```c
// J Hundley
// assign05
// March 5, 2015

/******READ A DATA FILE*****
 Read the input values from a text data file
 Compute the distance
 While not end of file read one set of input values
 count the balloon
 find the velocity
 compute the distance
 count hit and add water to amount in the pool
 keep up with the last balloon number that hit the pool
 Display the number of hits, number thrown, and percent hits.
 Display the number of the balloon that caused the pool water to reach capacity
 Display a message saying how many gallons spilled over the edge of the pool
*/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

//*****CONSTANT*****
#define BALCONY_HT  12.0      // balcony height in feet
#define G           32.0      // gravitational acceleration
#define PI           3.14159
#define POOL_RADIUS  1.0      // diameter of pool in feet
#define POOL_CENTER 35.0      // distance to the center of pool in feet
#define CAPACITY     7.0      // pool capacity in gallons

#define FILENAME    "balloonValues.txt"

int main()
{
// Problem Inputs:
   double theta,         // balloon launch angle (theta) in degrees
          velocity,      // balloon launch velocity (v) in ft/sec
          thrower_ht;    // thrower's height in feet
   int    diameter;      // diameter of balloon in inches
// Problem Outputs:
   int    numBalloons,   // number of balloons thrown
          numHits;       // number of balloons that hit the pool
// Other variables:
   double balloonVolume, // amount of water in a balloon in gallons
          poolWater,     // cumulative gallons of water in pool
          totalWater;    // cumulative gallons of water in all balloons
   double radians,       // angle in radians
          part1,part2,part3, // partial result holders
          distance;      // distance a water balloon travels in feet
   int    holdBalloonCount; // hold the last balloon that hit the pool

   FILE * filePtr;       // file pointer

//*****INPUT*****
//open input data file
   filePtr = fopen(FILENAME,"r");
// check for good file open
   if (filePtr == NULL)
      printf("File Open Error");
   else // good file open continue program
   {
```

```
   //*****INITIALIZATION*****
      numBalloons = 0;  // number of balloons thrown
      numHits     = 0;  // number of balloons that hit the pool
      poolWater   = 0;  // cumulative gallons of water in pool
      totalWater  = 0;  // cumulative gallons of water in all balloons

   //*****READ A DATA FILE*****
   // get the balloon input values from the data file on at a time
      while(fscanf(filePtr, "%lf %lf %lf %d", &theta, &velocity, &thrower_ht, &diameter) != EOF)
      {
         numBalloons = numBalloons + 1;     // count balloon
        // use balloon diameter to determine volume
         switch (diameter)
         {
            case 3: balloonVolume = 0.1;
               break;
            case 4: balloonVolume = 0.2;
               break;
            case 5: balloonVolume = 0.3;
               break;
           case 6: balloonVolume = 0.55;
               break;
            case 7: balloonVolume = 0.8;
               break;
            case 8: balloonVolume = 1.25;
               break;
            case 9: balloonVolume = 1.7;
         } // no default needed because of input validation above

      //*****COMPUTATION*****
      // compute total water
         totalWater = totalWater + balloonVolume;
      // convert degrees to radians
         radians = theta * (PI/180);
      // compute distance
         part1    = velocity*cos(radians)/G;
         part2    = velocity*sin(radians);
         part3    = 2*G*(BALCONY_HT+thrower_ht);
         distance = part1 * (part2 + sqrt(part2*part2 + part3));

      // is it a hit?
         if(distance > POOL_CENTER - POOL_RADIUS && distance < POOL_CENTER + POOL_RADIUS) // >= <=
         {                                                                               // also accepted
            poolWater = poolWater + balloonVolume; // add balloon volume
            numHits = numHits + 1;                 // count hit
            if (poolWater < CAPACITY)
               holdBalloonCount = numBalloons;     // remember the balloon
                                                   //   that hit the pool
                                                   //   before it was full
         }
      } // end while loop

   //*****OUTPUT*****
      printf("%d balloons hit the pool. \n", numHits);
      printf("%d balloons were thrown. \n", numBalloons);
      printf("%.2f%% balloons hit the pool. \n", (double)numHits/numBalloons*100);
      printf("balloon #%d filled the pool\n", holdBalloonCount);
      printf("%.2f gallons of water spilled over the edge pool.\n",poolWater-CAPACITY);
   } //end good open
   return 0;
}
```

Read all instructions before beginning your work.

