1. Suppose we want to write a function \texttt{randomInteger} that takes two integer parameters and returns a random integer that is between them. A first attempt to write the function definition for \texttt{randomInteger} might be:

```c
int randomInteger(int rangeStart, int rangeEnd) {
    int temp;
    temp = rand() % (rangeEnd - rangeStart + 1) + rangeStart;
    return temp;
} // end randomInteger
```

A \textit{driver} program is a program that tests a function by simple calling it. The following driver program calls the \texttt{randomInteger} function ten times so we get a feel for how random the numbers generated are.

```c
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

// Function prototype
int randomInteger(int rangeStart, int rangeEnd);

int main() {
    int num1, num2, myRandomInt;
    cout << "Enter two integers: ";
    cin >> num1 >> num2;
    for (int count=1; count < 10; count++) {
        myRandomInt = randomInteger(num1, num2);
        cout << "Random # = " << myRandomInt << endl;
    } // end for
} // end main
```

a) If we run the driver program and enter 15 and 10 for \texttt{num1} and \texttt{num2}, respectively, then the driver program produces the output shown. How could we fix the problem?

```c
int randomInteger(int rangeStart, int rangeEnd) {
    int temp;
    temp = rand() % (rangeEnd - rangeStart + 1) + rangeStart;
    return temp;
} // end randomInteger
```
b) If we run the driver program several times and always enter 10 and 15 for num1 and num2, respectively, then the driver program always produces the same output. Why do we always get the same pattern of random numbers when we run the program?

<table>
<thead>
<tr>
<th>Enter two integers: 10 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random # = 15</td>
</tr>
<tr>
<td>Random # = 15</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 15</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 10</td>
</tr>
<tr>
<td>Random # = 10</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
</tbody>
</table>

c) A first attempt to fix the above problem with `randomInteger` might be:

```c
int randomInteger(int rangeStart, int rangeEnd) {
    int temp;
    int seed;
    seed = time(0);
    srand(seed);
    temp = rand() % (rangeEnd - rangeStart + 1) + rangeStart;
    return temp;
}
```

However, when run the driver program produced the output shown. Explain why the output is all the same.

<table>
<thead>
<tr>
<th>Enter two integers: 10 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
<tr>
<td>Random # = 14</td>
</tr>
</tbody>
</table>

d) How did we fix this problem before?

```c
int randomInteger(int rangeStart, int rangeEnd) {
    int temp;
    int seed;
    seed = time(0);
    srand(seed);
    temp = rand() % (rangeEnd - rangeStart + 1) + rangeStart;
    return temp;
}
```

// end randomInteger