Question 1. [20 MARKS]

In this question you have to prove that

"$10^{2n} - 1$ is divisible by 11 for all $n \geq 0, n \in \mathbb{N}$" in two different ways.

Part (a) [10 MARKS]

Use simple induction.

Basis: $n = 0$. Then 1-1= 0 which is divisible by 11.

I.H.: Suppose the theorem is true for n = k, that is, 11 divides $10^{2k} - 1$. We must prove that 11 divides $10^{2(k+1)} - 1$.

We can write $10^{2(k+1)} - 1 = 10^{2(k+1)} - 100 + 99 = 10^2 \cdot (10^{2k+1}) + 99$.

Since 11 divides both terms of this sum, it must divide $10^{2(k+1)} - 1$.

Part (b) [10 MARKS]

Use the *Well-Ordering Principle.* Do not use induction directly.

SOLN:

Let $C = \{n | 10^{2n} - 1 \neq 11k, \forall k \in \mathbb{N}\}$. Notice that $n = 0$ is not in C since 0 is divisible by 11. By WOP there exists a smallest element $a \geq 1$ in C. Notice that then $10^{2a} - 1 = 10^{2a} - 100 + 99 = 10^2(10^{2(a-1)} + 99)$. Since $0 \leq a-1 < a$, $a-1 \notin C$ so 11 divides $(10^{2(a-1)})$ and 11 divides 99, so 11 divides $10^{2a} - 1$, contradicting that $a \in C$.

Question 2. [10 MARKS]

Recall the Fibonacci Numbers:

$$F(n) = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ F(n-1) + F(n-2), & n \geq 2 \end{cases}$$

The *Lucas Numbers* are similar to the Fibonacci Numbers and are defined as

$$L(n) = \begin{cases} 2, & n = 0 \\ 1, & n = 1 \\ L(n-1) + L(n-2), & n \geq 2 \end{cases}$$

The first few *Lucas* numbers are: 2, 1, 3, 4, 7, 11, 18, ...

Prove $\forall n \geq 1, n \in \mathbb{N}$ that

$$L(n) = F(n-1) + F(n+1)$$

Soln:

(Marks) are in paranthesis.

(1) Let $S(n)$ be $L(n) = F(n-1) + F(n+1)$. RTP: $S(n)$ is true $\forall n \geq 1, n \in \mathbb{N}$.

(3) Base Case: $n = 1$. Then F(0) + F(2) = 1 = L(1)., $n = 2$, then $F(1) + F(3) = 3 = L(2)$.

(2) Induction Hypothesis: Assume that $S(k)$ is true for all $2 \leq k < n$.

(4) Induction Step: RTP $S(n)$:

$L(n) = L(n-1) + L(n-2) = F(n-2) + F(n) + F(n-3) + F(n-1)$ by I.H, since $0 \leq n-1, n-2 < n$

(2 marks for the induction hypothesis used properly. 1 mark for using the definitions of L(n) and F(n) properly. 1 for correct answer.)

$$L(n) = F(n) + F(n-1) + F(n-2) + F(n-3) = F(n+1) + F(n-1).$$

Question 3. [10 MARKS]

Suppose that there exists an algorithm with the following complexity $T(n)$:

$$T(n) = \begin{cases} a & n = 1 \text{ or } n = 0 \\ 9T(n-2) + n & n > 1 \end{cases}$$

where a is a constant. You may assume that n is even.

Use repeated substitution to find a closed form formula for $T(n)$.

Sample Solution.

Assume n is even:

$$\begin{aligned}
T(n) &= 9T(n-2) + n \\
&= 9^2T(n-4) + 9(n-2) + n \\
&= 9^iT(n-2i) + \sum_{j=0}^{j=i-1} 9^j(n-2j) \\
&= 9^{\frac{n}{2}}T(0) + \sum_{j=0}^{j=n/2-1} 9^j(n-2j) \\
&= 3^n + n\sum_{j=0}^{n/2-1} 9^j - 2\sum_{j=0}^{n/2-1} j9^j \\
&= 3^n + n(\frac{(9^{\frac{n}{2}-1} - 1)}{8}) - 2(\frac{9 - (n/2-1)9^{n/2-1} + (n/2-2)9^{n/2}}{(1-9)^2}) \\
&= 3^na + n(\frac{(\frac{3^n}{3} - 1)}{8}) - (\frac{9 - (n/2-1)\frac{3^n}{3} + (n/2-2)3^n}{32}) \\
&= 3^na + \frac{(\frac{4n3^n}{3} - 4n) - 9 + (n/2-1)\frac{3^n}{3} - (n/2-2)3^n}{32} \\
&\leq c3^n + bn3^n
\end{aligned}$$

Question 4. [10 MARKS]

The following algorithm iteratively computes $\sum_{i=0}^{n} i^3$.