COMP1200-C - Assign 05
Due midnight – Thursday – March 5, 2015
Submit `assign05.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:

Download the `balloonValues.txt` data file from the Assign05 Announcement and save in your COMP1200/assign05 folder. If you do not have folders set up for your assignment files, this is a good time to start. Your assign05.c will look in the folder where it is saved for the data file. A development plan is a process that guides you through solving a problem and creating an algorithm. Create your own algorithm and use it as comments throughout your program. Use section comments to group your statements as well as comments from your algorithm.

***Problem*:**

***Program*: assign04.c**

On a hot Saturday afternoon, you and your friends notice an empty baby swimming pool on the lawn of your apartment complex. So, why not see if you can fill it with water from water balloons thrown from your second floor balcony.

You will find out how many balloons it takes to fill the pool. To process a large number of balloon throws, you will run your program in **batch-mode** by reading the values needed to compute the distance each balloon travels as well as the diameter of each balloon from the `balloonValues.txt` data file. The file columns are **theta, velocity, thrower's height, and diameter**. The balloon diameter determines the amount of water in each balloon.

You will calculate the distance a water balloon will travel given the balloon launch angle (theta) in degrees, balloon launch velocity (v) in min/sec, and thrower's height in feet. Note that the height $\mathbf{y_0}$ is the sum of the balcony height and thrower's height.

$$d = \frac{v\cos\theta}{g}\left(v\sin\theta + \sqrt{(v\sin\theta)^2 + 2gy_0}\right)$$

You will continue "throwing" balloons while the end of the file has not been reached. After ALL the data in the file has been read, your program should display

- the total number of balloons thrown
- the number of balloons that hit the pool
- the percent of hits
- the number of the balloon that caused the pool water to reach capacity
- a message saying how many gallons spilled over the edge of the pool

***Problem Constants*:**

```
BALCONY_HT     12            // balcony height in feet
G              32            // gravitational acceleration
PI             3.14159
POOL_DIAMETER  2             // diameter of pool in feet
POOL_CENTER    35            // distance to the center of pool in feet
CAPACITY       7             // pool capacity in gallons
FILENAME       "balloonValues.txt"  // input data file
```

Balloons	
diameter inches	volume gallons
3	0.1
4	0.2
5	0.3
6	0.55
7	0.8
8	1.25
9	1.7

***Problem Inputs*:**

balloon launch angle (theta) in degrees
balloon launch velocity (v) in ft/sec
thrower's height in feet
balloon diameter

***Problem Outputs*:**

See the instructions above.

***Other variables*:**

As needed

Instructions:

- ☐ See Standards for Documentation of C Programs on the Resources page on Canvas.
- ☐ Insert comments at the top and throughout each file.

- Include the follow comments at the beginning of this (and ALL) files.
  // submitter's name, **GROUP #** **Grade of ZERO for files with submitter name not part of Canvas group**
  // other group members' names **Type "none" if submitting alone.**
  // assignment number **Zero points for comments if no collaboration statement**
  // date you completed the assignment
  // **statement(s) about collaboration**
  // a short narrative about what the file does
- Use the algorithm given as comments throughout your program.

-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.

- ☐ 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.

- ☐ Use a `while` loop to read the `balloonValues.txt` data file until the end of file has been reached.
- ☐ Think carefully about what needs to be done before the loop, in the loop and after the loop.
- ☐ Use the `switch/case` structure to determine the amount of water in the balloon for a given balloon diameter.
- ☐ Compute conversion: degrees to radians.
- ☐ Compute distance.
- ☐ Print the output with appropriate labels and decimals as shown. Print % with the percent of hits.

New commands:
`while`
`EOF`
`FILE, fopen, NULL`
file pointer
`fscanf`
`switch/case`

Sample Input/Output:

```
6 balloons hit the pool.
103 balloons were thrown.
5.83% balloons hit the pool.
Balloon #97 filled the pool.
0.50 gallons of water spilled over the edge of the pool.
```

Submit via* Canvas*:

`assign05.c` C program file

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

```
Rough algorithm -

INITIALIZATION
while more data, get balloon input values from the data file one balloon at a time
      count balloon
      use balloon diameter to determine volume
      add balloon volume to total water
      compute distance
      is it a hit?
            add balloon volume to pool water
            count hit
            remember balloon number that finished filling the pool
OUTPUT
```