HW9

1) a) A payment has the form $4x+13y$ for some $x,y \in \mathbb{N}$.

$y \backslash x$	0	1	2	3	4	5	6	7	8	9
0	0	4	8	12	16	20	24	28	32	36
1	13	17	21	25	29	33	37			
2	26	30	34	38						
3	39									

b) Every payment not shown above is >36 (because moving further right increases the value), so the absence of 35 in the above table implies 35¢ cannot be paid.

c) From the table, we find our base cases: 36, 37, 38, & 39 can all be made.

Inductive Hypothesis: Suppose for some $k \geq 39$, $\forall j \in \{36, \dots, k\}$ j¢ can be paid.

Inductive Step: [We want to pay $(k+1)$¢]

$$k+1 = (k-3)+4.$$

Since $k \geq 39$, $k-3 \geq 36$, so $(k-3)$¢ can be paid by the I.H.

Then we can pay $(k+1)$¢ by paying one additional 4¢ coin.

2) Ordinary induction is enough (!).

Base Case: $\gcd(f_1, f_0) = \gcd(1,0) = 1$ [Every integer is a divisor of 0, but only 1 divides 1.]

Ind. Hyp.: Suppose for some $k \geq 0$, $\gcd(f_{k+1}, f_k) = 1$.

Ind. Step: $\gcd(f_{k+2}, f_{k+1}) = \gcd(f_{k+1}+f_k, f_{k+1})$ by def. of $\{f_n\}$, since $k+2 \geq 2$

$= \gcd((f_{k+1}+f_k) \bmod f_{k+1}, f_{k+1})$ Thm 7.5.1

$= \gcd(0+f_k, f_{k+1})$

$= 1$ I.H.

So, by PMI, theorem is true. $\square$

3) a) BinStrings(n)

Input: n, a nonnegative integer

Output: The set S of all binary strings of length n.

```
If n = 0
   Return ( {λ} )
End-if

S := ∅

T := BinStrings(n-1)

For every x ∈ T
   Add 0x to S
End-for

For every x ∈ T
   Add 1x to S
End-for

Return S
```

The simpler version with just one For-loop works, but doesn't add the elements in the right order for Bonus points

b) Base Case: $n=0$: The "If" is triggered, returns $\{\lambda\}$, which is correct. [There is one empty string, λ.]

Ind. Hyp. Suppose for some $k \geq 0$, BinStrings(k) returns the set of binary strings [Bonus: in order].

Ind. Step Run BinStrings$(k+1)$. By the I.H., T is assigned the set of binary strings of length k [in order]. Every binary string of length $k+1$ is of the form $0x$ or $1x$ for some $x \in T$, and so is added to S in one of the For-loops. Conversely, every element of S is a binary string of length $k+1$, being either $0x$ or $1x$ for a string x of length k.

Bonus

Furthermore, if $(k+1)$-strings Y & Z have $Y \leq Z$, then either

1) $Y = 0y$ & $Z = 1z$ for k-strings y, z. Then Y is added to S in the first loop, Z in the 2^{nd}.

OR

2) $Y = 0y$ & $Z = 0z$

OR

3) $Y = 1y$ & $Z = 1z$

for k-strings $y \leq z$. Then Y & Z are added to S in the same loop, but by the I.H. y appears before z in T, so Y is added to S before Z.