```c
// J Hundley
// assign04b with user-defined call-by-value functions
// March 2, 2012
/* Input height and weight then compute and display BMI.
Print the classification for the computed weight.
Input a target weight then compute and display target weight.
Input gender (1=female, 2=male) then compute and display ideal weight
Validate all user enter data before using it.
height (59-78), weight (90-350), gender (1, 2), target BMI (18.5-30.0)
Do the above for one or more people.
*/
#include <stdio.h>

// FUNCTION PROTOTYPES ==============================================
// GET FUNCTIONS ====================================================
   double getInches();
   double getPounds();
   double getTargetBmi();
   char   getGender();
// CONVERSION FUNCTIONS =============================================
   double inches2meters( double inches );
   double pounds2kg( double pounds );
   double kg2pounds( double kg );
// BMI ==============================================================
   double computeBmi( double kg, double meters );
   void   displayBmi( double bmi );
// TARGET BMI, WEIGHT ===============================================
   double weightForBmi( double bmi, double meters );
   void   displayTargetWeight( double pounds );
// IBW ==============================================================
   double computeIdealWeight( char gender, double height );

    int main()
   {
      double inches, pounds,    // input
             meters, kilograms, // converted values
             bmi;               // output
      char   gender;            // female(F/f) or male(M/m)
      double ideal;             // ideal weight for gender
      int    numPeople,         // number of people
             count;             // count people
// Prompt for the number of people
   printf("Enter the number of people: ");
   scanf("%d", &numPeople);

// for each person enter and compute stats
   for (count=1; count<=numPeople; count++)
   {
   // === INPUT ================================
   // get user information within ranges
      inches = getInches();
      pounds = getPounds();

   // === BMI ==================================
   // compute converstions
      meters = inches2meters( inches );
      kilograms = pounds2kg( pounds );

   // calculate
      bmi = computeBmi( kilograms, meters );
   // display BMI and classification
      displayBmi( bmi );

   // === Target BMI and weight ==============
      bmi = getTargetBmi();
   // Compute weight
      kilograms = weightForBmi( bmi, meters );
   // Compute conversions
      pounds = kg2pounds( kilograms );
   // Display weight
      displayTargetWeight( pounds );

   // === IBW ==================================
      gender = getGender();
   // compute the IBW for the given height and gender
      ideal = computeIdealWeight( gender, inches );
   // Compute conversions
      pounds = kg2pounds( ideal );
   // display IBW
      printf("\nThe ideal weight is %.2f pounds.\n", pounds);

   }  // end for each person loop
   return 0;
}
```

```
// FUNCTION PROTOTYPES =========================================================
// GET FUNCTIONS ===============================================================
// get the inches
    double getInches()
   {
      double inches;
   // While not a good height, prompt the user to enter a value for height in inches
      do
      {
         printf("Enter the height in inches(59-78): ");
         scanf("%lf", &inches);
      } while ( inches < 59.0 || inches > 78.0 );
      return inches;
   }
// get the pounds
    double getPounds()
   {
      double pounds;
     // While not a good weight, prompt the user to enter a value for weight in poundss
      do
      {
         printf("Enter the weight in pounds(90-350): ");
         scanf("%lf", &pounds);
      }while ( pounds < 90.0 || pounds > 350.0 );
      return pounds;
   }
// get target BMI
    double getTargetBmi()
   {
      double bmi;
     // While not a good bmi, prompt the user to enter a value for BMI.
      do
      {
         printf("Enter the target BMI(18.5-30.0): ");
         scanf("%lf", &bmi);
      }while ( bmi < 18.5 || bmi > 30.0 );
      return bmi;
   }
// get gender
    char getGender()
   {
      char gender;
      // While not a good gender, prompt user to enter the gender (F/f or M/m).
