**1.** Encrypt the message " HALT " by translating the letters into numbers

(via A=0, B=1, C=2, D=3, E=4, F=5, G=6, H=7, I=8, J=9, K=10, L=11, M=12, N=13, O=14, P=15, Q=16, R=17, S=18, T=19, U=20, V=21, W=22, X=23, Y=24, Z=25)

and then applying the encryption function given, and then translating the numbers back into letters.

First, you must translate the message “HALT” into numbers.

“H” position: 7

“A” position: 0

“L” position: 11

“T” position: 19

Part A

Plug each position number into the equation. Then, synchronize these letters up according to the new position number.

* **J**
* **C**
* **N**
* **V**

So, the answer for this is **JCNV**.

Part B

Do the same as part A.

* **R**
* **K**
* **V**
* **D**

So, the answer for this is **RKVD**.

Part C

Do the same as part A.

* **L**
* **E**
* **P**
* **X**

So, the answer for this is **LEPX**.

**2.** Decrypt the following messages encrypted using the Caesar cipher:

The alphabet is A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z

1. EXVB KRVSLWDO
2. HDW GLP VXP
3. FVIRHKLD

To decrypt, you do the inverse of the Caesar cipher, thus: .

Part A

Find the positions of each letter in the encrypted message (and excluding repetitions):

Next, find the old positions of each letter using the inverse Caesar cipher:

* E = **B**
* X = **U**
* V = **S**
* B = **Y**
* K = **H**
* R = **O**
* S = **P**
* L = **I**
* W = **T**
* D = **A**
* O = **L**

The message is **BUSY HOSPITAL**.

Part B

Do the same as part A. Since all three encrypted messages use the same inverse Caesar cipher, you can ignore the letters that have been decrypted already. The following is current positions of letters:

Next, decrypting:

* H = **E**
* G = **D**
* P = **M**

The message is **EAT DIM SUM**.

Part C

Do the same as Part A and B.

Next, decrypting:

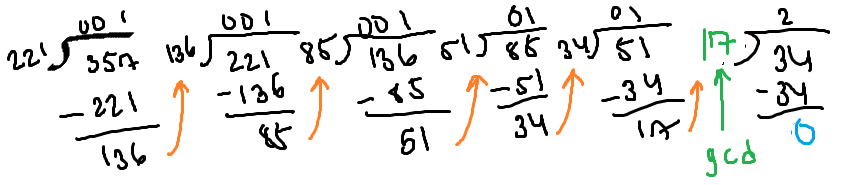
* F = **C**
* I = **F**

The message is **CSFOEHIA**.

**3.** Determine the greatest common divisor of 357 and 221.

[Khan Academy's article on The Euclidean Algorithm](https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/the-euclidean-algorithm) is helpful in solving this problem.

So, divide the larger number and the smallest number (so ). Then, take the remainder of that and divide it by the “B” of the expression (so in this case, it would become ). Keep going until the remainder of the division is 0, as shown in the picture:



So, the answer to this question is **17**.

You can prove this with the set of equations formatted using the Division Theorem, where in the first equation, :

Divide like this until the remainder comes out to be 0. The divisor is the 17 (or the *n* value).

**4.** Complete . Then, find a pair of integers *x* and *y* such that .

First, find the GCD and do the same as the last problem, where 61 is the *k* and 41 is the *j*. In the first equation, 20 is the *r* and 1 is the *q*:

You can make a table that looks like this:

| ***i* (steps)** | ***j*[*i*]** | ***k*[*i*]** | ***q*[*i*]** | ***r*[*i*]** |
| --- | --- | --- | --- | --- |
| **0** | 41 | 61 | 1 | 20 |
| **1** | 20 | 41 | 2 | 1 |
| **2** | **1** | 20 | 20 | 0 |

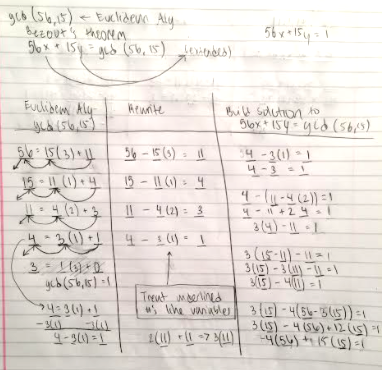
So, the greatest common divisor is **1**. Next, solving the equation. Reference Lecture 2 under the Extended GCD Algorithm.

