**1.** Determine if each of the following functions is . Answer Y for yes and N for no.

is only true

1. If there is some positive (a positive starting point exists, don’t care what it is)
2. If there is some positive

Both of which such that and , where *c* is a constant.

Big O is an equation of how time scales with respect to some input variables (a simplified analysis of an algorithm’s efficiency). You are interested in what happens when your problem input size gets large. Generally (not in real-world application), be sure to ignore constants, and for all, know that certain terms dominate others:



Thus, the answers are:

1. **Y**
2. **Y**
3. **N**
4. **Y**
5. **N**
6. **Y**
7. **Y**

**2.** Find the least integer *n* such that is for each of the following functions:

[Similar problems to show how it's done](https://www.slader.com/discussion/question/find-the-least-integer-n-such-that-f-x-is-oxn-for-each-of-these-functions/), [simplified examples](https://mathcs.clarku.edu/~djoyce/ma114/Sec32.pdf)

Part A

Refer to the domination chart above in problem 1. Note that . Since means less than or equal to, you can convert to the same term and to . Thus,

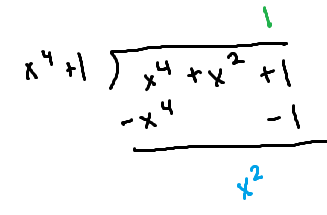
The highest exponent on the right side of the inequality is 10, so **10**.

Part B

The largest exponents in the problem is 8, so **8**. The exponent for the log does not matter, because the exponent surrounds the entire log. Also notice that the exponent is less than *x*’s one. is still dominated by any positive power of *x*, so it has no effect.

Part C

Divide the denominator into the numerator to simplify. So,



Thus, . The largest power of *x* is therefore **0** because firstly, . Secondly, the fraction resolves to a number lower than 1 (if you plug any number of *x* in it).

Part D

Note that the denominator is bigger than the numerator. Since dominates 1 and dominates , so you can disregard those and get . So, the exponent for *x* is **-1**.

**3.** You are the head of a division of a big Silicon Valley company and have assigned one of your engineers, Jim, the job of devising an algorithm to sort through an English text of *n* words and convert it into an Esperanto document. Jim comes up with an algorithm which takes bit operations to handle an input text with *n* words. Suppose the computers in your business can handle one bit operation every nanosecond (1 nanosecond = seconds).

1. How many nanoseconds would Jim’s algorithm take to convert a text with 15 words on these computers?
2. How many HOURS would Jim’s algorithm take to convert a text with 66 words on these computers? (Do not round your answers for WeBWorK.)
3. Recall a million is , a billion is , and a trillion is . For an input text of 100 words, the statement that best describes the performance of Jim’s algorithm is:

Part A

You would just plug in 15 for *n* in the algorithm to get **33218**.

Part B

First, compute the number of nanoseconds using the algorithm with *n* as 66, which is . To convert it to seconds, multiply it by , so you get .

To convert seconds to minutes, divide the number by 60 to get 1229782938, and from minutes to hours, divide that by 60 as well to get **20496382.3**.

Part C

First, find the number of nanoseconds using the algorithm with *n* as 100, which is . Multiply the number by , divide by 60 once for the minutes, divide by 60 for the hours, divide by 24 for the number of hours in a day, then divide by 365 for the number of days in a typical year. The number you get is .

Compare it to the choices (the bases of 10 for a million, billion, and trillion), and notice that if you move the decimal point one place to the right, you get , and represents a trillion. Notice the numbers after the decimals indicate that **his algorithm would take more than 40 trillion years to run**.

**4.** List the first four terms of each sequence.

The first four terms means *n* starts from 1. So, plug in 1, 2, 3, and 4 for *n* for each.

Part A

**-8, -10, -12, -14**

Part B

**-3, 9, -27, 81**

Part C

**-4, 9, -30, 87**

**5.** Give a recursive definition of the following sequences on . Match the definition with the corresponding sequence. (Definitions are bullets, sequences are letters.)

Part A

This is definitely (**d**) because if you think about it, when you raise the power of something by 1, you multiply that with that something again to get the next term. It’s multiplying (think exponents), and multiply the previous term by the 6 to get the correct answer. If you plug in 2 for *n*, for the sequence, you get 36, and doing the same for the definition also gets 36.

Part B

This is definitely (**c**) because . This is the only sequence where the first term is 7 because the other members, if you plug in 1, you get 6 for the rest of them.

