Solutions to Quiz Section Problems

1. Consider the following code. What will the output be? How would you fix the code so that the output is correct?

```
#define GOALS             5
#define ASSISTS           8
#define POINTS            2 * GOALS + ASSISTS
#define NUM_GAMES         10
#define POINTS_PER_GAME   (double)POINTS / NUM_GAMES
int main(void) {
  printf("Points per game = %.1f\n", POINTS_PER_GAME);
}
```

Recall that constants declared using #define are replaced verbatim by their definition. Thus, POINTS_PER_GAME is expanded as:

```
(double)2 * 5 + 8 / 10   /* This equals 10.0 */
/* 2.0 * 5 + 8 / 10 = 10.0 + 0 = 10.0 */
```

The trick is to place any calculations in parentheses. In fact, some programmers place parentheses around every #define, just to be safe. For example:

```
#define GOALS             (5)
#define POINTS            (2 * GOALS + ASSISTS)
/* This second definition MUST have the parentheses. */
```

2. Which of the following evaluate to 45 / 4, which equals 11? Why is it that this/these evaluate(s) correctly?

```
(15 / 2) * (3 / 2)        /* 7 */
15 * 3 / 2 / 2            /* 11 */
3 / 2 * 15 / 2            /* 7 */
15 / 2 * 3 / 2            /* 10 */
(15 * 3) / (2 * 2)        /* 11 */
```

By not dividing too early, there is no accumulation of error. The wrong answers all multiply a truncated intermediate result by some value.

3. Given the following declarations, write the code needed to convert from inches to inches, feet, yards and miles. For example, 13 inches equals 0 miles, 0 yards, 1 foot, 1 inch.

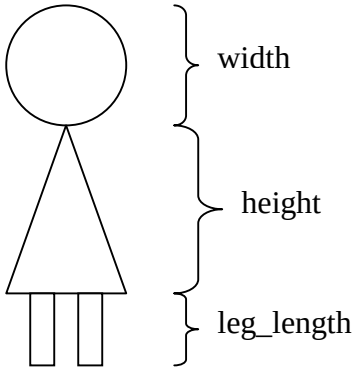
```
#define INCHES_PER_FOOT  12
#define FEET_PER_YARD    3
#define YARDS_PER_MILE   1760
int inches, feet, yards, miles;
inches = 156236;
/* Convert to feet and inches. */
feet = inches / INCHES_PER_FOOT;
inches = inches % INCHES_PER_FOOT;
/* Convert to yards, feet and inches. */
yards = feet / FEET_PER_YARD;
feet = feet % FEET_PER_YARD;
/* Convert to miles, yards, feet and inches. */
miles = yards / YARDS_PER_MILE;
yards = yards % YARDS_PER_MILE;
/* 2 miles, 819 yards, 2 feet, 8 inches */
```

4. In graphics applications, it is sometimes necessary to split the screen into (roughly) equal thirds based on a given width. These portions need to use the entire width, so the sum of the three portions must equal the original width. Which of the following will split the width into 3 portions (w1, w2, w3) that add up to w?

```
w1 = w / 3;          w2 = w / 3;          w3 = w / 3;
w1 = (w + 1) / 3;    w2 = (w + 1) / 3;    w3 = (w + 1) / 3;
w1 = (w + 0) / 3;    w2 = (w + 1) / 3;    w3 = (w + 2) / 3;
w1 = (w – 1) / 3;    w2 = (w + 0) / 3;    w3 = (w + 1) / 3;
```

Only the third approach will work correctly. The first approach is too small unless w is divisible by 3. The second approach can be too large (e.g., w = 5), or too small (e.g., w = 4). The final approach is always too small. Note that the third approach can be generalized: (w + 0) / n, (w + 1) / n, …, (w + n – 2) / n, (w + n – 1) / n.

5. Given the following figure, write the code that will calculate the area of the stick figure. Note that the width of each leg is one fifth of the width of the figure.




```
#define PI 3.14159
int width, height, leg_length;
double head_area, body_area, leg_area, total_area;
/* Assume these variables have been assigned values. */

/* Calculate the head area. */
double radius;
radius = ((double)width) / 2.0;
/* NOT: radius = (double)(width / 2); */
/* OK: radius = width / 2.0; */
head_area = PI * radius * radius;

/* Calculate the body area. */
body_area = radius * height;
/* NOT: body_area = width * height / 2; */
/* OK: body_area = width * height / 2.0; */

/* Calculate the leg area. */
leg_area = leg_length * width * 2 / 5.0;
/* NOT: leg_area = (2/5) * leg_length * width; */
/* OK: leg_area = (2.0/5.0) * leg_length * width; */

/* Calculate the total area. */
total_area = head_area + body_area + leg_area;
```

The above suggests that there are multiple ways to solve the problem, as well as numerous logical errors that could be made. You should determine why each of the incorrect solutions is incorrect.

6. What will be the output of the following (taken from Hanly and Koffman pg. 75):

```
double x;
x = -15.564;
printf("x:%8.4f\n", x);  /*-15.5640*/
printf("x:%8.3f\n", x);  /* -15.564*/
printf("x:%8.2f\n", x);  /*  -15.56*/
printf("x:%8.1f\n", x);  /*   -15.6*/
printf("x:%8.0f\n", x);  /*     -16*/
printf("x:%.2f\n", x);   /*-15.56*/
```

7. Indicate the appropriate control codes for each of the following:

```
int iq, age;
double ht, wt;
printf(“Enter your IQ: ”);
scanf(“%d”, &iq);
printf(“Enter your height: ”);
scanf(“%lf”, &ht);
printf(“Enter your age, then your weight: ”);
scanf(“%d%lf”, &age, &wt);
printf(“You are %d years old and %.0f inches tall.\n”,
          age, ht);  /* Round ht to the nearest inch. */
printf(“On average, you have grown %f inches/year.\n”,
          ht / age);  /* Display as precisely as
possible.*/
```

Be advised that the compiler will not complain if you use the wrong control code. The program will simply not work properly!