**Problem Statement:** Write a C++ program that continues to ask the user for \( n \) test scores. These test scores should range from 0 to 100, and your program needs to check that the scores supplied are valid numbers before moving forward. This may include making sure the user doesn’t enter a letter or string of letters.

- Ask the user for the number of test scores he/she wants to enter.
- Repeatedly, ask the user for that number of test scores.
  - For each test score received, check that the test score is a number and it is between 0-100.
  - Print an error message if the number is not in this range, and re-prompt the user for another number.
- After receiving good test scores, then calculate the average and output it to the screen.

**Understanding the Problem:**
This problem is asking me to read an unsigned whole number value, \( n \), from the user, and then read \( n \) unsigned real numbers, which represent test scores, from the user. These scores need to be between 0 and 100, as well as a valid real number. If the user doesn’t enter a valid number or a number in the range, then an error message is printed, and the user is prompted to enter a new number without taking away from the \( n \) valid numbers the user is entering. After the user enters \( n \) valid real numbers in the range of 0-100, then the average is calculated and printed to the screen.

I am assuming the number of tests is an unsigned whole number.
I am assuming the test scores can be unsigned real numbers, instead of just integers.
I am assuming that errors in the user input does not count against the \( n \) numbers to enter.

**Devising a Plan/Design:**

1. Ask for \( n \)
2. If \( n \) is not integer or \( n \) is not in the range of 0-100, print error and do not continue.
3. Read test score.
4. If test score is between 0 and 100, print error and do not continue.
5. Add test score to totals.
6. If test score equals \( n \), calculate average by dividing total by \( n \), and print average.
7. Add test score to totals and read another test score.
Declare n
Prompt user for n
Read n from user
While (n < 0) or (n not an int)
  Print error msg.
  Prompt user for n
  Read n

For n test scores
  Prompt user for score
  Read score
  While (score ≥ 100) or (score < 0)
    Print error msg.
    Prompt user for score
    Read score
  Add score to totalscores
  Increment test scores read

Calculate average by totalscores/n
Print test average

Testing:

<table>
<thead>
<tr>
<th>Value</th>
<th>Expected</th>
<th>Actual meet expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 0</td>
<td>Nothing, just exit</td>
<td>Yes</td>
</tr>
<tr>
<td>n = -1</td>
<td>Error message and re-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>prompt the user for a good</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n value</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>n = 1.5</td>
<td>Error message and re-prompt the user for a good n value</td>
<td>Yes</td>
</tr>
<tr>
<td>n = 1</td>
<td>Prompt user for 1 test score</td>
<td>yes</td>
</tr>
<tr>
<td>n = 5</td>
<td>Prompt user for 5 tests</td>
<td>yes</td>
</tr>
<tr>
<td>test score = -1</td>
<td>Error message and re-prompt the user for a good test value. This should not count as one of the n tests.</td>
<td>yes</td>
</tr>
<tr>
<td>test score = 100.5</td>
<td>Error message and re-prompt the user for a good test value. This should not count as one of the n tests.</td>
<td>Yes</td>
</tr>
<tr>
<td>test score = 0</td>
<td>Continue to prompt for another test score.</td>
<td>Yes</td>
</tr>
<tr>
<td>n = 1, test score = 100</td>
<td>Average should be 100</td>
<td>yes</td>
</tr>
<tr>
<td>n = 1, test score = 100.5, test score = 0</td>
<td>Error message for test score, re-prompt for a new test score, and only use valid n tests in average. Average should be 0</td>
<td>yes</td>
</tr>
<tr>
<td>n = 3, test score = 100, test score = 0, test score = 50</td>
<td>Average should be 50</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Design for next assignment:**
This should be just like the design and testing part done for the current problem, but you will not have data for actual testing. In addition, your design may be a rough design and not fully complete for this part.