2.
$f: N \rightarrow N$
$f(0) = 7$
$f(n) = 2^n - 7 + 2 \cdot f(n-1)$, if $n \geq 1$

$f(1) = 2^{(1)} - 7 + 2 \cdot f(1-1)$
$f(1) = 2 - 7 + 2 \cdot f(0)$
$f(1) = 2 - 7 + 2 \cdot 7$
$f(1) = 2 - 7 + 14$
$f(1) = 9$

$f(2) = 2^{(2)} - 7 + 2 \cdot f(1-1)$
$f(2) = 4 - 7 + 2 \cdot f(1)$
$f(2) = 4 - 7 + 2 \cdot 9$
$f(2) = 4 - 7 + 18$
$f(2) = 15$

$f(3) = 2^{(3)} - 7 + 2 \cdot f(3-1)$
$f(3) = 8 - 7 + 2 \cdot f(2)$
$f(3) = 8 - 7 + 2 \cdot 15$
$f(3) = 8 - 7 + 30$
$f(3) = 31$

$f(4) = 2^{(4)} - 7 + 2 \cdot f(4-1)$
$f(4) = 16 - 7 + 2 \cdot f(3)$
$f(4) = 16 - 7 + 2 \cdot 31$
$f(4) = 16 - 7 + 62$
$f(4) = 71$

$f(5) = 2^{(5)} - 7 + 2 \cdot f(5-1)$
$f(5) = 8 - 7 + 2 \cdot f(4)$
$f(5) = 8 - 7 + 2 \cdot 71$
$f(5) = 8 - 7 + 142$
$f(5) = 167$

Using a proof by induction we can prove $f(n) = n \cdot 2^n + 7$ for all integers $n \geq 0$, when $n = 0$:
$f(0) = 0 \cdot 2^{(0)} + 7$
$7 = 0 \cdot 0 + 7$
$7 = 7$

Since we know $f(0)$ = 7 (given) and $0 \cdot 2^{(0)} + 7$ = 7 (see above), we have shown LHS=RHS for the base case $n = 0$.
For the induction step, we assume that the same is true $n - 1$. Thus, we assume $f(n-1) = (n-1) \cdot 2^{n-1} + 7$:
$f(n) = 2^n - 7 + 2 \cdot f(n-1)$
$f(n) = 2^n - 7 + 2 \cdot [(n-1) \cdot 2^{n-1} + 7]$
$f(n) = 2^n - 7 + (n-1) \cdot 2^n + 14$
$f(n) = 2^n + (n-1) \cdot 2^n + 7]$
$f(n) = 2^n + (1 + [n-1]) + 7]$
$f(n) = 2^n \cdot n + 7$
$f(n) = n \cdot 2^n + 7$

$\therefore$ by substituting $f(n-1)$ with our assumption our induction step has shown that LHS=RHS and $f(n) = n \cdot 2^n + 7$ holds for all integers $n \geq 0$.

3.
$f: N \to N$
$g: N^2 \to N$
$h: N \to N$

$f(n) = g(n, h(n))$ if $n \geq 0.$
$g(m, 0) = 0$ if $m \geq 0.$
$g(m, n) = g(m, n-1) + m$ if $m \geq 0$ and $n \geq 1.$
$h(0) = 1$
$h(n) = 2 \cdot h(n-1)$ if $n \geq 1.$

We begin by assuming $g(m, n) = mn$, with our base case: $g(m, 0) = m(0) = 0$.
For the induction step we assume the same is true for $n-1$ to prove $g(m, n) = mn$:
$g(m, n-1) = g(m, n-1) + m$

$g(m, n-1) = (m)(n-1) + m$
$g(m, n) = mn - m + m$
$g(m, n) = mn$

Our next step for $f(n) = g(n, h(n))$ is to determine the nature of $h(n)$, and since we know $h(0) = 1$, we can begin with the assumption that $h(n) = 2 \cdot h(n-1)$ is equivalent to 2^n. To test this through induction we use a base case of $h(0) = 2^{(0)} = 1$. We then use the induction step to test if $h(n-1)$ is equivalent to 2^{n-1}:
$h(n) = 2 \cdot (2^{n-1})$
$h(n) = 2^1 \cdot 2^{n-1}$
$h(n) = 2^n$

$\therefore$ because we now know $g(m, n) = mn$, and we know $h(n) = 2^n$ we can express $f(n) = g(n, h(n))$ in terms of n:
$f(n) = g(n, h(n))$
$f(n) = g(n, 2^n)$
$f(n) = n \cdot 2^n$

$\therefore f(n) = n \cdot 2^n$ is the expression of f in terms of n.

