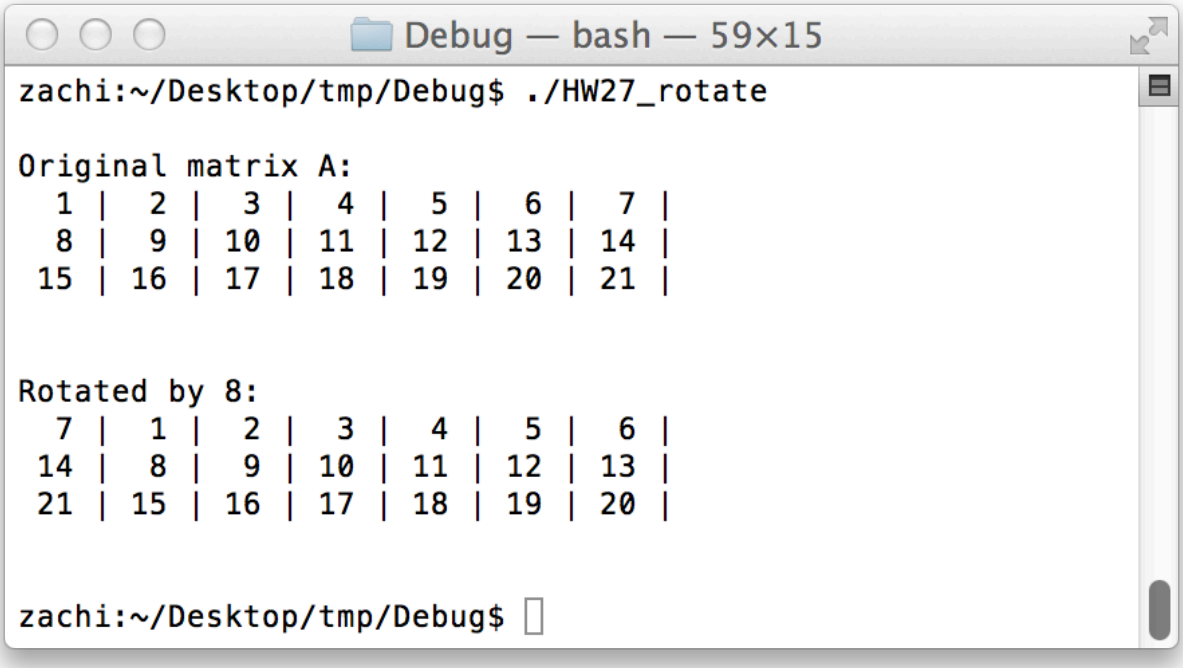
Write a program that takes a 2D array, and prints it once as-is, and once with all the rows shifted (circulary) by a given positive amount to the right. See two examples below.

Example 1: Using 2D array 3x7.

```
#define ROW 3
#define COL 7
int A[ROW][COL] = {
         1, 2, 3, 4, 5, 6, 7,
         8, 9,10,11,12,13,14,
        15,16,17,18,19,20,21
          };
```



Example 2: Make sure the program also works for a shift larger than the number of columns. For example: Shift of 8:



You can assume in your program the matrix is defined as in the example above. Specifically, you do NOT need to read the matrix from the user.

********** Start of code + Screen shot**********

********** End of code + Screen shot**********

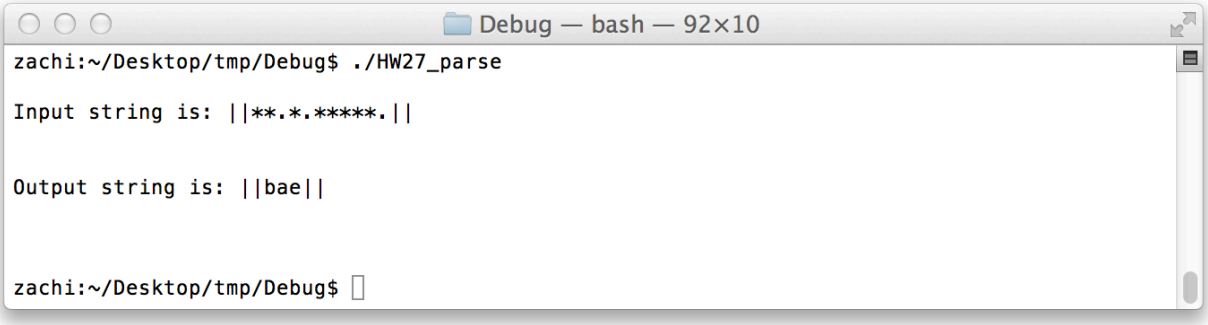
3. Strings (25pts)

Write a program that decodes a simplified Morse code and creates letters. Assume only non-capital letters, and the coding is as followd:
*. → a
**. → b
***. → c
.. and so on.