If you find the definition when , you see that should be 13. If you plug in 2 for *n* in the original equation, you get as well, so this is correct.

Part C

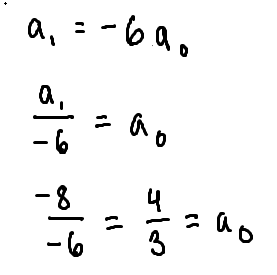
This is definitely (**b)** . Notice that no matter what number *n* is for the sequence, the answer will always be 6. This works the same for when you plug it into the equation.

Part D

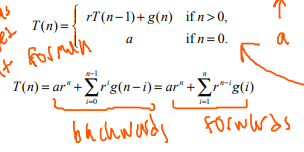
This is definitely (**a**) . This uses addition, where every time you increase *n*, you add the coefficient to it, because that’s essentially what the multiplication in the sequence is doing, adding 6 for *n* amount of times. If you test for using the given definition of , it matches up.

**6.** Find the solution for the following recurrence: .

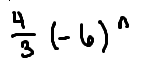
Since this is a first-order linear recurrence, you must find the first term of the recurrence relation, which is . This isn’t provided in the problem, so you must solve for it like so:



Next, you use the theorem for backwards.



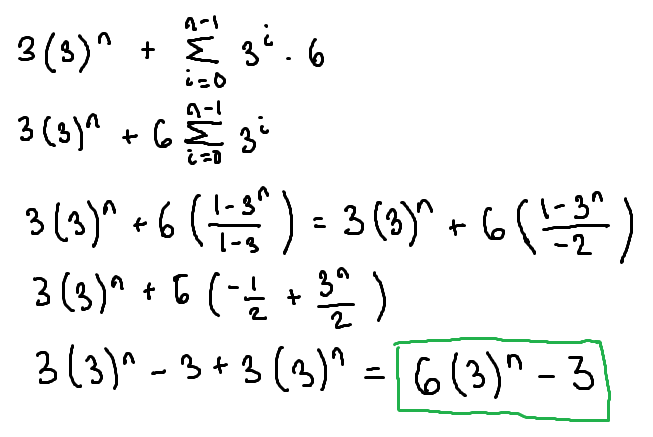
Thus, you get . The term goes to 0 because there is no term. So, if you multiply that part by 0, you get 0, which basically nullifies that whole part.

And the answer is  and if you test it for inputs when , you get the correct outputs.

**7.** Suppose the sequence satisfies the following linear recurrence: .

1. Find the closed (explicit) formula for .
2. Compute the value .

Part A



You can take the 6 outside of the sigma because it is a constant and doesn’t have to be included, also since it is multiplied. Also, always test the efficient formula out with sample values of *n* and see if you get the same answer as plugging it into the recurrence.

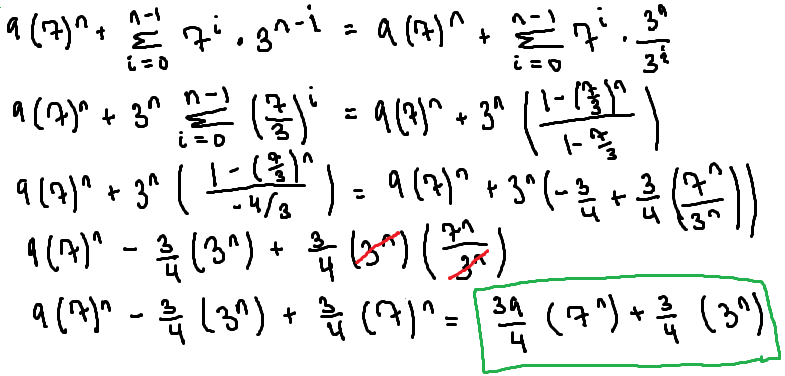
Part B

**3188643**

**8.** Suppose the sequence satisfies the following linear recurrence: .

1. Find a closed (explicit) formula for .
2. Compute the value .

Part A (the ¾ 3^n is a negative)



Part B

**2754177965**.

**9.** For each sequence, find a closed formula for the general term, .



Part A

Notice that the difference between each is 18 for each number. When given a list of sequence, unless otherwise specified, the first number in the sequence is always when . Using that difference, it’s always going to be that difference multiplied by *n* because the number of times 18 is subtracted from the first term will multiply by *n*. To make this work, you also have to subtract it by a certain amount; solve to find a number. Thus, the closed formula is **-18*n*-73**. (Also notice that if , -73 would be the 0th term.)