1. Set since the following holds: .
   1. Basically, here, what this means is the 1 and 20 are the coefficients of the *x* and *y* in the equation . The GCD of the two numbers is 1, so this would be , and the only way that this can be true is if *x* is equal to 1 and *y* is equal to 0.
   2. However, the problem asks for , so the *j* is the 61 and *k* is 41 (when finding the GCD, order doesn’t matter.
2. Fill out the table accordingly.
3. Find out that *x* = *x*[0] = **3** and *y* = *y*[0] = **-2**.

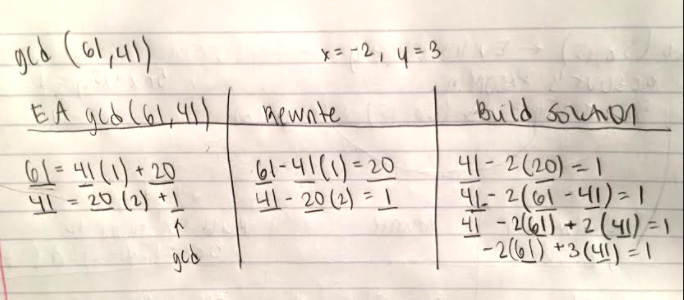
**CHECK LATER TO SEE THE EXTRA STEP FROM THE RIGHT ANSWER**

| ***i* (steps)** | ***j*[*i*]** | ***k*[*i*]** | ***q*[*i*]** | ***r*[*i*]** | ***x*[*i*] = *y*[*i +* 1] - *q*[*i*] *x*[*i* + 1]** | ***y*[*i*] = *x*[*i* + 1]** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 61 | 41 | 0 | 61 |  | 1 |
| **1** | 61 | 61 | 1 | 0 | 1 | 0 |

Easier way to do it, but probably not approved



Work for the easier



**5.** The goal of this exercise is to practice finding the inverse modulo *m* of some relatively prime integer *n*. We will find the inverse of 4 modulo 27, i.e., an integer *c* such that .

First, we perform the Euclidean algorithm on 4 and 27:

27 = 6 \* **4** + **3**

**4** = **3** \* 1 + 1

[Note: Your answers on the second row should match the ones in the first row.]

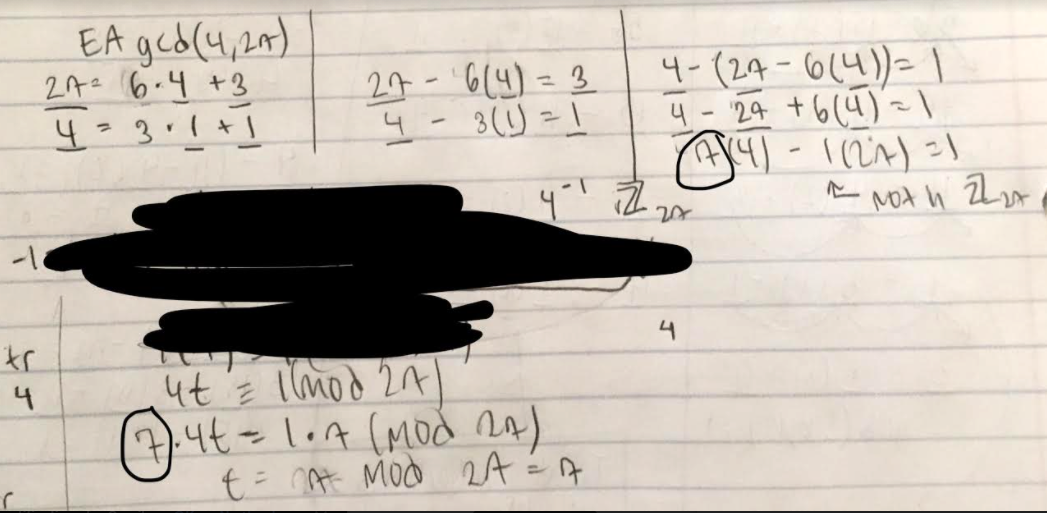
Thus, gcd(4,27) = 1, i.e. 4 and 27 are relatively prime.

Now we run the Euclidean algorithm backwards to write for suitable integers *s*, *t*.