4.
$a_0 = 0, a_1 = 1$
$a_n = 2 \cdot a_{n-1} + a_{n-2}$ if $n \geq 2.$

$a_0 = 2 \cdot a_{0-1} + a_{0-2}$
$a_0 = 0$

$a_1 = 2 \cdot a_{1-1} + a_{1-2}$
$a_1 = 1$

$a_2 = 2 \cdot a_{2-1} + a_{2-2}$
$a_2 = 2$

$a_3 = 2 \cdot a_{3-1} + a_{3-2}$
$a_3 = 5$

$a_4 = 2 \cdot a_{4-1} + a_{4-2}$
$a_4 = 12$

$a_5 = 2 \cdot a_{5-1} + a_{5-2}$
$a_5 = 29$

To help prove the formula: $a_n = \frac{1+\sqrt{2}^n - 1-\sqrt{2}^n}{2\sqrt{2}}$, we will refer to the equation $x^2 = 2x + 1$. If we rearrange the formula to $x^2 - 2x - 1 = 0$ we get our $a(1), b(-2)$ and $c(-1)$ values for the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$x = \frac{-(-2) \pm \sqrt{(-2)^2 - 4(1)(-1)}}{2(1)}$$

$$x = \frac{2 \pm \sqrt{4 - 4(1)(-1)}}{2a}$$
$$x = \frac{2 \pm \sqrt{4 + 4}}{2}$$
$$x = \frac{2 \pm \sqrt{8}}{2}$$
$$x = \frac{2 \pm 2\sqrt{2}}{2}$$
$$x = 1 \pm \sqrt{2}$$

We can then let Phi(Φ) represent $1 + \sqrt{2}$ and let Psi(Ψ) represent $1 - \sqrt{2}$.
We can prove the claim by induction on n:
Both a_0 and $\frac{\Phi^0 - \Psi^0}{2\sqrt{2}}$ are equal to 0.

Both a_1 and $\frac{\Phi^1 - \Psi^1}{2\sqrt{2}}$ are equal to 1.

We can then let $n \geq 2$ and assume that the claim is true for $n - 1$ and $n - 2$:
$$a_{n-1} = \frac{\Phi^{n-1} - \Psi^{n-1}}{2\sqrt{2}}$$
$$a_{n-2} = \frac{\Phi^{n-2} - \Psi^{n-2}}{2\sqrt{2}}$$

We have to prove that the claim is true for n as well. Using the definition of x^2, the two assumptions, and the identifiers $\Phi^2 = 2\Phi + 1$ and $\Psi^2 = 2\Psi + 1$, we get:
$$a_n = 2 \cdot \left(\frac{\Phi^{n-1} - \Psi^{n-1}}{2\sqrt{2}}\right) + \frac{\Phi^{n-2} - \Psi^{n-2}}{2\sqrt{2}}$$
Which simplifies to:
$$a_n = \frac{2\Phi^{n-1} - 2\Psi^{n-1}}{2\sqrt{2}} + \frac{\Phi^{n-2} - \Psi^{n-2}}{2\sqrt{2}}$$

$$a_n = \frac{\Phi^{n-2}(2\Phi + 1)}{2\sqrt{2}} - \frac{\Phi^{n-2}(2\Psi + 1)}{2\sqrt{2}}$$

$$a_n = \frac{\Phi^{n-2}(2\Phi + 1) - \Phi^{n-2}(2\Psi + 1)}{2\sqrt{2}}$$

$$a_n = \frac{\Phi^{n-2}(\Phi^2) - \Phi^{n-2}(\Psi^2)}{2\sqrt{2}}$$

$$a_n = \frac{\Phi^n - \Phi^n}{2\sqrt{2}}$$

Proving that the fraction on the right-hand side of (1) is an integer using only Newton's Binomial Theorem:

$(a + x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k}$ on $a_n = \frac{1+\sqrt{2}^n - 1 - \sqrt{2}^n}{2\sqrt{2}}$

