The Prime Number Program Write a program to:

- Prompt the user for a candidate number (integer)
- Determine whether or not that number is prime
- Display the result
- Prompt for another number (0 to quit)

Example run #1:

```
Welcome to the wiz-bang prime number thing!

Enter an integer greater than 2 (0 to quit): 35
35 is not prime.

Enter an integer greater than 2 (0 to quit): 13
13 is prime.

Enter an integer greater than 2 (0 to quit): 0

Thanks for playing.
```

By definition, prime numbers are divisible only by 1 and themselves. That means a number can be proven to be **not** prime by finding a factor other than 1 or the number. Therefore, your program should use repetition to try consecutive divisors until either a divisor is found or all potential divisors have been tried.

Example 1, to determine that 13 is prime:

- Try dividing 13 by 2 does not come out even
- Try dividing 13 by 3 does not come out even
- ….
- Try dividing 13 by 12 does not come out even, 13 must be prime

Example 2, to determine that 35 is not prime

- Try dividing 35 by 2 does not come out even
- Try dividing 35 by 3 does not come out even
- Try dividing 35 by 4 does not come out even
- Try dividing 35 by 5 comes out even (35 ÷ 5 = 7), 35 is not prime

For **5 points extra credit**, display a factor which makes the number not prime. Example:

```
35 is not prime because it's divisible by 5.
```