*s* = **-1**

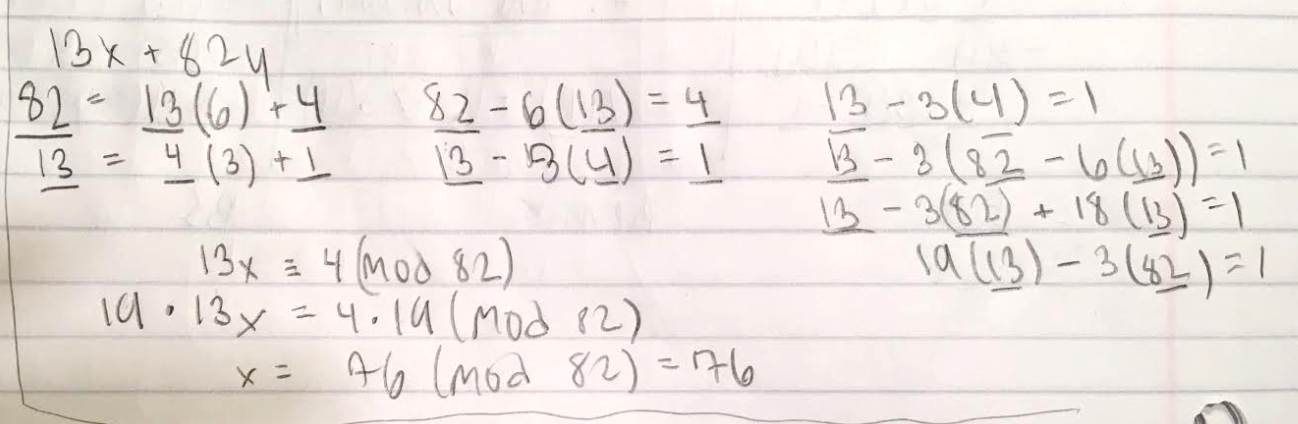
*t =* **7**

when we look at the equation , the multiple of 27 becomes zero and so we get . Hence the multiplicative inverse of 4 modulo 27 is **7**.



**6.** Find the smallest positive integer *x* that solves the congruence . (Hint: From running the Euclidean algorithm forwards and backwards we get . Find *s* and use it to solve the congruence.)

*x =* **76**



**7.** Here is another version of the Quotient-Remainder Theorem: Given any integers *n*, *d* with , there exist unique integers *q*, *r* satisfying

Find the quotient and remainder using the theorem above for the following pairs of integers:

Part A

Write the equation out (the one in (1)) and establish the range using the given value for *d* like so:

Now that you’re aware of the range, let’s try to get as close to 11 as you can. For this, *q* would have to be **5**.

Though *q* has an unlimited range, the remainder bounds it to be only 5. For example, if you put 6 for *q*, the remainder would be -1, but -1 is not in the remainder range.

2 times 5 is 10, and you only need a remainder of **1** to be in the range. (Remember this range asks that *r* can be greater or equal to 1.)

As you solve the rest of the problems, remember that as long as the remainder is in the range described, it’s okay if it’s negative, so just find whatever value works and make sure you check that it’s in the range. So, the answers are (*q*, *r*):

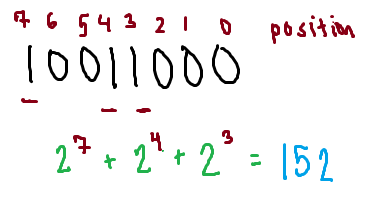
1. **5, 1**
2. **4, -1**
3. **-4, 1**
4. **8, -2**
5. **-8, 2**
6. **6, 4**
7. **-7, 4**

**8.** Convert the following integers from binary notation to decimal notation:

1. 10011000
2. 111111010

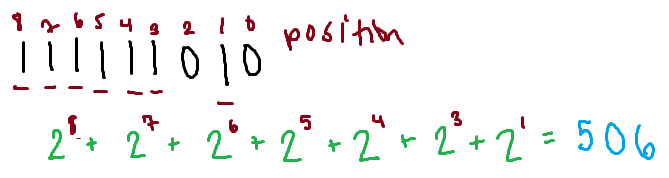
Part A

To convert from binary to decimal, count the position of each number. Start from right to left and start from position 0 (the rightmost number). Label the positions that have the number 1 occupying them. Use these positions as the powers of 2. The answer is **152**.



Part B

Do the same as Part A here.



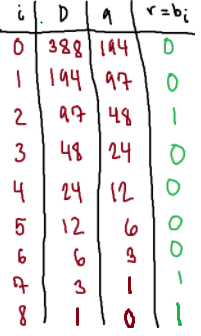
**9.** Convert the following integers from decimal notation to binary notation. (Do not put extra zeros in front of your binary notation or it might confuse WebWorK. So write 101 versus 0101, etc.)

