assign04a.c

Note that only the COMPUTE and OUTPUT sections do not change for part b and c.

```c
// J Hundley
// assign04a.c
// February 19, 2015
/*
a) You can use trigonometry to find the height of a building.
   Suppose you measure the angle between the line of sight
   and the horizontal line connecting the measuring point and the building.
   You can calculate the height of the building with the following formulas:
             tan(theta)=h/d        h=d/tan(theta)
   Assume that the distance to the building alone the ground is 120 meters
   and the angle measured alone the line of sight is 30 degrees plus/minus 3
   degrees.
   Ask the user to enter an angle 30 degrees plus/minus 3 degrees
   while the angle is not between the acceptable range.
   Compute and print the height of the building.
*/

#include <stdio.h>
#include <math.h>

//*****CONSTANT*****
#define MAX_ANGLE  33.0 // degrees
#define MIN_ANGLE  27.0 // degrees
#define DISTANCE  120.0 // meters
#define PI        3.14159

int main()
{
   double degrees,
          radians,
          height;  // meters

//*****INPUT*****
// get the angle between the line of sight and
// the horizontal line connecting the measuring point and the building
   do {
      printf( "Enter an angle in degrees (min=%.1f, max=%.1f): ", MIN_ANGLE, MAX_ANGLE );
      scanf( "%lf", &degrees );
   } while( degrees < MIN_ANGLE || degrees > MAX_ANGLE );

   //*****COMPUTATION*****
   // convert degrees to radians
   // find the height of the building
   radians = degrees * (PI/180);
   height  = DISTANCE * tan(radians);

   //*****OUTPUT*****
   // print the degrees and height of the building
   printf( "Using %.1f degrees, the building height is %.2f meters. \n",degrees,height );

   return 0;
}
```

assign04b.c

```c
//*****INPUT*****
// get the angle between the line of sight and
// the horizontal line connecting the measuring point and the building
   degrees = 0;
   while( degrees < MIN_ANGLE || degrees > MAX_ANGLE ){
      printf( "Enter an angle in degrees (min=%.1f, max=%.1f): ", MIN_ANGLE, MAX_ANGLE );
      scanf( "%lf", &degrees );
   }
```

```
// J Hundley
// assign04b.c
// February 19, 2015
/*
a) You can use trigonometry to find the height of a building.
   Suppose you measure the angle between the line of sight
   and the horizontal line connecting the measuring point and the building.
   You can calculate the height of the building with the following formulas:
             tan(theta)=h/d        h=d/tan(theta)
   Assume that the distance to the building alone the ground is 120 meters
   and the angle measured alone the line of sight is 30 degrees plus/minus 3
   degrees.
   Ask the user to enter the number of angles to enter.
   Ask the user to enter an angle 30 degrees plus/minus 3 degrees
   while the angle is not between the acceptable range.
   Compute and print the height of the building.
*/

#include <stdio.h>
#include <math.h>

//*****CONSTANT*****
#define MAX_ANGLE  33.0 // degrees
#define MIN_ANGLE  27.0 // degrees
#define DISTANCE  120.0 // meters
#define PI        3.14159

int main()
{
   int    numAngles, n; // counters
   double degrees,
          radians,
          height;       // meters

//*****INPUT*****
// ask the user for the number of angles to be entered
   printf( "Enter the number of angles to be entered: " );
   scanf( "%d", &numAngles );

   for ( n=0; n<numAngles; n++ ){
   // get the angle between the line of sight and
   // the horizontal line connecting the measuring point and the building
      do {
         printf("Enter an angle in degrees (min=%.1f, max=%.1f): ",MIN_ANGLE,MAX_ANGLE );
         scanf( "%lf", &degrees );
      } while( degrees < MIN_ANGLE || degrees > MAX_ANGLE );

   //*****COMPUTATION*****
   // convert degrees to radians
   // find the height of the building
      radians = degrees * (PI/180);
      height  = DISTANCE * tan(radians);

   //*****OUTPUT*****
   // print the degrees and height of the building
      printf( "Using %.1f degrees, the building height is %.2f meters. \n\n", degrees, height );
   }

   return 0;
}
```

assign04c.c

NEW

Read all instructions before beginning your work.

