EE222 Homework 5

Write a C program, that can guess any number between 1 and a maximum number by way of asking simple questions that can be answered with either 'Yes' or 'No'. The form of the questions asked is generally something like this: "Is your number less than XXX?" XXX will need to be computationally determined by your program and is part of your program's logic.

The idea of the algorithm underlying your program's logic is that you continuously divide the range of numbers that contains the number to be guessed into two (roughly) equally sized sub-ranges and then determine which part the number to be guessed lies in using the simple kind of question mentioned above. Once you know in which range the number is, you cut the corresponding interval in half again, ask the question, cut the proper interval in half, ask, cut ... you get the idea.

Your algorithm terminates when there is nothing to cut any more, which occurs exactly when the final interval has a length of 1.

I want you implement your program honoring these conventions:

- I want you to implement the maximum number that could possibly be guessed as a symbolic constant named **MAX**. For the purpose of our assignment, I ask you to set MAX to 1000.
- I want you to read the yes/no input from the user from the keyboard. We want to assume that entering the letter **y** means **Yes** and the letter **n** means **No**. Any other character is being ignored and results in the same question to be asked again.
- I want you to number the questions starting at 1. Given the original range of numbers from 1 ... 1000, you should never see more than 10 questions. For the most part, 10 questions will be asked, under rare circumstances, it may be nine questions.

Here is an example of a run of the algorithm that tries to find the number 801:

```
> ./main
1  Is  number less than 500 :n
   ====================================
2  Is  number less than 750 :n
   ====================================
3  Is  number less than 875 :y
   ====================================
4  Is  number less than 812 :y
   ====================================
5  Is  number less than 781 :n
   ====================================
6  Is  number less than 796 :n
   ====================================
7  Is  number less than 804 :y
   ====================================
8  Is  number less than 800 :n
   ====================================
9  Is  number less than 802 :y
   ====================================
10 Is  number less than 801 :n
   ====================================
Your number is  801>
```

What to submit
I ask you to submit the following:

- The (abundantly commented) source code of your program.
- Screen shots of three runs of your program, showing the output for guessing the numbers 1, 501 and 1000.

Hint

When you use scanf() or getchar() to read the single character input, will prompt and wait for you to enter something on the keyboard. You need to understand that entering a character, followed by Enter will enter actually two characters, the second of which is the unwanted carriage return. You need to get rid of the second character, as it will confuse your program logic. In order to achieve that, I suggest to call getchar() right after the call to scanf() or getchar(), which has the effect of silently discarding the unwanted character.

Carefully read the grading guide below to see what the crucial parts of the program are that you will pay attention to.

Grading guide

Your submission will be graded according to the following guidelines for a total of 80 points:

- Your submission will not earn points if it doesn't compile.
- Your submission does not earn points, if the screen shot and/or program code is missing.
- Program logic:

Aspect	Points
Correct loop that recognizes, and terminates, when the number to be guessed has been found	30
Proper computation of the length of a half-range, integer-based (so that you don't have float range boundaries)	10
Come up with the correct readjustment of range boundaries, when the response is Yes	20
Come up with the correct readjustment of range boundaries, when the response is No	20