Solutions to End of Chapter 3 Problems

1. Use the array declaration below to answer the questions that follow.
   ```c
   float wins[6] = {3.4, 7, 4, 6.1, 9, 10};
   ```

   (a) How many bytes are allocated for this array?  
   6 array elements * 4 bytes each = 24 bytes

   (b) What value is stored in `wins[1]`?  
   7.0

   (c) What value is stored in `wins[6]`?  
   Undefined, garbage

2. What is the exact output of the following C program if the user enters 4 when prompted to enter a start value?

   ```c
   #include<stdio.h>
   int main ()
   {
       int start_value , number;
       printf("Enter a start value: ");
       scanf("%d" , &start_value);
       for(number = start_value; number != 12; number = number + 2)
       {
           printf("I love cyber!\n");
       }
   }
   ```

   Solution:

   ![Output Image]

   Number = 4:
   ```
   I love cyber!
   ```

   Number = 6:
   ```
   I love cyber!
   ```

   Number = 8:
   ```
   I love cyber!
   ```

   Number = 10:
   ```
   Enter a start value: 4
   I love cyber!
   ```

3. Consider again the program shown in the question above. What happens if the user enters 7 when prompted to enter a start value?

   Solution: infinite loop because `number` starts at 7, then 9, 11, 13, 15, 17, …

   `(number != 12)` will always be true

4. Let’s pretend there are 5 students in EC310. We want to write a C program that declares an array named `EC310midterm` that will hold the midterm grades for the class. The program should allow the user to enter the midterm grades at runtime, and should then print out the average of the midterm grades. Here is an example of how the program should appear:
You are provided with the source code, but there are several strategic lines of code missing.

```c
#include <stdio.h>
int main()
{
    int num_students = 5;
    float EC310midterm[num_students];
    int number;
    float sum = 0;
    float average;
    for ( number = 0; number < num_students; number = number + 1 )
    {
        printf( "Enter score for student %d : ", number + 1);
        scanf( "%f" , &EC310midterm[number] );
        sum = sum + EC310midterm[number];
    }
    average = sum / num_students;
    printf( "Class midterm average: %f \n" , average );
}
```

Solution: highlighted above.

5. In the program presented in Problem 4 above, if the line of code:

```c
EC310midterm[5] = 100;
```

was added to the end of the existing code as shown below, would the program still compile? If so, describe what we might expect the effect to be.

```c
#include <stdio.h>
int main()
{
    int num_students = 5;
    float EC310midterm[num_students];
    int number;
    float sum = 0;
    float average;
    for ( number = 0; number < num_students; number = number + 1 )
```
{  
    printf( "Enter score for student %d : ", number + 1);
    scanf( "%f" , &EC310midterm[ number ] );
    sum = sum + EC310midterm[ number ];
}

average = sum / num_students;
printf( "Class midterm average: %f \n" , average );

EC310midterm[5] = 100;
}

Solution: Yes, it would compile. It would have no effect on the program output, as it was added to the end of the code, following all output.

Because it writes out-of-bounds, it could possibly produce a segmentation fault or other unexpected behavior.

6. Given the string declaration below, mark each strcpy() function call as Safe (S) if the string literal can be safely stored within the bounds of the array, or Unsafe (U) if the string literal cannot be stored safely in the array. The array is declared as:

    char President[8];

    (a) strcpy( President , "Monroe\n");   — S (7 characters + Null) —
    (b) strcpy( President , "Polk\t");    — S (5 characters + Null) —
    (c) strcpy( President , "Cleveland\n"); — U (10 characters + Null) —
    (d) strcpy( President , "Garfield");   — U (8 characters + Null) —