Project 3: Simple Display Program

Handed out Wednesday, November 13.
Due Monday, November 25, Midnight.

Project Description:
This program will display a sequence of simple repeated figures with incremental sizes based on the following figure types:

- 'S': solid squares of asterisks (*)
- 's': hollow squares of asterisks (*)
- 'T': solid upper right triangles of asterisks (*)
- 't': hollow upper right triangles of asterisks (*)
- 'q' or 'Q': quit the program

The input required to display each of these figures are:

- Minimum dimension (integer ≥ 2): width for square, width and height are equal for upper right triangles.
- Incremental dimension (integer ≥ 1).
- Number of figures to be drawn (integer ≥ 1).

Error checking must be done for above input for positive integers.

For example, given a figure type of 'S', a minimum dimension of 3, an incremental dimension of 2, and a number of figures of 4, the output would consist of a sequence of solid squares of asterisks of sizes 3, 5, 7, 9, drawn one after the other.

Write a C program to perform the above tasks and display the corresponding figures. The program should include C functions to display the figures. The program will be in a loop until user enters a 'q' or 'Q' character to quit the program.

Project Grading:

- All projects are to be done independently. Instances of academic dishonesty will be dealt with harshly, and usually result in a grade of F.
- Late project is subject to the following late penalties: subtract 20% each day after the due date (Saturday, Sunday and holidays count as days).
- Please write project according to the C Coding Standards and Indentation Style as posted in our class website, http://www.cs.umbc.edu/courses/undergraduate/104/fall02/chen/projs/.
To submit your project, type “submit cs104-0301 proj3 filename”. See above website for more information.

Your project will be graded based on the following:

- Correctness: 80 points
  - solid squares and hollow squares: 15 points each
  - solid triangles and hollow triangles: 20 points
  - Quit case and continuous input: 10 points
  - Program does not compile: 0 points
  - No error checking: subtract 10 points
  - No C functions: subtract up to 70 points

- Good Structure and Documentation: 20 points

Here is a sample of comment blocks required for this project:

```c
/******************************/
** Course: CMSC104 Section 0301
** Project: proj3.c
** Date: 11/25/2002
** Author: Li-Chuan Chen
** SSN: 123-45-67689
** E-mail: lichen@umbc.edu
**
** Description:
**
** Input:
**
** Output:
**
** Pseudocode:
**
*******************************/
```

Here is a sample output run:

1. Run the program.
   
   linux2% a.out
   
   This program will display the following figures:

   Enter one of the following figure types:
   'S': solid squares
   's': hollow squares
   'T': solid triangles
Enter one of the following figure types:
'S': solid squares
's': hollow squares
'T': solid triangles
't': hollow triangles
'q' or 'Q': quit the program.

S

Display solid squares ...
Enter width in integer (>= 2):
-5
width must be >= 2. Enter width in integer (>= 2):
0
width must be >= 2. Enter width in integer (>= 2):
5
Enter an incremental size in integer (>= 1):
-3
incremental size must be >= 1. Enter incremental size in integer (>= 3
Enter number of figures to be drawn (>= 1):
0
number must be >= 1. Enter number of figures in integer (>= 1):
2

*****
*****
*****
*****
*****

********
********
********
********
********
********
********

Enter one of the following figure types:
'S': solid squares
‘s’: hollow squares
‘T’: solid triangles
‘t’: hollow triangles
‘q’ or ‘Q’: quit the program.

Display hollow squares ...
Enter width in integer (>= 2):
5
Enter an incremental size in integer (>= 1):
3
Enter number of figures to be drawn:
2
******
* *
* *
* *
******

*********
* *
* *
* *
* *
* *
*********

Enter one of the following figure types :
‘S’: solid squares
‘s’: hollow squares
‘T’: solid triangles
‘t’: hollow triangles
‘q’ or ‘Q’: quit the program.

Display solid triangles ...
Enter width in integer (>= 2):
5
Enter an incremental size in integer (>= 1):
3
Enter number of figures to be drawn:
2
******
****
***
Enter one of the following figure types:
   'S': solid squares
   's': hollow squares
   'T': solid triangles
   't': hollow triangles
   'q' or 'Q': quit the program.