Psi(Ψ) $1-\sqrt{2}$

$$(1-\sqrt{2})^n = \sum_{k=0}^{n} \binom{n}{k} (-\sqrt{2})^k 1^{n-k}$$

$$a_n = \frac{\left(\sum_{k=0}^{n} \binom{n}{k} (\sqrt{2})^k\right) - \left(\sum_{k=0}^{n} \binom{n}{k} (\sqrt{2})^k\right)}{2\sqrt{2}}$$

$$a_n = \frac{\left(\sum_{k=0}^{n} \binom{n}{k} (\sqrt{2})^k - (\sqrt{2})^k\right)}{2\sqrt{2}}$$

$$a_n = \frac{\left(\sum_{k=0}^{n} \binom{n}{k} (0)\right)}{2\sqrt{2}}$$

$$a_n = 0$$

Phi(Φ) $1+\sqrt{2}$

$$(1+\sqrt{2})^n = \sum_{k=0}^{n} \binom{n}{k} (\sqrt{2})^k 1^{n-k}$$

$$a_n = \frac{\left(\sum_{k=0}^{n} \binom{n}{k} (\sqrt{2})^k\right) - \left(\sum_{k=0}^{n} \binom{n}{k} (-\sqrt{2})^k\right)}{2\sqrt{2}}$$

$$a_n = \frac{\left(\sum_{k=0}^{n} \binom{n}{k} (\sqrt{2})^k - (-\sqrt{2})^k\right)}{2\sqrt{2}}$$

$$a_n = \frac{\left(\sum_{k=0}^{n} \binom{n}{k} (2\sqrt{2})^k\right)}{2\sqrt{2}}$$

Since we know k is an odd number, we can assume k is equal to $2k+1$.

$$a_n = \frac{\left(\sum_{k=0}^{n} \binom{n}{k} (2\sqrt{2})^{2k+1}\right)}{2\sqrt{2}}$$

$$a_n = \sum_{k=0}^{n} \binom{n}{k} (\sqrt{2})^{2k+1} \cdot (\sqrt{2})^{-1}$$

$$a_n = \sum_{k=0}^{n} \binom{n}{k} (\sqrt{2})^{2k}$$

$$a_n = \sum_{k=0}^{n} \binom{n}{k} \left(\sqrt{2}^2\right)^k$$

$$a_n = \sum_{k=0}^{n} \binom{n}{k} (2)^k$$

∴ in both cases we have used Newton's Binomial Theorem to prove that a_n is an integer, whether 0 or 2^k.

5.

$T_1 = 2^1 = 2$

A board that consists of 1 tile is a single square, and we know that a single square can be either red <u>or</u> blue, hence 2.

$T_2 = 2^2 + 1 = 5$

A board that consists of 2 tiles can be two single squares or a domino, and we know that two single square can be either RR, BB, RB, BR, OR G.

$T_3 = 2^3 + 4 = 12$

A board that consists of 3 tiles can be three single square, or a single square and a domino. We could have any of the following: RRR, RBR, RRB, BRR, BBB, BRB, BBR, RBB, RG, BG, GR, GB.

To express T_n in terms of numbers that appear in this assignment we can reference question 4. In question 4 we noticed a similar pattern to what is happening with T_n. The pattern is that: T_1=a_2, T_2=a_3, T_3=a_4.
∴ this shows that $a_n$is an increment of T_n so we can conclude that $a_{n+1} = T_n$ and can be expressed as:

$a_{n+1} = \frac{1+\sqrt{2}^{n+1} - 1-\sqrt{2}^{n+1}}{2\sqrt{2}}$, very similar to the original a_n formula: $a_n = \frac{1+\sqrt{2}^{n} - 1-\sqrt{2}^{n}}{2\sqrt{2}}$.

6.

Colours	?	G	G	G	G	G
Positions	1	2	3	4	5	6

.....

To determine the number of tilings of the board B_{2n+1} has the rightmost square in position 1, we must that the remaining tiles after the rightmost square are green dominos. Therefore, because we know the rightmost square can only be either red or blue, and the rest of the tiles are green, we know the number of possible tilling's is **2**. This is because the board can either be red and green, or, blue and green.

k is an integer with $1 \le k \le n$. To determine the number of tilings of the board B_{2n+1}, in which the rightmost square is at position $2k+1$, must recognize the board as follows:

Colours	?	?	?	?	?	?	?	?	G	G	G	G	G
Positions	T_{2k}							$2k+1$	$Remaining$				

.....

Although, visually, it may not seem that $2k+1$ is a square, it is the rightmost square as outlined in the question. Now, because we know that $2k+1$ is the rightmost square, we know the remaining tiles after $2k+1$ will be green dominos (obvious). The initial part of the board is T_{2k} because the tile next to it, is the rightmost square ($2k+1$). You subtract 1, from $2k+1$ because you do not want to count the rightmost tile to get the length of the initial part of the board. To determine the total number of tilings of the board B_{2n+1}, we use the product rule to multiply the number of possible tiles that $2k+1$ can be ($2, red\ or\ blue$) by T_{2k}. $\therefore$ The total number of tilings of the board B_{2n+1} is $2 \cdot T_{2k}$.

