Assignment HW3
Due date (on or before): Announced in class.

1. Textbook, page 88, Exercise 3.11 (5 points)

```
a. if ( age >= 65 );
     printf("Age is greater than or equal to 65\n");
   else
      printf("Age is less than 65\n");
```

Ans: The semicolon on the if statement.

```
b. int x = 1, total;

   while ( x <= 10 ){
     total += x;
     ++x;
   }
```

Ans: total was not being initialized.

```
c. While ( x <= 100 )
     total += x;
     ++x;
```

Ans: capital W, and need braces. The correct code should be:

```
int x;
while ( x <= 100 ){
  total += x;
  ++x;
}
```

```
d. while ( y > 0 ){
     printf ("%d\n", y);
     ++y;
   }
```

Ans: it is either never start the loop (y <= 0) or execute the loop infinitely.

2. Textbook, page 133, Exercise 4.6

State which values of the control variable x are printed by each of the following for statements:

```
e. for (x = 2; x <= 13; x += 2){
```

```
        printf("%d ", x);
    }
```
Ans: 2 4 6 8 10 12

```
f. for (x = 5; x <= 22; x += 7){
        printf ("%d ", x);
    }
```
Ans: 5 12 19

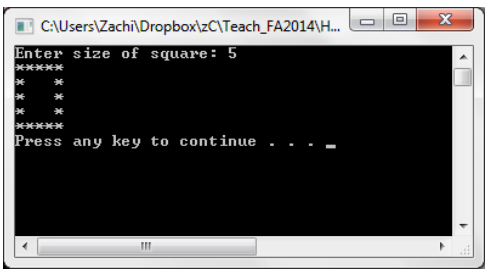
```
g. for (x = 3; x <= 15; x += 3){
        printf ("%d ", x);
    }
```
Ans: 3 6 9 12 15

```
h. for ( x = 1; x <= 5; x += 7 ){
        printf ("%d ", x);
    }
```
Ans: 1

```
i. for ( x = 12; x >= 2; x -= 3 ){
        printf ("%d ", x);
    }
```
Ans: 12 9 6 3

3. Textbook, page 94, Exercise 3.34

(Hollow Square of Asterisks) Modify the program you wrote in Exercise 3.3 so that it prints a hollow square. For example, if your program reads a size of 5, it should print:



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

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
```

```
        int size;
        int row;
        int column;

        printf("Enter size of square: ");
        scanf("%d", &size);

        for (row = 1; row <= size; row++){
                for (column = 1; column <= size; column++){
                        if (  (row == 1 || row == size) ||  (column == 1 || column == size)
)
                                printf ("*");
                        else
                                printf (" ");
                }               // end loop column
                printf("\n");
        }               // end loop row

        system("pause");
        return(0);
}
```

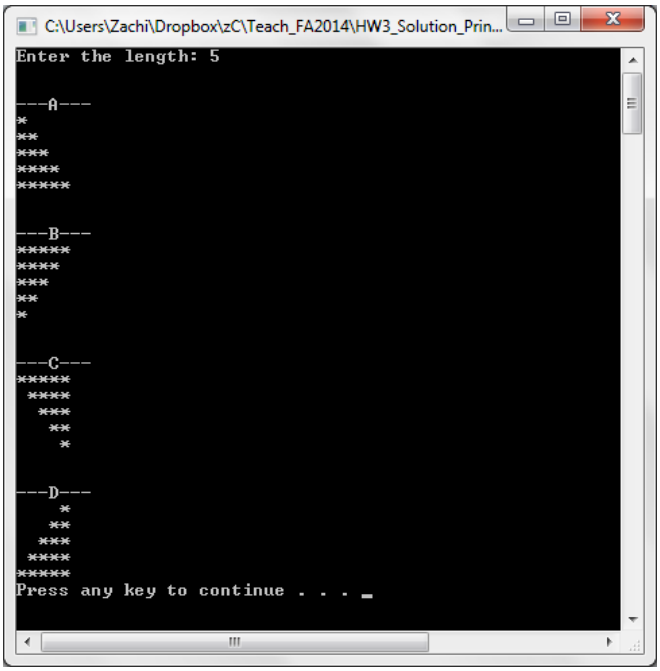
********** End of code + Screen shot**********

4. Textbook, page 135, Exercise 4.16

4.16 *(Triangle Printing Program)* Write a program that prints the following patterns separately, one below the other. Use `for` loops to generate the patterns. All asterisks (`*`) should be printed by a single `printf` statement of the form `printf( "*" );` (this causes the asterisks to print side by side). [*Hint:* The last two patterns require that each line begin with an appropriate number of blanks.]

