

Problem 1

Function signatures are used to distinguish different functions in one program, so if one factor is included in the function signature, it means two functions can be distinguished just by this factor when being called, which means the factor must be given when the function is called. So if return type is included in the function signature, two functions with exactly the same other factors including names can be declared, but when the name of the function is called, the program can't determine which function should be called because return type is not a known factor when the function is called.

For example, if two functions below can be declared as having two different function signatures:

```
int f(int n);  
bool f(int n);
```

then when a function f is called in the main function, the program won't be able to tell which one it is calling.

Problem 2

(a)

Because the function calculate the answer and output it as well, we say that they are coupled because those two tasks are done once the function is called.

(b)

```
bool prime(int n)  
{  
    bool isPrime = true;  
    for(int i = 2; i * i <= n; i++)  
    {  
        if(n % i == 0)  
        {  
            isPrime = false;  
            break;  
        }  
    }  
    return isPrime;  
}
```

The output is taken out from the function, to use this implement you should call the function and output the return value.

(c)

```
bool prime(int n)  
{  
    bool isPrime = true;  
    for(int i = 2; i * i <= n; i++)  
    {  
        if(n % i == 0)  
        {  
            isPrime = false;  
            return isPrime;  
        }  
    }  
    return isPrime;  
}
```

I prefer to return the boolean value right after the value is known, because in this way I can prevent lots of potential problems that could be occurred before the return statement.