1. (10 points) Where possible, write the equivalent for the following statements using compound assignment operators. If it is not possible to rewrite using compound assignment operators, say so.

(a) \( x = x + 2; \)
(b) \( z = z + r \times m; \)
(c) \( m = m \times y + 1; \)
(d) \( x = x - (a + b - c); \)
(e) \( \text{total} = 5 \times \text{total}; \)

2. (10 points) What is displayed by the following code fragment when the user inputs the value 16?

```c
scanf("%d", &n);
ev = 1;
while(ev <= n){
    printf("%d
", ev);
    ev += n % ev + 2;
}
```

Answer Box:
3. (10 points) In class we saw how to use a for loop to compute the product of all numbers from 1 to 100. Take that for loop and convert it so that it computes the product of all even numbers from 1 to 100.

Answer Box:

4. (10 points)

Correct the syntax and logic of the following code fragments.

(a) This fragment is supposed to print all numbers starting at 5 and counting down to 1.

    do
    count = 5;
    printf("%d\n", count);
    count = count - 1;
    while count > 0;

(b) This fragment is supposed to print all multiples of 5 from 0 to 100.

    for sum = 0;
    sum < 100;
    sum += 5;
    printf("%d\n", sum);
5. (10 points)

Write a function called \texttt{sum\_range} that takes two arguments \(x\) and \(y\). This function will return the sum of all integers between \(x\) and \(y\). You must write this function using either a \texttt{for} loop or a \texttt{while} loop.

\begin{answerbox}

\end{answerbox}
6. (10 points)
Write a program fragment that first asks the user to enter an integer value and store it in a variable called \texttt{base}. Then write a \texttt{do-while} loop that keeps asking the user to enter another value until the user enters a value that is a multiple of \texttt{base}.

\textbf{Answer Box:}

7. (10 points)
Write a program that asks the user to enter a number, and then displays the multiplication table for all numbers from 0 to the number they entered. This should be done with nested for loops. For example, if the user enters 3, they should see:

\begin{verbatim}
0 0 0 0
0 1 2 3
0 2 4 6
0 3 6 9
\end{verbatim}

8. (10 points)
Once again, we want to find the complement of a DNA molecule! However, this time the DNA molecule is \textbf{huge}. The number of nucleotides may also be unknown. For this reason, you will have to implement this with a loop that reads until the end of the file. After reading from "DNA.dat" and printing the complement out to "DNA\_complement.dat", you should also print the number of nucleotides that you have read from the file. To test your program, you can download HIV.dat from the examples from class website. \textit{Note: Because of the size of the files, you will also have to take care of two more cases besides ‘A’, ‘C’, ‘G’, and ‘T’: These are ‘ ’ and ‘\n’. These two cases should be ignored. This means that they should not count towards the number of nucleotides, as well.}
9. (10 points)
Write a program that determines how long it will take a town's population to reach a certain number. Your program will ask the user for two values - a starting population and an ending population. Assuming that the population increases by 10 percent each year, your program should use a loop to determine how many years it will take for the population to surpass the specified ending population. Output this result to the user.

10. (10 points)
Write a program to display a Celsius to Fahrenheit conversion table. Ask the user to enter two values - the bottom and top of a range. Your program will then display the conversion of all temperatures between those two values that are multiples of 10. The conversion should be done in a function called \texttt{fahrenheit}. For example, if the user enters 3 and 44, your program should display the following:

\begin{tabular}{|c|c|}
\hline
Celsius & Fahrenheit \\
\hline
10 & 50 \\
20 & 68 \\
30 & 86 \\
40 & 104 \\
\hline
\end{tabular}

EC I have written a small program, number.o, that is an executable. Thus, you cannot know what is inside of this program. Directions for this problem will be given in class.
<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>EC</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>