```
(A)         (B)          (C)          (D)
*           **********    **********          *
**          *********      *********         **
***         ********        ********        ***
****        *******          *******       ****
*****       ******            ******      *****
******      *****              *****     ******
*******     ****                ****    *******
********    ***                  ***   ********
*********   **                    **  *********
**********  *                      *  **********
```

Important: Write the program so it prompts the user for the size (number of rows), and then prints these. See screen shot below.



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

```c
#include <stdio.h>
#include <stdlib.h>


int main()
{
	int len;
	int row, column;

	// Prompt user for input
	printf("Enter the length: ");
	scanf ("%d", &len);

	printf("\n\n---A---\n");
	for (row = 1; row <= len; row++){
		for (column = 1; column <= row; column++){
			printf("*");
		}
		printf("\n");		// new row
	}


	printf("\n\n---B---\n");
	for (row = len; row >= 1; row--){
		for (column = row; column >= 1; column--){
			printf("*");
		}
		printf("\n");		// new row
	}

	printf("\n\n---C---\n");
	for (row = 1; row <= len; row++){
		for (column = 1; column < row; column++){
```

```
                printf(" ");
            }
            for (column = row; column <= len; column++){
                printf("*");
            }
            printf("\n");        // new row
        }

        printf("\n\n---D---\n");
        for (row = 1; row <= len; row++){
            for (column = row; column < len; column++){
                printf(" ");
            }
            for (column = len-row; column < len; column++){
                printf("*");
            }
            printf("\n");        // new row
        }

        system("pause");
        return 0;
}
```

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

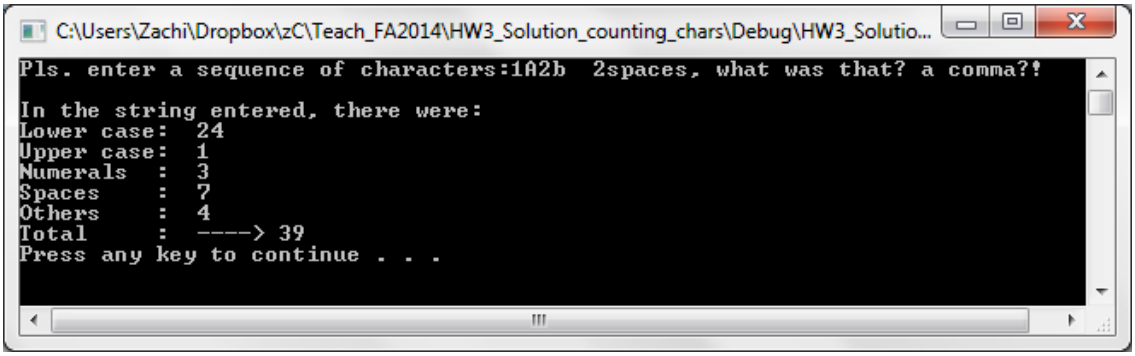
(You need to at least ONE of the challenge questions. No extra credit for doing both, other than bragging rights and learning!)

5. Challenge question (1/2):

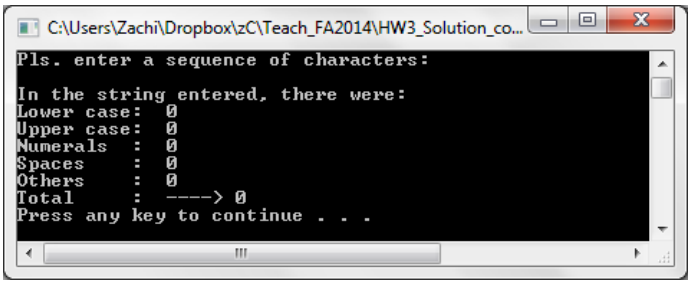
Write a program to prompt the user for a sequence of characters, and then counts how many of the following are in the sequence:

1. Total number of characters
2. Lower case letters.
3. Upper case letters.
4. Numerals (0…9)
5. White spaces
6. Others (not including New line).

The sum of 2-6 should be equal to item 1 of course.



And of course also:



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