      do
      {
         printf("Is the person a female or male? Enter F or M: ");
         scanf(" %c", &gender);
      }while(!(gender=='m' || gender=='M' || gender=='f' || gender=='F'));
      return gender;
   }
// CONVERSION FUNCTIONS ========================================================
// convert inches to meters
    double inches2meters( double inches )
   {
      return inches * 0.0254;
   }
// convert pounds to kilograms
    double pounds2kg( double pounds )
   {
      double kilograms;
      kilograms = pounds / 2.2046;
      return kilograms;
   }
// convert kilograms to pounds
    double kg2pounds( double kg )
   {
      return 2.2046 * kg;
   }
// BMI ==========================================================================
// compute BMI
    double computeBmi( double kg, double meters )
   {
      return kg /(meters * meters);
   }
   // display BMI and classification
    void   displayBmi( double bmi )
   {
      printf("\nThe BMI is: %.2f\n", bmi);
      printf("BMI Classification: ");
      if (bmi < 25)
         printf("Normal\n\n");
      else if (bmi >= 30)
         printf("Obese\n\n");
      else
         printf("Overweight\n\n");
   }
// TARGET BMI, WEIGHT ===========================================================
// compute the target weight
    double weightForBmi( double bmi, double meters )
   {
      double kilograms;
      kilograms = bmi * meters * meters;
      return kilograms;
   }
// compute the IBW for the given height and gender
    double computeIdealWeight( char gender, double height )
   {
      double ideal;
      if( gender=='F' || gender=='f')
         ideal = 45.5 + 2.3 * (height-60);
      else
         ideal = 50.0 + 2.3 * (height-60);
      return ideal;
   }
// display the target weight
    void displayTargetWeight( double pounds )
   {
      printf("\nThe ideal weight is %.2f pounds.\n\n", pounds);
   }
```

```c
// J Hundley
// assign04b with user-defined call-by-value functions
// March 2, 2012
/* Input height and weight then compute and display BMI.
Print the classification for the computed weight.
Input a target weight then compute and display target weight.
Input gender (1=female, 2=male) then compute and display ideal weight
Validate all user enter data before using it.
height (59-78), weight (90-350), gender (1, 2), target BMI (18.5-30.0)
Do the above for one or more people.
*/
#include <stdio.h>

// FUNCTION PROTOTYPES ==============================================
// THE GET FUNCTIONS ================================================
   void   getHealthStats( double *inches, double *pounds, char *gender );
   double getInches();
   double getPounds();
   char   getGender();
   double getTargetBmi();
// THE CONVERSION FUNCTIONS =========================================
   double inches2meters( double inches );
   double pounds2kg( double pounds );
   double kg2pounds( double kg );
// BMI ==============================================================
   void   computeDisplayBmi( double inches, double pounds );
   double computeBmi( double kg, double meters );
   void   displayBmi( double bmi );
// TARGET WEIGHT ====================================================
   void   displayTargetBmiWeight( double bmi, double pounds );
   double weightForBmi( double bmi, double meters );
   void   targetWeightForBmi( double inches );
// IBW ==============================================================
   void   computeDisplayIbw( char gender, double inches);
   double computeIdealWeight( char gender, double height );
```

```
 int main()
{
   double inches, pounds;    // input
   char   gender;            // female(F/f) or male(M/m)
   int    numPeople,         // number of people
          count;             // count people

// Prompt for the number of people
   printf("Enter the number of people: ");
   scanf("%d", &numPeople);

// for each person enter and compute stats
   for (count=1; count<=numPeople; count++)
   {
   // === GET =================================
   // get user health information within ranges
      getHealthStats( &inches, &pounds, &gender );

   // === BMI =================================
      computeDisplayBmi( inches, pounds );

   // === Target BMI and weight ===============
      targetWeightForBmi( inches );

   // === IBW =================================
      computeDisplayIbw(gender, inches);

   }  // end for each person loop
   return 0;
}
```

```
// FUNCTION PROTOTYPES ==========================================================
// GET FUNCTIONS WITH DATA VALIDATION ===========================================
// get health stats for a person
    void  getHealthStats( double *inches, double *pounds, char *gender )
   {
      *inches = getInches();
      *pounds = getPounds();
      *gender = getGender();
   }
// get the inches
    double getInches()
   {
      double inches;
   // While not a good height, prompt the user to enter a value for height in inches
      do
      {
         printf("Enter the height in inches(59-78): ");
         scanf("%lf", &inches);
      } while ( inches < 59.0 || inches > 78.0 );
      return inches;
   }
// get the pounds
    double getPounds()
   {
      double pounds;
     // While not a good weight, prompt the user to enter a value for weight in poundss
      do
      {
         printf("Enter the weight in pounds(90-350): ");
         scanf("%lf", &pounds);
      }while ( pounds < 90.0 || pounds > 350.0 );
      return pounds;
   }
// get gender
    char getGender()
   {
      char gender;
      // While not a good gender, prompt user to enter the gender (F/f or M/m).