Part B

Since the numbers are increasing more rapidly this time, test out dividing the second term by the first term, and you get the GCD 9. You can do this for the others too; if you multiply the previous by 9, you get the next term. Since the first term (when ) is 97, you should find what the 0th term is by dividing 97 by 9 to get . The closed formula is **(97/9)9*n***.

Part C

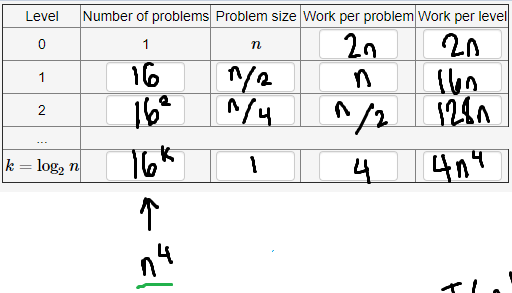
Try to relate each term to its *n* term. Since the first term () is 0, there must be a ( multiplied by something to get the term. You can conclude multiplication is involved because the terms do not increase constantly like in an arithmetic sequence. If there is an , in order to get 3 (when ), you must multiply it by 3. Since *n* is equal to 2, you can relate this back to *n* again and add 1 to 2, in which then you get 3. Test this out for other terms and see that works, in which you get the closed formula **(*n*-1)(*n*+1)**.

**10.** Suppose the sequence satisfies the following divide-and-conquer recurrence: .

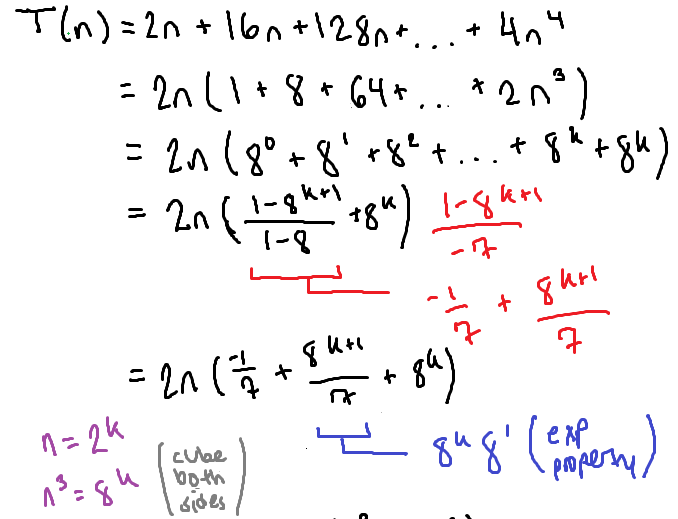
1. Assuming , fill in the recursion tree diagram for .
2. Find the closed (explicit) formula for .

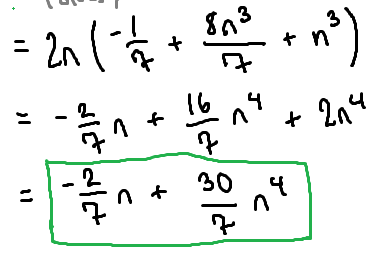
For Part B, usually, you would solve this using the geometric series formula () or use the way the handshake formula was made.

Part A



Part B





**11.** You will find the solution to the following recurrence: for with initial conditions . The first step in any problem like this is to find the characteristic equation by trying a solution of the “geometric” format , assuming . You get . Since you are assuming , you can divide by the smallest power () to get the characteristic equation . (Notice since your recurrence was degree 2, the characteristic equation is degree 2.)

1. Find the two roots of the characteristic equation and . When entering your answers, use .

Since the roots are distinct, the general theory tells you that the general solution to your recurrence is for suitable constants alpha 1 and 2. To find the constants, use the initial conditions. Plug them and the roots into the theory (and .

1. Find the constants. (Be careful to note that when *n* is even.)

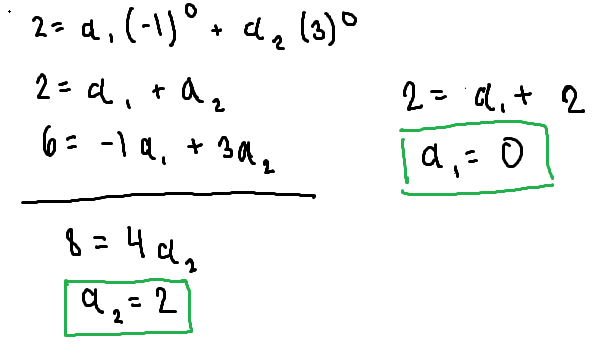
Note that the final solution of the recurrence is the theorem, where the roots and constants are plugged in, giving an explicit numerical formula in terms of *n* for *a(n)*.

