COMP500 / ENSE501: Week 12 – Exercise:

EXERCISE NAME: *AI Mind Reader GUI*

The **`WinMain`** prototype is:

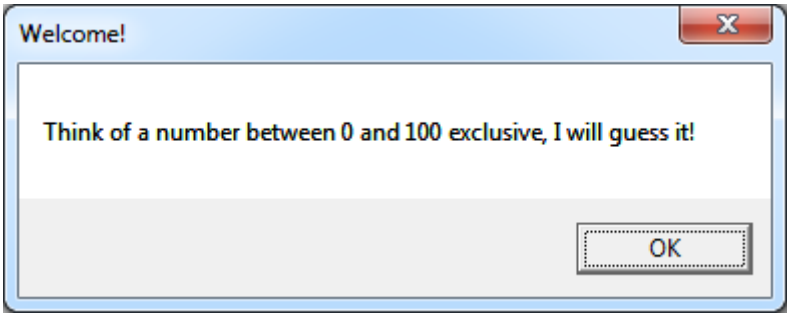
```
int WINAPI WinMain(HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPSTR lpCmdLine,
                   int nCmdShow);
```

In the "Guess the Number" game one player thinks of a number between 0 and 100 and the other player must guess it.

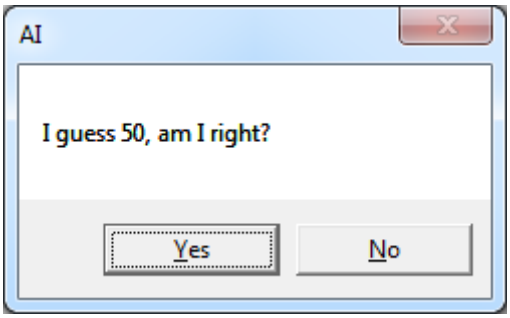
Implement this game with the computer as the player who is guessing, and the human user is the player thinking of the number, using the Windows API.

Remember, to convert a variable into a C-string you will need to use **`sprintf`** (ASCII) or **`wsprintf`** (Unicode).

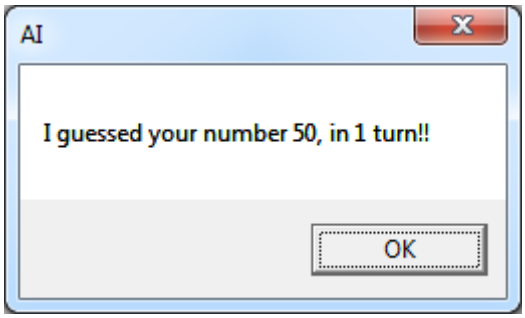
Write a program that uses the Windows API function **`MessageBox`**. At the start of the program, the following is presented to the user:



When the user clicks "Ok", the following is presented:

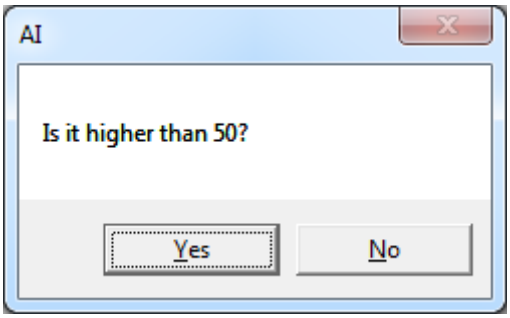


Next the user can click “Yes” or “No”. If they were thinking of 50, they would click “Yes”, and the following would be presented:

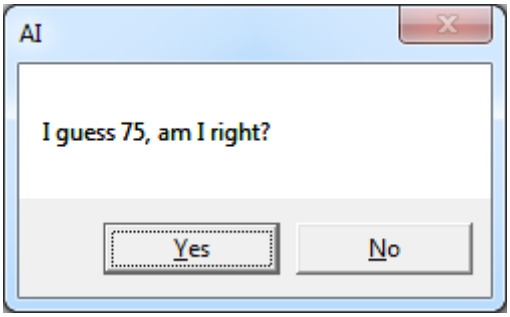


At this point, clicking “Ok” would end the program.

Otherwise, if 50 was not their secret number, they would have clicked “No”, and then been presented with the following:

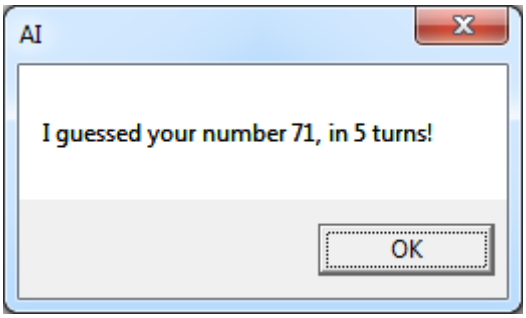


Based on their secret number, the user would again then click “Yes” or “No”, and the program would then respond in the following style:



This routine will continue until the AI guesses the user’s secret number.

Once the program successfully guesses the user's secret number it will present the following:



The program should also be able to detect if the user is cheating!

You should design your algorithm before you begin coding. You may want to also code a prototype using the console first, in this case you should refer to the previous exercise *AI Mind Reader*. Once you have the algorithm designed, then you should program it using the Windows API.

Ensure the program output is exactly as described, and that the whitespace of your source code is well formatted. Utilise good naming practices when declaring variables.

Test your program with a variety of input to ensure the implementation is robust.