Programming Problem 2: Primality testing

JAN 27 2012

Our second programming problem

Primality Testing

Given a positive integer (> 1), determine whether it is a prime number or not.

Examples:

Input	Output
31	prime
2001	composite
987654321	composite

Algorithmic Idea

- Generate all "candidate" factors of n, namely

  2, 3, …, n-1

- For each generated "candidate" factor, check if n is evenly divisible by the factor (i.e., the remainder is 0).

- If a "candidate" factor is found to be a real factor, then n is composite.

- If no "candidate" factor is found to be a real factor, then n is a prime.

Algorithm in pseudocode

1. Input n
2. For each factor = 2, 3, ..., n-1 do the following
3.   if n is evenly divisible by factor then
4.   remember that n is a composite
5. If we have detected that n is a composite
6.   output that n is a composite
7. Otherwise output that n is a prime

Python code (Version 1)

```python
number = int(raw_input("Enter a positive integer: "))

factor = 2
isPrime = True
while(factor <= number  - 1):
        if(number % factor == 0):
                isPrime = False
        factor = factor + 1

if(isPrime):
        print number, "is prime"
else:
        print number, "is composite"
```