Part A

Since you have the characteristic equation, you can subtract 2*r* and 3 to the other side to get and factor it to get two roots, like . So, solve for *r* for each set of parentheses, and note which one is the first or second root by finding which one is smaller. In this case, **r1 = -1** and **r2 = 3**.

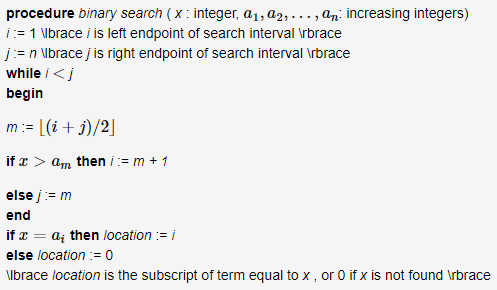
Part B

Plug the roots into the theories and solve the system of linear equations.



Solved with help from [this PDF](http://wrean.ca/cazelais/rec.pdf).

**12.** Refer to the binary search algorithm given below:



Suppose your list of increasing integers is shown:

| 1 | 2 | 3 | 4 | 5 | 6 | 9 | 13 |
| --- | --- | --- | --- | --- | --- | --- | --- |

Suppose you conduct the binary search algorithm on this list where you search for 9.

1. In the language of the above algorithm, find the values for variables *x*, *n*, , and .
2. After the first iteration of the while loop, what are the values of the variables *i*, *j*, and *m*?
3. Intuitively, after the first iteration of the while loop, you have to cut down your search to a set *S* of roughly half of the original numbers on the list. Check the numbers that are in the set *S* after the first iteration of the while loop. *S* has the following elements (choose).
4. Find the values of the variables *i*, *j*, and *m* after the second iteration of the while loop.
5. Find the values of the variables *i*, *j*, and *m* after the third iteration of the while loop.
6. After the third iteration of the while loop, the while loop terminates and the variable ‘location’ is computed. Find the value for the variable *location*.

What is binary search? That is basically what this procedure is talking about. Refer to the Powerpoint presentation (noted as Section 4.3 Recurrences).

* Searching for a number within a given range (or in this case, a given list)
* Need to discover *x*, so ask two questions:
  + Is *x* greater than *k*, where *k* is (at first) the middle number of the list (kind of)?
    - Depending on this answer (Y/N), can vary on which half you’re not searching anymore from
  + Is *x* equal to *k*?
    - At this point, *k* is the value you’re looking for. This is usually the final question, when the range (list) contains only 1 number. This should have a yes answer if done correctly.

Part A

*x* is the integer you’re searching for, so in this case, it’s **9**.

*n* is the number of terms in the list; it’s **8**.

*a*2is the second term of the list; it’s **2**.

*a*4is the fourth term of the list; it’s **4**.

Part B

The meaning of the algorithm, in words (also important to note that list is and must be in increasing order):

* The algorithm is saying that *i* is the left endpoint of the search interval. During the first iteration, it would be *a*1, because it is the smallest. The value of *i* here would be 1. *j* is the right endpoint of the search interval, so during the first iteration, it would be *a*8, or 13, because it is the largest.
* The while loop condition says , so this is only going to be when you’re asking the first question (is *x* greater than *k*?). If *i* and *j* became equal to each other, the condition would no longer uphold and would be forced to move outside it. This will only happen when the bounds go to 1 number and then pose the last question (equals). Both variables are subscripts of *a*.
* What is *m*? *m* is the bounds added together and divided by 2, so effectively cutting out half of the possibilities. This is also a subscript of *a*, because this is what the half value is. The symmetrical “L”s are the “floor”. Since this is not exactly explicitly defined anywhere, you can assume the floor is rounding down the answer if a decimal is reached.
* The algorithm continues, so if *x* is greater than *a*m, then the left bound will be the next term. Otherwise (if *x* is less than *a*m), the right bound is set to *a*m (making them equal to each other, as shown). After, the while loop is performed again (if condition is met).
* After the loop finishes (when condition fails), the location is set to *i* if the last question (equals) is true. If not, the location is 0 (no location).