COMP1200-C - Assign 04
Due 11:59 pm – Thursday – February 19, 2015
Submit `assign04a.c, assign04b.c, assign04c.c` **via Canvas**

NOTE:
Your submitted file(s) MUST be spelled and cased as instructed.
[-5 points for not doing so.]

Before you start writing your program:

Read these instructions including the development plan. A development plan is a process that guides you through solving a problem and creating an algorithm. Save a copy of your **`assign03.c`** as **`assign04a.c.`** The assignment number should reflect a, b, or c also. Modify the problem description as needed to reflect assign04 requirements. Remove the statements that do not apply to the assign04 requirements.

New commands:
User input
`do..while`
`while`
`for`
data validation loop
counting loop

***Program*: assign04a.c**

The only part of `assign03.c` that is changed for `assign04a.c` is the input. Add a `do..while` loop around the `printf/scanf` used for user input. Your program should re-prompt the user when an angle value outside the range in entered.

***Program*: assign04b.c**

Save a copy of your **`assign04a.c`** as **`assign04b.c.`** Change the `do..while` loop to a `while` loop to re-prompt the user when an angle value outside the range in entered.

***Program*: assign04c.c**

Prompt the user for the number of angles to be entered. Use a `for` loop around the data validation loop. Add a comment for this new loop. You may use either a `do..while` or `while` loop to re-prompt the user when an angle value outside the range in entered. Save a copy of your **`assign04a.c`** or **`assign04b.c`** as **`assign04c.c.`** Add a comment above the new prompt and loop.

Instructions:

- ☐ See Standards for Documentation of C Programs on the Resources page on Canvas.
- ☐ Insert comments at the top and throughout each file.
  - o Include the follow comments at the beginning of this (and ALL) files.
    - `//` submitter's name, **GROUP #**
    - `//` other group members' names
    - `//` assignment number
    - `//` date you completed the assignment
    - `//` **statement(s) about collaboration**
    - `//` a short narrative about what the file does
  - o Use the algorithm given as comments throughout your program.
- ☐ Use descriptive variable names.
- ☐ Use Sample Input/Output as a guide.
- ☐ Use **Generate CSD** to ensure correct indenting.
- ☐ Represent ALL given values as constants.
- ☐ Format the angle with 1 decimal place.
- ☐ Format the building height with 2 decimal places.
- ☐ Label output using the `printf()` function in sentence form.

Grade of ZERO for files with submitter name not part of Canvas group
Type "none" if submitting alone.
Zero points for comments if no collaboration statement

-5 points for absence of any of these required comments at the top at the top of each file.

If you do not submit individually,
there will be a 5 POINTS PENALTY for not joining a group.
Groups can be 2-4 students.
DO NOT join a group unless you have worked with the other members. If you do, you will be removed from the group and given the grade of zero.

Sample Output:

assign04a.c and assign04b.c

```
Enter an angle in degrees (min=27.0, max=33.0): 23
Enter an angle in degrees (min=27.0, max=33.0): 37
Enter an angle in degrees (min=27.0, max=33.0): 31
Using 31.0 degrees, the building height is 72.10 meters.
```

assign04c.c

```
Enter the number of angles to be entered: 3
Enter an angle in degrees (min=27.0, max=33.0): 27
Using 27.0 degrees, the building height is 61.14 meters.

Enter an angle in degrees (min=27.0, max=33.0): 23
Enter an angle in degrees (min=27.0, max=33.0): 37
Enter an angle in degrees (min=27.0, max=33.0): 30
Using 30.0 degrees, the building height is 69.28 meters.

Enter an angle in degrees (min=27.0, max=33.0): 29.8
Using 29.8 degrees, the building height is 68.72 meters.
```

Submit via Canvas:

`assign04a.c`	C program file
`assign04b.c`	C program file
`assign04c.c`	C program file