To prove the equation $T_{2n+1} = 2 + 2\sum_{k=0}^{n} T_{2k}$ we must combine parts from part A and B, and use the sum rule:

$$T_{2(1)} + T_{2(2)} + T_{2(3)} \dots T_{2k} = \sum_{k=1}^{n} T_{2k}$$

The board is multiplied by 2 because $2k+1$ can be either red or blue: $2\sum_{k=1}^{n} T_{2k}$, and we know the 1st position to be either red or blue, having 2 possibilities: $2 + 2\sum_{k=1}^{n} T_{2k}$.

$\therefore$ by using position 1, the possibilities for T_{2k}, we have determined the board $T_{2n+1} = 2 + 2\sum_{k=0}^{n} T_{2k}$.

7.

The number of such strings of length $n-k$, that contains exactly k many D's is $\boldsymbol{\binom{n-k}{k}}$. This is because $n-k$ is the string length that you are choosing from and k is the D's that you are choosing from $n-k$.

To determine the number of tilings of the board B_n that use exactly k many dominos, we must consider that if there are $2k$ squares, there must be $n-2k$ many *remaining* squares, which again, could either be red or blue. This means that there are $\boldsymbol{2^{n-2k}}$number of tilings. The base is 2 because $n-2k$ remaining squares could either be red or blue.

To prove $T_n = \sum_{k=0}^{n/2}\binom{n-k}{k} \cdot 2^{n-2k}$ we can use summation of dominos (green tiles) and the product rule to determine T_n.

$T_n = \sum_{k=0}^{n/2}\binom{n-k}{k}$ represents the summation of all green tiles that you are choosing from in $n-k$ choices multiplied by 2^{n-2k} which represents the number of red or blue tilings. The product of all these is T_n.

8.

Since the algorithm is recursive, we are going to set up a recurrence for $P(n)$.

Assume that $n \ge 2$ is even. When we run $\text{ELISAGOESTOTHEPUB}(n)$, Elisa does the following:

- During the first recursive call, she drinks $P(n/2)$ pints of cider.
- Next, she drinks one pint of cider.
- During the second recursive call, she drinks $P(n/2)$ pints of cider.

We conclude that:
$P(n) = 2 \cdot P(n/2) + 1$

Assume that $n \geq 1$ is odd. When we run $\text{ELISAGOESTOTHEPUB}(n)$, Elisa does the following:

- She drinks one pint of cider.
- During the first and only recursive call, she drinks $P(n-1)$ pints of cider.
- Next, she drinks one pint of cider.

We conclude that:
$P(n) = 1 + P(n-1) + 1$
$P(n) = P(n-1) + 2$

Determining $P(n)$ for some small values:	
n	$P(n)$
1	1
2	3
3	5
4	7

After determining $P(n)$ for some small values, it looks like $P(n) = 2n - 1$.

Using the base case first we can show $P(1) = 2(1) - 1 = 1$. Then on the odd and even formulas we deduced from the pseudo code earlier, we just the induction step:

$P(n) \geq 2$ is even	$P(n) \geq 1$ is odd
$P(n) = 2 \cdot 2(n/2) - 1$ $P(n) = 2 \cdot (n) - 1$ $P(n) = 2n - 1$	$P(n) = (2(n-1) - 1) + 2$ $P(n) = (2n - 2 - 1) + 2$ $P(n) = 2n - 3 + 2$ $P(n) = 2n - 1$

$\therefore$ we have shown that the value of $P(n)$ is equal to $2n - 1$ when running the $\text{ELISADRINKSCIDER}(n)$ algorithm.

9.

Here is what the recursive algorithm MAXELEM(S, n) structure would be composed of:

```
MERGESORT(S,n){
        //S is a list of n >=1 numbers.
        if(n>= 2)
        then m=[n/2];
                S1=list containing the first m elements of S;
                S2=list containing the last n-m lements of S;
                S1=MERGESORT(S1, m);
                S2=MERGESORT(S2, n-m);
                S=MERGE(S1, S2);
        endif;
        return S;
}
```

Below is the MERGE function:

```
Merge (S1, S2){
    //S1 and S2 are sorted lists
    S=empty list;
    while S1 is not empty and S2 is not empty
    do x=first element of S1;
      y=first element of S2;
      if x<=y
            then remove x from S1;
            append x to S;
      else
            remove y from S2;
            append y to S;
    endwhile;
    if S1 is empty
            append S2 to S;
    else
            append S1 to S;
    break;
    return S;
    }
```