```c
#include <stdio.h>
#include <stdlib.h>


int  main(void)
{

	int c;
	int total=0,
		lower=0,
		upper=0,
		nums=0,
		space=0,
		others=0;

	printf("Pls. enter a sequence of characters:");


	c = getchar();

	while (c != '\n')
	{
		total++;
		if (c>='a' && c<='z') lower++;
		if (c>='A' && c<='Z') upper++;
		if (c>='0' && c<='9') nums++;
		if (c==' ') space++;

		c = getchar();
	}
	others = total - lower-upper-nums-space;
```

```
        printf("\nIn the string entered, there were:\n");
        printf("Lower case:  %d\n",lower);
        printf("Upper case:  %d\n",upper);
        printf("Numerals  :  %d\n",nums);
        printf("Spaces    :  %d\n",space);
        printf("Others    :  %d\n",others);
        printf("Total     :  ----> %d\n",total);


        system("pause");
        return(0);
}
```

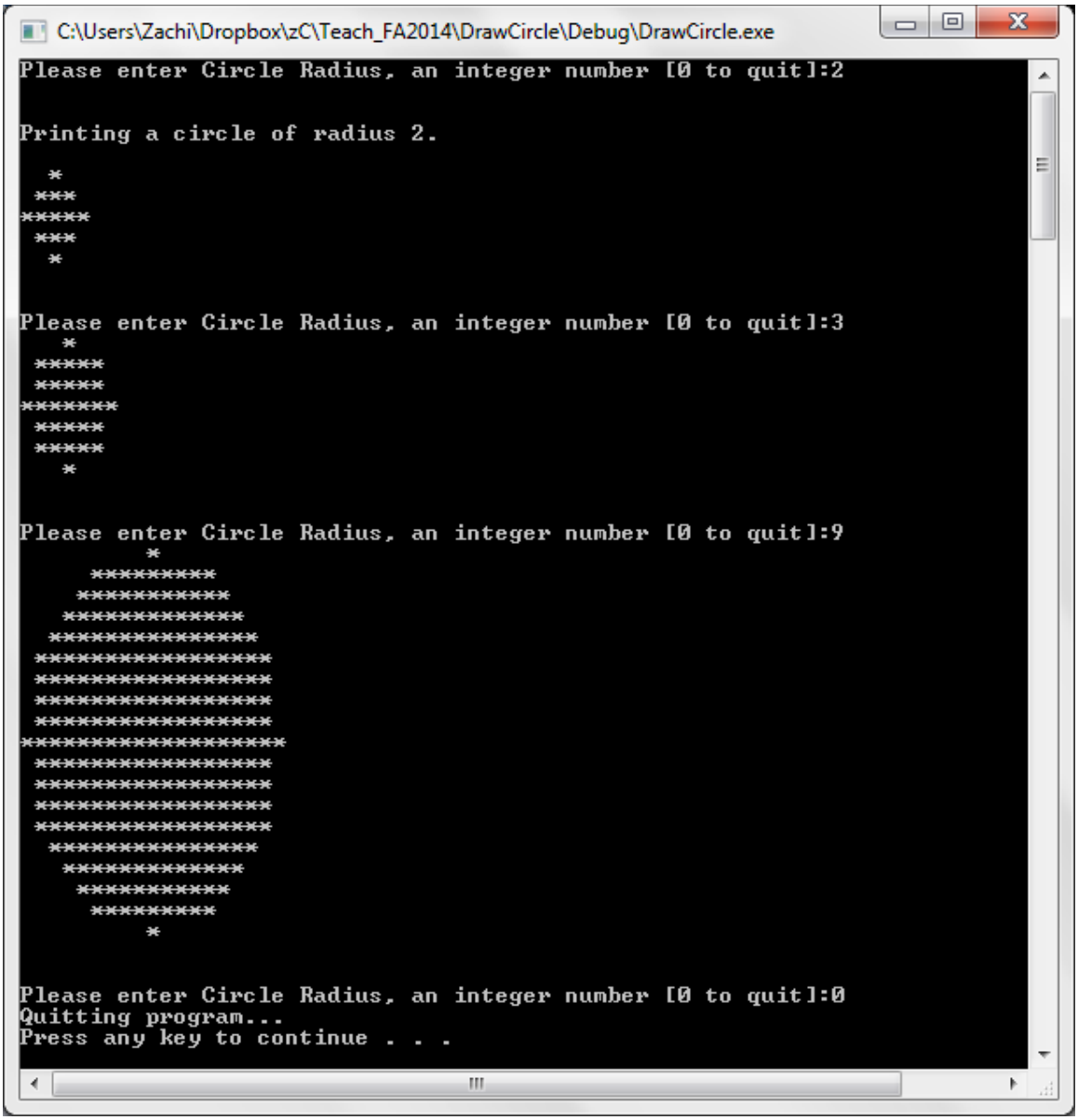
********** End of code + Screen shot**********

6. Challenge question (2/2):

Write a program to draw a circle of "*". See example below.
The program will ask the user for the radius of the circle, an integer number. The program will then draw the circle, and prompt the user to input new radius value.
The process will stop when the user inputs a radius of 0 (if entered at the first time, no circle is drawn).

What is a circle?
If the center of the circle is at Row0 and Col0 (you decide what these are), you should draw an "*" symbol at location Row and Col if the distance from (Row0,Col0) is less or equal to the given radius. In formaulas:
