# Optimus Prime

## Stage 1 – Prime Or Not

Here is some pseudocode for calculating if a number is prime.

numberToCheck 🡨 9

 isPrime 🡨 true

 FOR count 🡨 2 to Math.Floor(Math.Sqrt(numberToCheck))

 IF numberToCheck Mod count = 0 THEN isPrime=false

 END FOR

 IF isprime THEN

 OUTPUT " {0} is a prime number", numberToCheck

 ELSE

 OUTPUT "{0} is not a prime number", numberToCheck

 END IF

1. There is an operator used here called **Mod**. This is short for **modulus** find out what this means and what it tells you.
2. Use the pseudocode to create a program to check if a number is prime. You must declare the variables before you assign values to them.
3. Change the program so that the user enters the number that is being checked.

## Stage 2 – Many Primes

1. Change the program so that it outputs the number if it is prime and nothing else. Remove the ‘else’ part of the IF statement so that nothing is displayed if the number is not prime. Test.
2. Remove the lines that allow the user to enter to the number that is being checked.
3. The algorithm works by assuming that a number is prime. We try to divide the number by all of the possible factors except itself and 1. If we get no remainder on any of these calculations, we know the number if not prime. After the loop, if have not found any factors for the number, we know it must be a prime number. The algorithm begins with the line isPrime=true. We need to repeat everything from here to just after the IF statement. You need a FOR…NEXT loop that counts from 3 to 100 and has the variable name numberToCheck. Test the program.
4. The loop you added at the previous step does not include the number 2, which is prime. Change the program to output this before all of the other numbers. It must only be displayed once.

## Stage 3 – Nth Prime

1. Add a variable called **n** and set it to 1 at the start of the program. Replace the For…Next loop that keeps track of the numberToCheck with a Do…Loop. The **loop** part should say **until n=20**. Before the DO statement, you need to have declared the variable numberToCheck and have set it to 3. If a prime number is found, you need to increase n by 1 (n=n+1). Just before the **loop until n=20**, you need to add 2 to numberToCheck.
2. Change the program so that the user tells you how many prime numbers they want to see.
3. Change the program so that it outputs only the last prime number found.