Display hollow triangles ... 
Enter width in integer (>= 2): 
5
Enter an incremental size in integer (>= 1): 
3
Enter number of figures to be drawn: 
2

*****
* *
* *
**
*

Enter one of the following figure types:
   'S': solid squares
's': hollow squares
'T': solid triangles
't': hollow triangles
'q' or 'Q': quit the program.

q

Quitting the program ...

2. Run the program again to test 'Q' for quitting the program.

linux2% a.out

This program will display the following figures:

Enter one of the following figure types :
' S ': solid squares
's': hollow squares
'T': solid upper right triangles
't': hollow upper right triangles
'q' or 'Q': quit the program.

Q

Quitting the program ...
Here are the algorithms for this project:

1. Give an algorithm that will display a solid square of asterisks (*). See figure below for an example of \( w = 5 \).

```
*****
*****
*****
*****
*****
```

**Algorithm:**
Here is how to display a solid square of asterisks (*) whose height and width are both equal to an integer \( w \).

```
For i = 1 to w
   For j = 1 to w
      Display "*"
   End_For
   Display new_line
End_For
```

2. Give an algorithm that will display a hollow square of asterisks (*) whose height and width are both equal to an integer \( w \). See figure below for an example of \( w = 5 \).

```
*****
*   *
*   *
*   *
*****
```

**Algorithm:**
Here is how to display a hollow square of asterisks (*) whose height and width are both equal to an integer \( w \).

```
// Display first line
For i = 1 to w
   Display "*"
End_For
Display new_line

// Display 2nd to (last - 1) lines
For i = 1 to w - 2
   Display "*"
```
3. Give an algorithm to display a solid upper right triangle whose height and width are both equal to an integer length \( w \). See figure below for an example of \( w = 5 \).

\[
\begin{array}{c}
***** \\
**** \\
*** \\
** \\
*
\end{array}
\]

**Algorithm:**
Here is how to display a solid upper right triangle whose height and width are both equal to an integer \( w \).

\[
\text{For } i = w \text{ downto } 1 \\
\text{For } j = 1 \text{ to } i \\
\text{Display } "*" \\
\text{End_For} \\
\text{Display new_line} \\
\text{End_For}
\]

4. Give an algorithm to display a hollow upper right triangle whose height and width are both equal to an integer length \( w \). See figure below for an example of \( w = 5 \).

\[
\begin{array}{c}
***** \\
*  * \\
*  * \\
**  \\
* 
\end{array}
\]

**Algorithm:**
Here is how to display a hollow upper right triangle whose height and width are both equal to an integer \( w \).
// Display first line
For i = 1 to w
    Display "*"
End_For
Display new_line

// Display 2nd to (last - 1) lines
For i = 1 to w - 2
    Display "*"
    For j = 1 to (w - 2 - i)
        Display " \\
    End_For
    Display "*"
    Display new_line
End_For

// Display last line
Display "*"
Display new_line
Here is a sample skeleton of Project 3:

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

int readWidth(void);
int readIncSize(void);
int readNumFig(void);
void dispSolidSquare(int width);
void dispHollowSquare(int width);
void dispSolidTriangle(int width);
void dispHollowTriangle(int width);

int main( ) {
    /* Variable Declarations */
    /* Initialization */
    printf("This program will display the following figures: \n") ;
    /* Read Input: Get input from the user */
    do {
        printf("\n\nEnter one of the following figure types: \n") ;
        /* Display figures .... */
        switch ( ch ) {
            case 'S' :
                printf("\nDisplay solid squares ...\n") ;
                ...
                break ;
            case 's' :
                printf("\nDisplay hollow squares ...\n") ;
                ...
                break ;
            case 'T' :
                printf("\nDisplay solid triangles ...\n") ;
                ...
                break ;
            case 't' :
                printf("\nDisplay hollow triangles ...\n") ;
                ...
                break ;
            case 'q' : case 'Q' :
                printf("\nQuitting the program ...\n") ;
                ...
                break ;
            default :
                printf("\nIllegle input figure type ...\n") ;
                break ;
        }
    }
```

/ * Flush out whitespace */
scanf("%c", &whiteSpace) ;
} while (ch != 'q' && ch != 'Q') ;

return 0 ;
}

int readWidth(void) {
...
}

int readIncSize(void) {
...
}

int readNumFig(void) {
...
}

void dispSolidSquare (int width) {
...
}

void dispHollowSquare (int width) {
...
}

void dispSolidTriangle (int width) {
...
}

void dispHollowTriangle (int width) {
...
}