Name your program **hw3.cpp**. Program must be able to compile or you will get at most 1 out 3 points for the assignment.

**Due: Wednesday October 18, 2017 by 11:59 PM**

Write **one single** complete C++ program to do all sections below:

**Part A:**

I. The program asks user to enter a positive odd integer $9 \leq \text{num} \leq 15$. If the input value is incorrect, the program repeatedly force user to input values until \text{num} is within the expected value.

II. The program then prints out two big “asterisks” with four spaces in between like below.

**Sample run with user's input shown in red:**
Enter an odd integer between 9 and 15 (inclusive): **17**
Enter an odd integer between 9 and 15 (inclusive): **7**
Enter an odd integer between 9 and 15 (inclusive): **10**
Enter an odd integer between 9 and 15 (inclusive): **11**

```
-    &    -    -    &    -
-   &   -      -   &   -
-  &  -        -  &  -
- & -          - & -
-&-            -&-
```

```
xxxxx-+++++    xxxxx-+++++
-&-            -&-
- & -          - & -
-  &  -        -  &  -
-   &   -      -   &   -
-    &    -    -    &    -
```
Part B:

Display the following compound interest table. Be sure to mention the title "Future Value of $1 After a Given Number of Periods".

Note:

\[ FV = PV \times (1 + r)^n \]

\[ PV = \text{Present Value} \]
\[ r = \text{rate of return} \]
\[ n = \text{number of periods} \]

<table>
<thead>
<tr>
<th>Periods</th>
<th>1%</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
<th>6%</th>
<th>7%</th>
<th>8%</th>
<th>9%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0100</td>
<td>1.0200</td>
<td>1.0300</td>
<td>1.0400</td>
<td>1.0500</td>
<td>1.0600</td>
<td>1.0700</td>
<td>1.0800</td>
<td>1.0900</td>
<td>1.1000</td>
</tr>
<tr>
<td>2</td>
<td>1.0201</td>
<td>1.0404</td>
<td>1.0609</td>
<td>1.0816</td>
<td>1.1025</td>
<td>1.1236</td>
<td>1.1449</td>
<td>1.1664</td>
<td>1.1881</td>
<td>1.2100</td>
</tr>
<tr>
<td>3</td>
<td>1.0303</td>
<td>1.0612</td>
<td>1.0927</td>
<td>1.1249</td>
<td>1.1576</td>
<td>1.1910</td>
<td>1.2250</td>
<td>1.2597</td>
<td>1.2950</td>
<td>1.3310</td>
</tr>
<tr>
<td>4</td>
<td>1.0406</td>
<td>1.0824</td>
<td>1.1255</td>
<td>1.1699</td>
<td>1.2155</td>
<td>1.2625</td>
<td>1.3108</td>
<td>1.3605</td>
<td>1.4116</td>
<td>1.4641</td>
</tr>
<tr>
<td>5</td>
<td>1.0510</td>
<td>1.1041</td>
<td>1.1593</td>
<td>1.2167</td>
<td>1.2763</td>
<td>1.3382</td>
<td>1.4026</td>
<td>1.4693</td>
<td>1.5386</td>
<td>1.6105</td>
</tr>
<tr>
<td>6</td>
<td>1.0615</td>
<td>1.1262</td>
<td>1.1941</td>
<td>1.2653</td>
<td>1.3401</td>
<td>1.4185</td>
<td>1.5007</td>
<td>1.5869</td>
<td>1.6771</td>
<td>1.7716</td>
</tr>
<tr>
<td>7</td>
<td>1.0721</td>
<td>1.1487</td>
<td>1.2299</td>
<td>1.3159</td>
<td>1.4071</td>
<td>1.5036</td>
<td>1.6058</td>
<td>1.7138</td>
<td>1.8280</td>
<td>1.9487</td>
</tr>
<tr>
<td>8</td>
<td>1.0829</td>
<td>1.1717</td>
<td>1.2668</td>
<td>1.3686</td>
<td>1.4775</td>
<td>1.5938</td>
<td>1.7182</td>
<td>1.8509</td>
<td>1.9926</td>
<td>2.1436</td>
</tr>
<tr>
<td>9</td>
<td>1.0937</td>
<td>1.1951</td>
<td>1.3048</td>
<td>1.4233</td>
<td>1.5513</td>
<td>1.6895</td>
<td>1.8385</td>
<td>1.9990</td>
<td>2.1719</td>
<td>2.3579</td>
</tr>
<tr>
<td>10</td>
<td>1.1046</td>
<td>1.2190</td>
<td>1.3439</td>
<td>1.4802</td>
<td>1.6289</td>
<td>1.7908</td>
<td>1.9672</td>
<td>2.1589</td>
<td>2.3674</td>
<td>2.5937</td>
</tr>
</tbody>
</table>