The program should print the original string (code), and then create a NEW string that holds the letter, and print that as well.

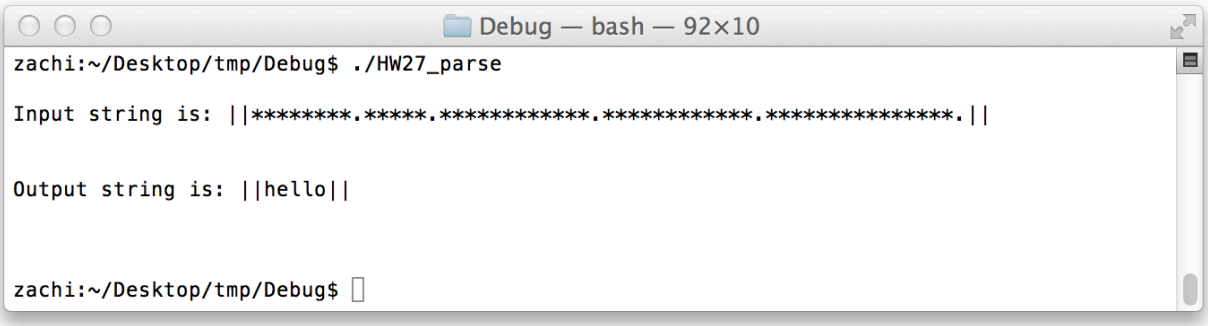
Example 1:

```
char* inStr = "**.*.*****.";
char outStr[80];
```



Example 2:

```
char* inStr = "********.*****.**********.**********.***************." ;
char outStr[80];
```



You can assume in your program the strings are defined as in the examples above.

Specifically, you do NOT need to read the strings from the user, or be concerned about too long of a string.

** You are allowed to use the function strlen(), from <strings.h>, if you so wish. (not a must!!)
** You are not allowed to use any other string manipulation functions from the library.

********** Start of code + Screen shot**********

********** End of code + Screen shot**********

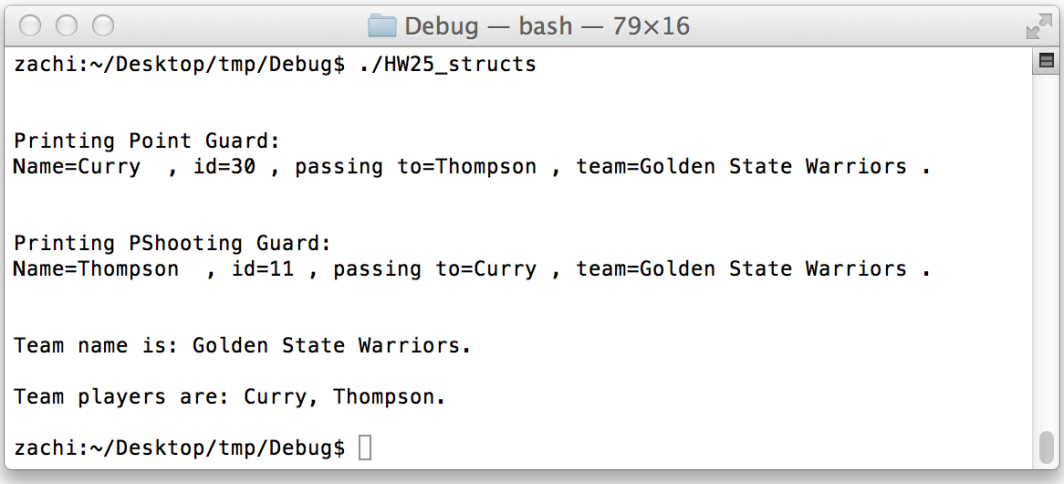
4. Structures (and pointers) (25pts)

Having the following structures declared and intialized as described:

```
#include <stdio.h>
struct Player{
    char name[80];
    int  id;
    struct Player *ptr1;
    struct Team   *ptr2;
};

struct Team{
    char name[80];
    struct Player *p1,*p2;
};

int main(void)
{
    struct Player point = {"Curry",30,NULL};
    struct Player shooting = {"Thompson",11,NULL};
    struct Team   GSA = {"Golden State Warriors",NULL,NULL};


    // Your code HERE
    return(0);
}
```

Fill in the part indicated above, such that you will assign values to the pointers in the above structures, with the approriate values as evident from the output below. (right now these pointers are assigned NULL). DO NOT change the program above. Just add things below as you see apprpriate.

Then, print the structures to get the below output.



********** Start of code + Screen shot**********

********** End of code + Screen shot**********

=== END ===