1. 388
2. 1936
3. 150063

Part A

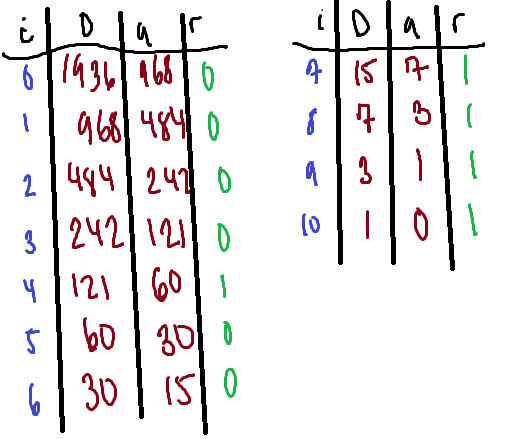
Arrange a table, where *i* is the step number, *D* is the dividend, *q* is the quotient, and *r* is the binary number. Then, fill in 388 as the first *D* for step 0. Divide by 2, and the remainder of that operation is either 0 or 1. If *D* is an even number, the remainder is 0; if it’s odd, it’s 1.

After completion, order all the numbers from right to left, so the number is **110000100**.



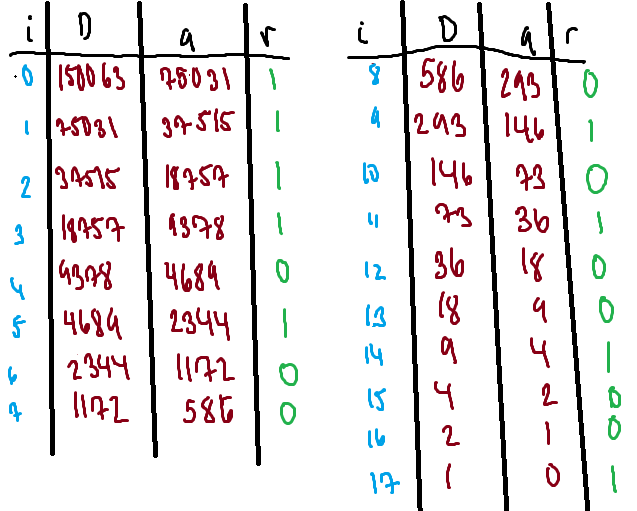
Part B

Do the same as Part A and get the answer **11110010000**.



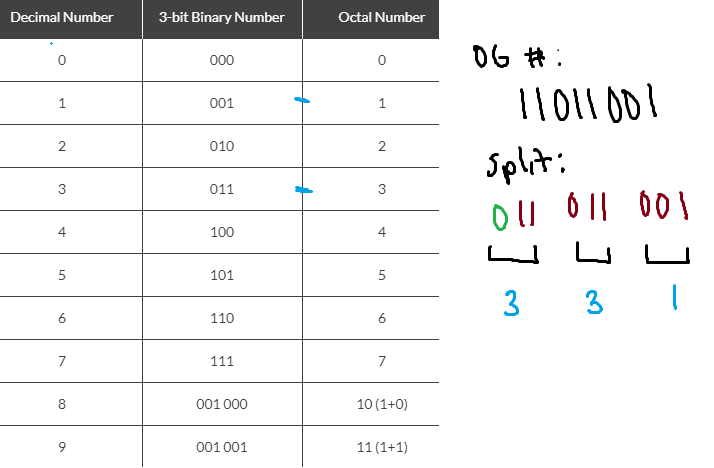
Part C

Do the same as Parts A and B and get the answer **100100101000101111**.



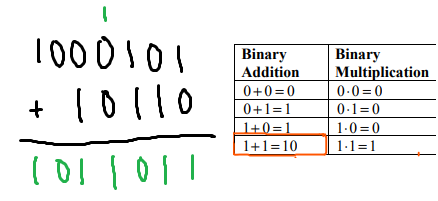
**10.** Without using base 10, convert 110110012 to base 8.

To convert a number from binary to base 8, split the binary number into groups of 3, starting from the rightmost digit. If the leftmost group does not have 3, prefix 0’s to them to make it a group of 3. Then, reference the octal number chart and input the corresponding octal number.