      do
      {
         printf("Is the person a female or male? Enter F or M: ");
         scanf(" %c", &gender);
      }while(!(gender=='m' || gender=='M' || gender=='f' || gender=='F'));
      return gender;
   }
```

```
// get target BMI
    double getTargetBmi()
   {
      double bmi;
     // While not a good bmi, prompt the user to enter a value for BMI.
      do
      {
         printf("Enter the target BMI(18.5-30.0): ");
         scanf("%lf", &bmi);
      }while ( bmi < 18.5 || bmi > 30.0 );
      return bmi;
   }
// CONVERSION FUNCTIONS ======================================================
// convert inches to meters
    double inches2meters( double inches )
   {
      return inches * 0.0254;
   }
// convert pounds to kilograms
    double pounds2kg( double pounds )
   {
      double kilograms;
      kilograms = pounds / 2.2046;
      return kilograms;
   }
// convert kilograms to pounds
    double kg2pounds( double kg )
   {
      return 2.2046 * kg;
   }
// BMI =======================================================================
// Compute and display BMI
    void computeDisplayBmi( double inches, double pounds )
   {
      double bmi;
   // calculate bmi
      bmi = computeBmi( pounds2kg( pounds ), inches2meters( inches ) );
      // display BMI and classification
      displayBmi( bmi );
   }
// compute BMI
    double computeBmi( double kg, double meters )
   {
      return kg /(meters * meters);
   }
// display BMI and classification
    void   displayBmi( double bmi )
   {
      printf("\nThe BMI is: %.2f\n", bmi);
      printf("BMI Classification: ");
      if (bmi < 25)
         printf("Normal\n\n");
      else if (bmi >= 30)
         printf("Obese\n\n");
      else
         printf("Overweight\n\n");
   }
// TARGET BMI - TARGET WEIGHT ===================================================
// compute and find the target weight for the target BMI
    void targetWeightForBmi( double inches )
   {
      double bmi, kg;
      bmi = getTargetBmi();
      // Compute weight
      kg = weightForBmi( bmi, inches2meters( inches ));
      // Display weight
      displayTargetBmiWeight( bmi, kg2pounds( kg ) );
   }
// compute the target weight
    double weightForBmi( double bmi, double meters )
   {
      double kilograms;
      kilograms = bmi * meters * meters;
      return kilograms;
   }
// display the target bmi and weight
    void displayTargetBmiWeight( double bmi, double pounds )
   {
      printf("For the target BMI %.2f, the ideal weight is %.2f pounds.\n", bmi, pounds);
   }
// IBW ==========================================================================
// compute and display IBW
    void computeDisplayIbw( char gender, double inches)
   {
      double ideal; // kg
      // compute ideal weight (kg)
      ideal = computeIdealWeight( gender, inches );
      // display IBW
      printf("The ideal weight is %.2f pounds.\n\n", kg2pounds( ideal ));
   }
// compute the IBW for the given height and gender
    double computeIdealWeight( char gender, double height )
   {
      double ideal;  // kg
      if( gender=='F' || gender=='f')
         ideal = 45.5 + 2.3 * (height-60);
      else
         ideal = 50.0 + 2.3 * (height-60);
      return ideal;
   }
```