

Assignment HW25 - Dry run
Due date (on or before): Announced in class.

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: Efficiency, clear, concise. Hardcoded.

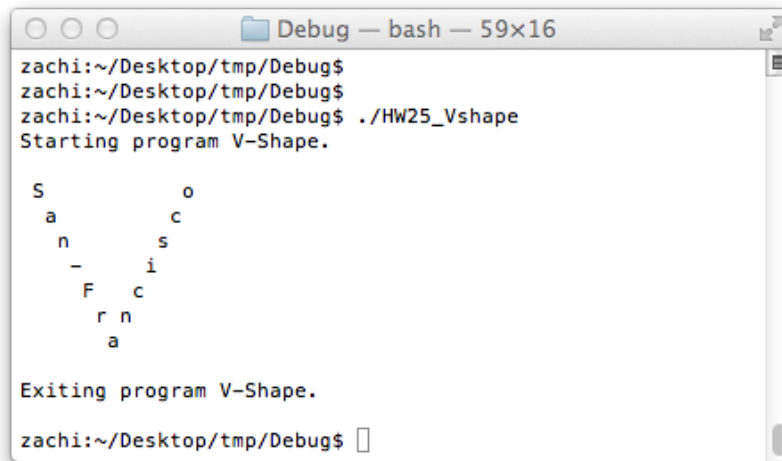
0-7 Hardcoded, not clear, long.

8-15 uses constants, clear and concise, well documentd

1. V-Shape printing (25pts)

Write a program that takes a string, and prints it in a V-Shape, as shown below.

Example 1: String is "San-Francisco"

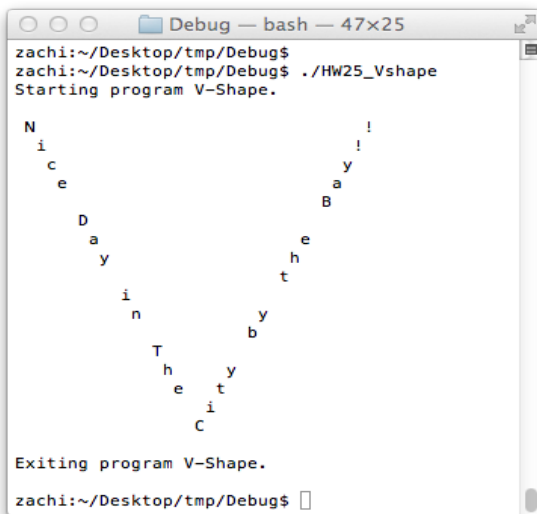


```
zachi:~/Desktop/tmp/Debug$
zachi:~/Desktop/tmp/Debug$
zachi:~/Desktop/tmp/Debug$ ./HW25_Vshape
Starting program V-Shape.

S          o
 a         c
n         s
 -        i
  F       c
   r      n
    a

Exiting program V-Shape.
zachi:~/Desktop/tmp/Debug$
```

Example 2: String is "Nice Day in The City by the Bay!!!"



```
zachi:~/Desktop/tmp/Debug$
zachi:~/Desktop/tmp/Debug$ ./HW25_Vshape
Starting program V-Shape.

N           !
 i          y
  c         a
   e        B
    D       h
     a      t
      y     e
       i    b
        n   y
         T  y
          h e
           i t
            C

Exiting program V-Shape.
zachi:~/Desktop/tmp/Debug$
```

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

```
//char* str="San-Francisco";
char* str="Nice Day in The City by the Bay!!!";
```

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

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

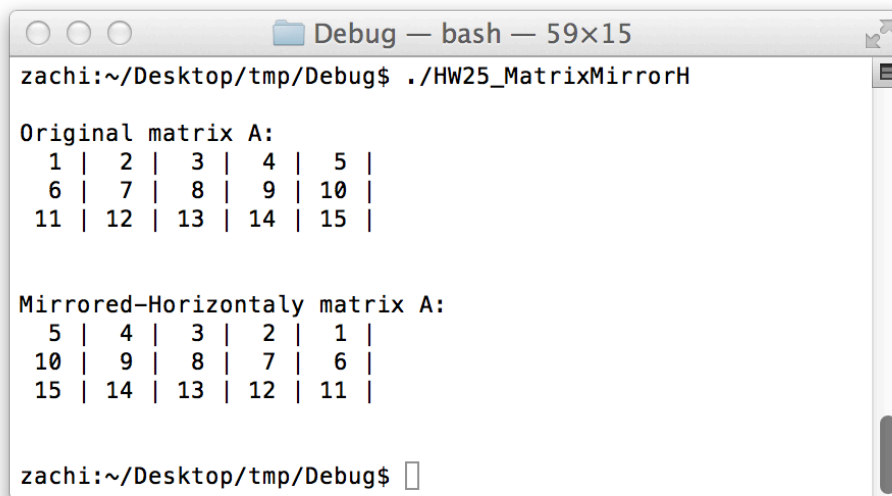
***** End of code + Screen shot*****

2. 2D array (25pts)

Write a program that takes a 2D array, and prints it once as-is, and once mirrored around the middle column. See two examples below, one for odd number of columns (and therefore the 'middle' column doesn't change), and then for even number of columns (Where all columns are moved around).

Example 1: Using 2D array 3x5. Odd number of columns.

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



```
zachi:~/Desktop/tmp/Debug$ ./HW25_MatrixMirrorH

Original matrix A:
 1 | 2 | 3 | 4 | 5 |
 6 | 7 | 8 | 9 | 10 |
11 | 12 | 13 | 14 | 15 |

Mirrored-Horizontal matrix A:
 5 | 4 | 3 | 2 | 1 |
10 | 9 | 8 | 7 | 6 |
15 | 14 | 13 | 12 | 11 |

zachi:~/Desktop/tmp/Debug$
```

Example 2: Using 2D array 3x4. Even number of columns.

```
#define ROW 3
#define COL 4
int A[ROW][COL] = { 1, 2, 3, 4,
                   5, 6, 7, 8,
                   9, 10, 11, 12};
```

```
Debug — bash — 59x15
zachi:~/Desktop/tmp/Debug$ ./HW25_MatrixMirrorH

Original matrix A:
 1 | 2 | 3 | 4 |
 5 | 6 | 7 | 8 |
 9 | 10 | 11 | 12 |

Mirrored-Horizontal matrix A:
 4 | 3 | 2 | 1 |
 8 | 7 | 6 | 5 |
12 | 11 | 10 | 9 |

zachi:~/Desktop/tmp/Debug$
```

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

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

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

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)

In this question, you will have (three) structures that are already defined and partially initialized. These structures will include, for example, char arrays, and pointers to other structures.

You will have a printout describing the relations between the structures, and you will have to do two things:

1. Assign the pointers in the structures to point to the right place, as described by the printout.
2. Print the structures so the result is indeed similar to the printout.

To put things in perspective: The part you need to add is about:

6 lines of 'assignments'. (something like "ptr=qw1;")

6 lines of 'printing'. (something like "printf(...);")

=== END ===