Problem 3:

Calculate all verification conditions generated by the following annotated specification

```
{x = n}
y:=1;
        while x!= 0 do
                invariant x! ∗ y = n!
            y:=y*x;
            x:=x-1
{x = 0 ∧ y = n!}
```

Precondition {x=n}

- This specifies the initial condition where ‘x’ is equal to ‘n’

Postcondition {x = 0 ∧ y = n!}

- This specifies the final state of the programm where ‘x=0’ and ‘y=n’

The Loop variant hold true at the beginning of the Programm and after each while-loop

1. After the first while loop
- verification condition: because ‘x = n’ -> x!*y = n!*1=n!

2. Inside the while loop
- verification condition: x!*y = n! ∧ x != 0 -> (x-1)!*(y*x) = (x-1)!*n = (n-1)!*n = n!

3. After the while loop
- Verification condition: x!*y = n! ∧ x = 0 -> y =n!
- This condition is valid because it ensures that the loop terminates (‘x’ become 0) and ‘y’ is equal to ‘n!’

➔ All the conditions generate by the annotated specification are valid