2.

```
                                        3   6 .  0  6  9
                                       13  01 . 00 00 00
2*0=0               03 * 3     - 9
                                        4  01
2*3=6               66 * 6      -3  96
                                            5  00
2*36=72            720 * 0           -       0
                                            5  00  00
2*360=720         7206 * 6          -4  32  36
                                                67  64 00
2*3606=7212      72129 * 9             -64  91 61
```

3\. i.

```
  209    13    +          209 + 836 + 1672 = 2717
  418     6
  836     3    +
 1672     1    +
```

ii.

```
         1101 0001       209
     *        1101        13
     -----------
         1101 0001       209
     0 0000 000
    11 0100 01           836
 + 110 1000 1          +1672
 ---------------       -----
   1010 1001 1101       2717
```

4.

$$\{\sqrt{n}\}\ \left\{\frac{n}{(\lg n)^5}\right\}\ \{n^2\}\ \{7^{\lg n}\}\ \left\{\frac{n^3}{\lg n},\ \frac{n^3}{\ln n},\ n^2\lg n+\frac{n^3}{\lg n}\right\}$$

$$\left\{n^3,\ 17n^3+391n^2,\ \sum_{j=0}^{n}7n^2\right\}\ \{(\lg n)^{\lg n}\}\ \{(\ln n)^{\lg n}\}\ \{(2^3)^n\}\ \left\{\frac{n!}{2^n}\right\}\ \{2^{(3^n)}\}$$

5. If you already have the min and max of the first $n-2$ items, then compare the last 2, then compare the min (respectively max) of these with the current min (max). So adding two elements requires three more comparisons. A working program appears on the course webpages, section Prologue.

6. Claim: for all integers $n \geq 1$, `ff(n)` returns `(fib(n-1), fib(n))`.

   Proof by induction. Base case. Let $n = 1$. Then, in `ff(1)`, the `if` condition is true, so (0,1) is returned, which is `(fib(0),fib(1))`, so the claim holds in this case.

   Inductive case. Assume that the claim holds for a fixed value of $n \geq 1$, say $n = t$. We want to show that, under this assumption, the claim holds for the next value of n, namely $t + 1$.

   So consider the call `ff(t+1)`. $t = n \geq 1$, so $t + 1 \geq 2$, so the `if` condition is false, so `x,y = ff(t+1-1) # = ff(t)` executes, followed by `return y, y+x`. By our inductive hypothesis, the claim holds for $n = t$, so `ff(t)` returns `(fib(t-1), fib(t)`, so `x,y = fib(t-1), fib(t)`, so `y, y+x` is returned, namely `fib(t),fib(t)+fib(t-1)`. But `t >= 1`, so `fib(t)+fib(t-1)=fib(t+1)`, so the pair returned by `ff(t+1)` is indeed `fib(t),fib(t+1)`, so the claim holds in this case.

   So, by the principle of induction, the claim holds for all integers $n \geq 1$. $\square$

7. Define $f(n) = 2(n - 1)$. Claim: for all integers $n \geq 1$, $L(n) = f(n)$.

   Proof by induction. Base case. Let $n = 1$. Then $L(n) = 0$, and $f(n) = 0$, so the claim holds in this case.

   Inductive case. Assume that the claim holds for a fixed value of $n \geq 1$, say $n = t$. Now we want to show that, under this assumption, the claim holds for the next value of n, namely $t + 1$.

   So consider the execution of `ff(t+1)`. It performs two arithemtic operations (computing `n-1` and `y+x`), and also calls `ff(t+1-1)=ff(t)`, where — by the inductive hypothesis — it performs $f(t) = 2(t - 1)$ arithops. So the total number of arithops is $2 + 2(t - 1) = 2t = 2(t + 1 - 1) = f(t + 1)$, so the claim holds in this case.

   So, by the principle of induction, the claim holds for all integers $n \geq 1$. $\square$

8. Arguing as in the previous question, the total number of function calls is in $\Theta(n)$. Within each call, there are a constant number of assignments and arithmetic operations. Each of these takes $\Theta(t)$ time, where t is the number of bits of the numbers involved. We are computing Fibonacci numbers, so $fib(n)$ is in $\Theta(r^n)$, where $r = \sqrt{1+5}/2 = 1.618\ldots$, so t is in $\Theta(\lg(r^n)) = \Theta(n)$, so the total runtime is in $\Theta(n^2)$.

   You can check this: time an iterative python program as you double the input value n. Assuming a runtime of $\approx cn^2$, then doubling the input value yields a runtime of $\approx c(2n)^2 = c4n^2$, i.e. four times as long. See the course web pages for some python code that does this.