The question is asking after the *first* iteration, so after the loop is performed once, meaning what are the values of *i*, *j*, and *m* after (in the if-else block for *i* and *j*, and within the loop itself for *m*)?

Is 1 < 8? Yes. So, **4** due to the floor rounding.

Now, is 9 (value you’re searching for) > *a*4(because *m* is equal to 4 right now)? So, in terms of the actual values, is 9 > 4? Yes. Thus, *j* continues to be **8**, and *i* = 4 + 1 = **5**.

Part C

*S* has the elements between and including *a*5and *a*8. Thus, the values included are **5**, **6**, **9**, and **13**.

Part D

After the *second* iteration:

* Is 5 < 8? Yes. So, **6** due to floor rounding.
* Is 9 > *a*6 (9 > 6)? Yes, so *j* continues to be **8** and **7**.

Part E

After the *third* iteration:

* Is 7 < 8? Yes. So, **7** due to floor rounding.
* Is 9 > *a*7 (9 > 9)? No. So, *j* is equal to **7** (*j* = *m* in algorithm) and *i* is also equal to **7**.

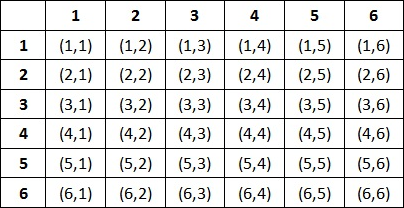
Part F

Though the algorithm is stated to have ended in the problem, it’s important to know why. Is 7 < 7? No, so the loop condition is false and therefore ends. Does 9 = *a*7(9 = 9)? Yes! So, the location is **7**.

**12.** Two fair dice are tossed, and the up face on each die is recorded. Find the probability of observing each of the following events:

1. The sum of the numbers is 10 or more
2. The difference of the numbers is 2
3. The sum of the numbers is even

First, find the number of possibilities for how many sums you can get. There are 2 spaces for the number of outcomes for each die. Repetition is allowed (you can get two 3’s) and order matters (whether you get 2 and 1 or 1 and 2 is different); thus, this is a list. You can use product principle to multiply the 2 elements together to get 36. You can list possibilities as ordered pairs in a chart like so for visual effect:



Part A

The possibilities of getting a sum of 10 or more is through getting a (4,6), (6,4), and (5,5), AND all the pairs above that, thus totalling in 6 possibilities out of 36. You can represent this as a fraction, **6/36**.

Part B

The possibilities of the two numbers being a difference of 2 is (6,4), (4,6), (5,3), (3,5), (4,2), (2,4), (3,1), and (1,3), so this is **8/36**.

Part C

To get a sum of an even number, the two numbers must both be even or must both be odd. Thus, you can count how many there are in the chart, or you can just divide by 2 because each sum is going to be even or odd and there are half of each. If you notice, when you begin counting, the first row has 3, and the second row also has 3, so there is a pattern if you continue. Thus, this probability is **18/36**.

**14.** A fair coin is tossed 3 times and the events *A*, *B*, and *C* are defined like so:

A: At least one head is observed.

B: At least two heads are observed.

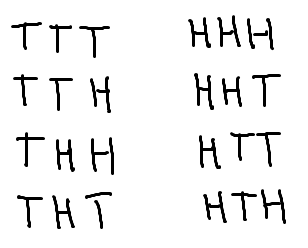
C: The number of heads observed is odd.

Find the probabilities by summing the probabilities of the appropriate sample points:

The overbar (line above the letter) in Discrete means “not” as defined in [this Wikipedia page](https://en.wikipedia.org/wiki/Complement_(set_theory)#Absolute_complement), and theequals the probability of *A* plus the probability of *B* minus the probability that both events occur.

First, you also need to find the number of possibilities. Let’s say there are three spaces, like a list. Since each space has 2 possibilities (heads or tails), you can do to get 8 possible outcomes.

The possibilities are as follows:



Part A

The condition says at least one head is observed. What is the opposite of at least? Less than. So, less than 1 head is observed. What are the instances where there are no heads at all? Once, when all coins flip to tails. So, this is **1/8**.

Part B

What is the probability that at least two heads are observed? This means outcomes with 2 or 3 heads. There are 4 in them, so this is 4/8.

What is the probability that the number of heads observed is odd? This means outcomes with 1 or 3 heads. There are 4, so this 4/8.