Precondition: $n \in \mathbb{N}$ and $n \geq 0$.
Postcondition: Returns $\sum_{i=0}^{n} i^3$.

```
def Sum(n)
    i = 0
    sum = 0
    while (i ≤ n):
            sum = sum + i*i*i
            i = i+1
    return sum
```

Prove that `Sum` meets the postcondition, given the precondition. Make sure that you *clearly* define your loop invariant and that you show termination and total correctness.

Solns.

First we prove *partial correctness.*

LI(j): "*After the* j^{th} *iteration (if it exists),* $sum_j = \sum_{i=0}^{i=j} i^3$, $i_j = j + 1$."

We will prove by simple induction $LI(j)$ for all $j \in \mathbb{N}$.

Base Case: $j = 0$. Then by the code $sum_0 = 0 \checkmark$ and $i_0 = 1 = 0 + 1 \checkmark$.

I.H. Assume that $LI(j)$ holds for arbitrary $j \in \mathbb{N}$.

I.S. Prove that $LI(j + 1)$.

By the code $sum_{j+1} = sum_j + i_j * i_j * i_j$. Since $LI(j)$ holds, $sum_{j+1} = \sum_{i=0}^{j} i^3 + (j+1)*(j+1)*(j+1)$ $= \sum_{i=0}^{j+1} i^3$. By the code $i_{j+1} = i_j + 1$ and by the induction hypothesis, $i_{j+1} = j + 1 + 1 = (j + 1) + 1$ proving that $LI(j + 1)$ holds.

Now we need to prove that the loop terminates. Let $d_j = n + 1 - i_j$. Then d_j is a decreasing sequence of natural numbers and further when $n + 1 = i_j$, the loop exits.

To show total correctness, we note that when the loop exits $i_j = n + 1$ and by the loop invariant $i_j = j + 1 = n + 1$ so $j = n$ and therefore by $LI(j)$, $sum = \sum_{i=0}^{n} i^3$

NOTE: if they don't write it up exactly as I have but write a correct proof, please give the marks - use your judgement, rather than following the marking scheme exactly.

Question 5. [10 MARKS]

The following algorithm recursively computes $\sum_{i=0}^{n} i^3$.

Precondition: $n \in \mathbb{N}$ and $n \geq 0$.
Postcondition: Returns $\sum_{i=0}^{n} i^3$.

```
def SumRec( n):
    if (n==1 or n==0):
        return n
    else:
        return n*n*n + (n-1)*(n-1)*(n-1) + SumRec(n-2)
```

Use induction to prove that SumRec meets the postcondition give the precondition.

Solns.

We use strong induction to prove that $S(n)$ is true for all $n \geq 0, n \in \mathbb{N}$ where:

$$S(n) : \text{“SumRec(int } n\text{) returns } \sum_{i=0}^{n} i^3$$

I.H. Assume that $S(k)$ holds for $k \in \mathbb{N}$ $0 \leq k \leq n$ where n is some arbitrary natural number.

I.S.

$n = 0$ then the code returns $n = 0 = \sum_{i=0}^{0} i^3$.

$n = 1$ then the code returns $n = 1 = \sum_{i=0}^{1} i^3$.

$n \geq 2$ then by the code, SumRec returns $n^3 + (n-1)^3 +$ SumRec$(n-2)$. By the induction hypothesis, SumRec$(n-2)$ returns $(n-2)^3$. The induction hypothesis holds since $n \geq 2$, $n > n-2 \geq 0$.

Question 6. [10 MARKS]

Prove that $x \wedge (\neg y \leftrightarrow z)$ is logically equivalent to $((x \rightarrow y) \vee \neg z) \rightarrow (x \wedge \neg(y \rightarrow z))$

$x \wedge (\neg y \leftrightarrow z)$	LEQV $x \wedge ((\neg y \wedge z) \vee (y \wedge \neg z))$	($\leftrightarrow$ law)
	LEQV $(x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z)$	(Distr. law)
	LEQV $\neg(x \wedge \neg y \wedge z) \rightarrow (x \wedge y \wedge \neg z)$	($\rightarrow$ law)
	LEQV $(\neg x \vee y \vee \neg z) \rightarrow (x \wedge y \wedge \neg z)$	(De Morgan's law)
	LEQV $((x \rightarrow y) \vee \neg z) \rightarrow (x \wedge y \wedge \neg z)$	($\rightarrow$ law)
	LEQV $((x \rightarrow y) \vee \neg z) \rightarrow (x \wedge \neg(\neg y \vee z))$	(De Morgan's law)
	LEQV $((x \rightarrow y) \vee \neg z) \rightarrow (x \wedge \neg(y \rightarrow z))$	($\rightarrow$ law)

This page is for extra work space...