Sam is a new Computer Science student and is surprised how often powers of two come up in programming languages. It seems like every time a new topic is introduced, another power of two is used. Sam decides he needs to practice his powers of two. To help him with this, he writes a program.

### Program

Write a program to prompt the user for how many powers of two to display. Next the program will display the powers of two in an easy-to-read table.

### Example

User input is **Bold**.

```
How many powers of two: 8
2^0     1
2^1     2
2^2     4
2^3     8
2^4     16
2^5     32
2^6     64
2^7     128
```

Note that there is a tab before the first column and before the second column.

### Assignment

Please:

- Start from the standard template we use for homework assignments:

  ```cpp
  /home/cs124/template.cpp
  ```

- Make sure the name “Test 2” is in your program header. Make sure your professor’s name is also in the program header.
- Run test bed with

  ```
  testBed cs124/practice23 test2.cpp
  ```

- Run style checker
## Grading

<table>
<thead>
<tr>
<th></th>
<th>Exceptional 100%</th>
<th>Good 90%</th>
<th>Acceptable 70%</th>
<th>Developing 50%</th>
<th>Missing 0%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Expressions 10%</strong></td>
<td>The expression for the equation is elegant and easy to verify</td>
<td>The expression correctly computes the equation</td>
<td>One bug exists</td>
<td>Two or more bugs exist</td>
<td>The expression is missing</td>
</tr>
<tr>
<td><strong>Modularization 20%</strong></td>
<td>Functional cohesion and loose coupling is used throughout</td>
<td>Zero syntax errors in the use of functions, but room exits for improvements in modularization</td>
<td>Data incorrectly passed between functions</td>
<td>At least one bug in the way a function is defined or called</td>
<td>All the code exists in one function</td>
</tr>
<tr>
<td><strong>Loop 40%</strong></td>
<td>The loop is both elegant and efficient</td>
<td>The loop is syntactically correct and used correctly</td>
<td>The loop is syntactically correct or is used correctly</td>
<td>Elements of the solution are present</td>
<td>No attempt was made to use a loop</td>
</tr>
<tr>
<td><strong>Output 20%</strong></td>
<td>Zero test bed errors</td>
<td>Looks the same on screen, but minor test bed errors</td>
<td>One major test bed error</td>
<td>The program compiles and elements of the solution exist</td>
<td>Program output does not resemble the problem or fails to compile</td>
</tr>
<tr>
<td><strong>Style 10%</strong></td>
<td>Well commented, meaningful variable names, effective use of blank lines</td>
<td>Zero style checker errors</td>
<td>One or two minor style checker errors</td>
<td>Code is readable, but serious style infractions</td>
<td>No evidence of the principles of elements of style in the program</td>
</tr>
</tbody>
</table>

The solution is presented at:

```
/home/cs124/tests/practice23.cpp
```