Add them together to get 8/8. However, what is the probability that the outcome’s number of heads is odd AND at least 2? When there are three heads. There is only 1 possibility. Subtract 1/8 (this probability) from 8/8 to get **7/8**.

Part C

What is the probability that less than 1 head is observed? Once, where the outcome is all tails. This gives 1/8.

How about less than 2 heads are observed? This is only when there are either 0 heads or 1 heads. You can exclude the 0 heads because that’s already included in the probability for not *A*. The number of outcomes with one heads is 3. Thus, this is 3/8.

What is the probability of *C*? This would include 1 and 3 heads. Since 1 heads is already included in not *B*, this doesn’t have to be included. Thus, this leaves 1/8. Add these together to get 5/8.

What is the probability of all 3 conditions being true? Nothing. So, this is **5/8**.

**15.** An experiment consists of flipping a coin, rolling an 11 sided die, and spinning a roulette wheel. What is the probability that the coin comes up with heads and the die comes up less than 4 and the roulette wheel comes up with a number greater than 10?

Imagine there are spaces for each. There are a number of possibilities for each space. This is so you can find the total amount of possibilities. There are three spaces for the three different elements to account for in the probability. These are three independent events, so you must multiply their individual probabilities together to get their probability.

For this, use American roulette; it has 38 possibilities and includes numbers 0, 00, and 1 to 36.

So, **39/418**

**16.** Suppose that *X* is an event, and that . What is (i.e. the probability that *X* will not occur)?

This is **0.75**, because if you subtract the probability of *X* from 1 (the maximum probability in math), you get this number.

**17.** Four candidates are running for mayor. The four candidates are Adams, Brown, Collins, and Dalton. Employing the subjective approach, a political scientist has assigned the probabilities:

*P*(Adams wins) = 0.45

*P*(Brown wins) = 0.06

*P*(Collins wins) = 0.25

*P*(Dalton wins) = 0.24

1. Adam loses
2. Either Brown or Dalton wins
3. Either Adams, Brown, or Collins wins

Part A

Refer to problem 16. This is **0.55**.

Part B

If this is an “or” or “either”, you can add their probabilities up since they are both dependent events. Thus, this is **0.30**.

Part C

Refer to Part B. You could also just subtract Dalton’s probability of winning from 1. This is **0.76**.

**18.** In the 3/49 lottery game, a player selects 3 numbers from 1 to 49. What is the probability of picking the 3 winning numbers?

This is a combination because you’re selecting 3 winning numbers from a collection of numbers. However, this is assuming that the order of the numbers don’t matter and repetition is not allowed. So, this is , and there is only 1 outcome out of the total number of possibilities, so it’s **1/18424**.

**19.** 3 cards are drawn at random from a standard deck.

1. Find the probability that all the cards are hearts.
2. Find the probability that all the cards are face cards (kings, queens, jacks).
3. Find the probability that all the cards are even (consider aces to be 1, jacks to be 11, queens to be 12, and kings to be 13).

Part A

Cards have 4 suites: hearts, spades, diamonds, and clubs. There are 13 of each, thus having 13 heart cards, and note there are a total of 52 cards. Once you pick the first heart, there are 51 cards left in the deck, and 12 heart cards left in the deck. These are dependent events. Like so, the probability is **11/850**. Do this 3 times because there are 3 cards. Think of this as the probability of getting a heart card for the first card drawn multiplied by the probability of getting a hard card for their second card after the whole and part is subtracted (removing the card from the deck), and continue on.

Part B

There are 3 face cards per suite. Thus, there are 12 total face cards in the deck. Do the same as Part A and multiply the probabilities of each dependent event together: **11/1105**.

Part C

How many even cards are in the deck? The even cards in each suite are 2, 4, 6, 8, 10, and queen. This is a total number of 6 even cards per suite. Multiply this by 4 to get 24 total even cards in the deck. Thus, do the same thing as Part A: **506/5525**.

**20.** You flip a fair coin 10 times.

1. What is the probability that it lands on heads exactly 6 times?
2. What is the probability that it lands on heads at least 6 times?

Part A

What is the total amount of outcomes? This is a list, so you can do , with 2 being the amount of possibilities for each toss and 10 being the number of tosses. This is 1024 total outcomes.

The number of outcomes with exactly 6 heads is (10 *choose* 6) **210/1024**.

Part B

If you want to determine this, you could add the number of outcomes of 6, 7, 8, 9, and 10. So, . So, add them all together () = 386. Divide this by the total number of outcomes and get **386/1024**.