Task:

Write a C program that conforms to the requirements listed below.

Requirements:

* Part 1: Create a Boolean function to determine if a year passed to the function is a leap year and pass back a Boolean result.
* Part 2:
  + Prompt the user to enter a date. Validate that it is a number that represents a year (no characters, no floating point, no negative numbers, etc.)
  + Validate if the date is a leap year and inform the user of the result
  + Prompt the user to continue with another year.
  + Loop, prompting for more years, until the user enters the letter ‘N’ to indicate they wish to stop. Be clear in your prompt that this is the terminating condition.
* Part 3:
  + Generate a Boolean table representing all of the years from 1 to 2019. Determine which years are leap years and which are not, storing the true/false results in the Boolean table.
  + Print only the leap years, one century per line.
* General requirements:
  + Include clear comments
  + Maintain a standard layout/format for the code. Be consistent with spacing or tabbing, use the layout o make nested operation visually clear.
  + Provide clear visual feedback to the user

Evaluation:

This assignment is worth 20 marks. Please see the marking rubric below.

Assignment Notes:

The assignment must be demonstrated to the instructor on or before the due date during class.

Submission time to the provided Brightspace drop box will constitute the timestamp for evaluating any late penalty the assignment may incur.

See the **Marking Rubric** below.

| Criteria | Below Standard | Developing | Acceptable | | Professional | Marks |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | | 3 |
| Boolean function | * Function not used | * Function used but contains more than 2 errors | * Function used and almost matches specification but has an error | | * Function implemented completely to specification |  |
| Input Loop | * User input not in a loop | * Loop used but validation has more than 2 errors | * Loop works mostly as expected but still one error | | * Use input works completely to specification | \_\_\_\_\_\_ |
| Boolean Table | * Boolean table not used | * Boolean table used but contains more than 2 errors | * Boolean table used but contains more an error | | * Boolean table implemented completely to specification | \_\_\_\_\_\_ |
| Aesthetics of Output | * incorrect or non existent use of whitespace in output * output is confusing and hard to follow | * fair use of   whitespace   * most output is clear, but poorly presented | * good use of whitespace * output is clear and well presented | | * excellent use of whitespace   output is clear and attractively presented | \_\_\_\_\_\_ |
| Readability | * source code does not match specification |  | * source code is exceptionally well organized and easy to follow | |  | \_\_\_\_\_\_ |
| Comments | * comments do not match specification |  | * not over/under commented * comments are meaningful and easily understood * files and functions have headers   Code is self-documenting | |  | \_\_\_\_\_\_ |
| **Naming** Convention | * no standard naming convention followed |  | * industry standard naming convention used throughout the program | |  | \_\_\_\_\_\_ |
| Consistency | * no consistency in formatting or layout of source code |  | * source code formatting never deviated from expected layout | |  | \_\_\_\_\_\_ |
| Total | | **20** |

0 - Assignment not submitted or work not original.