If (sqrt($(Row-Row0)^2 + (Col-Col0)^2$) <= Rad) print "*"

See screenshot below, with example for Rad=2, Rad=3, and Rad=9.


C:\Users\Zachi\Dropbox\zC\Teach_FA2014\DrawCircle\Debug\DrawCircle.exe
Please enter Circle Radius, an integer number [0 to quit]:2
Printing a circle of radius 2.
Please enter Circle Radius, an integer number [0 to quit]:3
Please enter Circle Radius, an integer number [0 to quit]:9
Please enter Circle Radius, an integer number [0 to quit]:0
Quitting program...
Press any key to continue . . .

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

```
#include "stdio.h"
#include "stdlib.h"

int main(void)
{
  int rad;
  int rr,cc;

  int dist2, rad2 ;

  printf("Please enter Circle Radius, an integer number [0 to quit]:");
  scanf("%d",&rad);

  printf("\n\nPrinting a circle of radius %d.\n\n",rad);

  while (rad>0)
  {
       rad2 = rad*rad ;

       for (rr=0; rr <= 2*rad ; ++rr)
       {
              for (cc=0; cc <=2*rad ; ++cc)
              {
                     dist2 = (rr-rad)*(rr-rad) +  (cc-rad)*(cc-rad);
                     if (dist2 <= rad2)
                            printf("*");
                     else
                            printf(" ");
              }
              printf("\n");
       }

       printf("\n\n");

       printf("Please enter Circle Radius, an integer number [0 to quit]:");
       scanf("%d",&rad);
  }

  printf("Quitting program...\n");
  system("pause");

  return 0;
}
```

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