Programmering for computerteknologi
Hand-in Assignment Exercises

Week 11: Verifying Correctness of Recursive Programs

Please make sure to submit your solutions **by next Monday**.

In the beginning of each question, it is described what kind of answer that you are expected to submit. If *Text and code answer* is stated, then you need to submit BOTH some argumentation/description and some code; if just (*Text answer*) or (*Code answer*) then just some argumentation/description OR code. The final answer to the answers requiring text should be **one pdf document** with one answer for each text question (or text and code question). When you hand-in, add a link to your GitHub reposetory in the beginning of your pdf file. Make sure that you have committed your code solutions to that reposetory.

Note: the **Challenge** exercises are *optional*, the others mandatory (i.e. you **have** to hand them in).

Link to repository: https://github.com/Aarhus-University-ECE/assignment-11-Teun0n

Exercises

(1) Write down a proof that the following recursive factorial function is correct using *proof by induction*. Put your inductive proof into a pdf file (`text_answers.pdf`). *Hint:* review the lecture slides for the two components of a proof by induction, i.e. (a) the base case and (b) the inductive step.

```
/* 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;
}
```

Base case:

The simplest case for our function is n=1, which gives fact(1)=1

This is true therefor we can move on to the inductive step.

Inductive step:

Now we assume for n=k where k is any integer input greater than 1, where when k is substituted for n:

$$k \cdot fact(k-1) = k!$$

This is assumed to be true, when assuming this then:

$$(k-1) \cdot fact\big((k-1)-1\big) = (k-1)!$$

Must also be true, because the output is used to find that $k \cdot fact(k-1) = k!$

This can be argued for all integers number greater than 1 in descending order until the base case is reached.

Which is has been shown to be true, meaning the output of the factorial function is equal to the factorial of the integer input n.

(2) Consider the inductive proof below. It proves that the sum of the first n positive odd numbers equals n^2, that is, $(2*1-1)+(2*2-1)+...+(2*n-1)=n^2$. Your task is to use the content of the inductive proof as inspiration to create a recursive function that calculates the sum of the first n positive odd numbers.

case and *recursive step* that you need to implement in your recursive function.

Proof by Induction

Base case:

$2*1-1=1$

Inductive step:

Assume $(2*1-1)+\cdots+(2*(n-1)-1)=(n-1)^2$

Then

$$\begin{aligned}(2*1-1)+(2*2-1)+...+(2*n-1)=\\ (n-1)^2+(2*n-1)=\\ n^2-2n+1+2*n-1=\\ n^2\end{aligned}$$

(3) Convert the following recursive program into (a) an equivalent tail recursive program, and (b) a program using a *while* loop. Add test cases for the two functions in `tests\src\tests.cpp` within `TEST_CASE("sumwhile")` and `TEST_CASE("sumtail")`

```
/* Sum integers 1 to n */
int sum(int n)
{
  /* pre-condition */
  assert (n >= 1);

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

(4) Convert the following recursive Fibonacci function into an equivalent tail recursive program.

```
int fib(int n)
{
  /* pre-condition */
  assert (n >= 1);

  /* post-condition */
  if(n == 1)
    return 1;
  else if(n == 2)
    return 1;
  else
    return fib(n - 1) + fib(n - 2);
}
```