(1) Write down a proof that the following recursive factorial function is correct using proof by induction.

```
/* Factorial function definition */
int fact(int n)
{
  /* pre-condition */
  assert (n >= 1);

  /* post-condition */
  if(n > 1)
    return n * fact(n - 1);
  else
    return 1;
}
```

The purpose of the program is to compute the factorial of integers. The program consists of a base case and a recursive step. To proof that the recursive factorial function is correct, we start by looking at the base case. If n=1 then the function returns 1. We can confirm that by saying $n \cdot fact(n-1) = 1 \cdot fact(1-1) = 1 \cdot 0! = 1$, since the factorial of 0 is 1. So, the program is correct for the base case.

Now we look further for n > 1, the recursive step. We know that the absolute truth about the factorial function is in this case, the function will return $fact(k) = k \cdot fact(k-1),\ for\ k > 1$. If $fact(k-1)$ is not correct, then it doesn't compute the factorial integer. So, let's suppose that $fact(k-1)$ is correct. Then:

$$fact(k) = k + [factorial\ of\ integers\ 1\ to\ (k-1)], for\ k > 1$$

So, fact (3) is correct when fact (2) is correct. Fact (2) is correct, when fact (1) is correct and so on. But our base case was fact (1), which was correct. Therefore, fact (2) is correct and fact (3), all the way to fact (k). This proves that the program is correct for all positive integers, n.