

CSC112                      Spring 2011  
Fundamentals of Computer Science  
Practice Problem for Recursion

1. **Recursion:**

Write a recursive function to decide if a number  $n$  is a prime number. Recall a number is prime if it is only divisible by one and itself. To make the recursion work, it is necessary to keep track of the trial divisors, so there needs to be a parameter for that. We can either let the divisor count up or count down. For simplicity, let us agree to let the divisor start at 2 and count up. Then, the base case is when the divisor exceeds the square root of  $n$ . If no divisor has been found (remainder is zero), then the number is prime. If at any point in the recursion, a divisor is found, then the number is not prime. The function should return true if the number is prime, and false otherwise.

When using recursion, a wrapper function is often used to simplify the interface presented to the main program. The wrapper function initializes any extra variables and starts the first call to the recursive function.

**Your task is to:**

- (a) Write a recursive function `recursive_prime_check`. There should be no loops in your function. Use recursion instead of loops.

The header for function `recursive_prime_check` is:

```
bool recursive_prime_check( int divisor, int n )
```

- (b) Write a wrapper function `prime_check` that:

- Checks if ( $n < 2$ ). If so,  $n$  is not prime.
- Starts the top-level call to `recursive_prime_check`.

The header for function `prime_check` is:

```
bool prime_check( int n )
```

Function `prime_check` returns true if  $n$  is prime, false otherwise. The main program is then:

```
int main()
{
    int n ;

    cout << "Enter n: " ;
    cin  >> n ;
    if ( prime_check( n ) {
        cout << n << " is prime" << endl ;
    }
    else {
        cout << n << " is not prime" << endl ;
    }
}
```

**Answer:**

```
#include <iostream>
#include <cmath>      // <cmath> needed for square root.
using namespace std ;

// -----
bool recursive_prime_check( int divisor, int n )
{
    if ( divisor > sqrt(n) ) {
        return true ;
    }
    else {
        return ( ( n % divisor ) > 0 ) &&
            recursive_prime_check( divisor + 1 , n ) ;
    }
}

// -----
bool prime_check( int n )
{
    if ( n < 2 ) return false ;
    return recursive_prime_check( 2, n ) ;
}

// -----
int main()
{
    int n ;

    cout << "Enter n:  " ;
    cin  >> n ;
    if ( prime_check( n ) ) {
        cout << n << " is prime" << endl ;
    }
    else {
        cout << n << " is not prime" << endl ;
    }
}
}
```