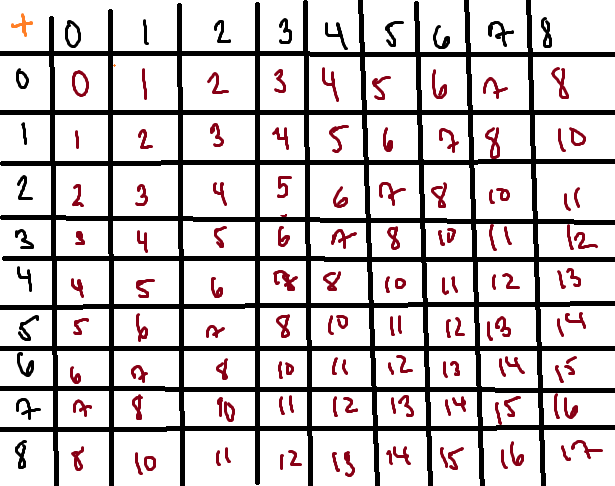
**11.** Evaluate the following expression without the use of base 10: .

Add just like normal addition. When you encounter 1 + 1 though, it equals 10 in binary, so carry the 1 to the next row and solve as normal (leaving the 0 in the answer). So, you get **1011011**.

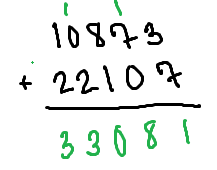


**12.** Evaluate the following expression without the use of base 10: .

For adding any base number other than 2 or 10, [make an addition table in any base using this video's reasoning.](https://www.youtube.com/watch?v=qGMdfomEONU) Make sure to check with an online addition table if possible (or look it up).



Then, line up the numbers to add like you normally would. The answer is **33801**.



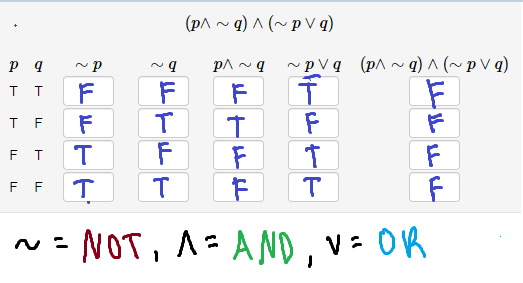
To do this quicker, realize there is a pattern whenever counting in other bases. For example, base 9 never includes exactly the number 9 (because you count from 0). There is a difference of 1 between the unreachable number 9 and the number 8, so you can add 1 to the normal addition.

What this means is that for example, in this problem, 7 + 3 is normally 10. However, since 9 doesn’t exist in base 9, you can add 1 (the difference, remember) to get the base 9 actual answer 11. This also means if you do something like, 8 + 5, you can get the normal answer 14 and add 1 to it to get the actual answer 15.

**13.** Complete the truth table for the following statement: .

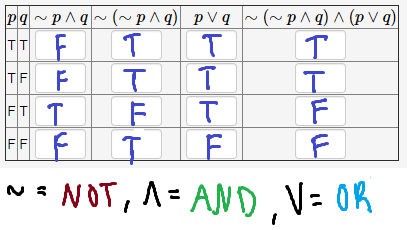
Correct table and make sure to list out all possible solutions and solve each part of the statement individually. Remember, in the boolean world, there’s only true and false.

* To complete this quickly, when solving “and” statements, any false means that the whole thing is false, and there’s only one possible true value for the row (true true).
* For any “not” statements, the output is the opposite of whatever the input was.
* For the “or” statements, the only false statement is when it's false false.



**14.** Complete the truth table and determine whether or not the following statement is a tautology, a contradiction, or neither: .

According to [this website](http://sites.millersville.edu/bikenaga/math-proof/truth-tables/truth-tables.html) and [this one as well](https://www.csm.ornl.gov/~sheldon/ds/sec1.1.html), *tautology* is a formula which is “always true” -- that is, it is true for every assignment of truth values to its simple components. The truth doesn’t rely on the values of the individual statements, but the logical structure (i.e. You will get an A in this class or you will not). On the other hand, a *contradiction* is a formula which is “always false”, and the falsity lies in the logical structure of the statement (i.e. I always tell lies).



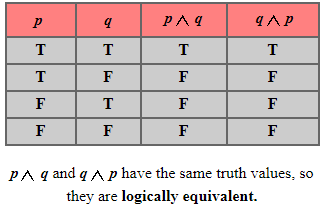
And the statement is **neither**.

**15.** Complete the following truth table by filling the blanks with T or F as appropriate. Are

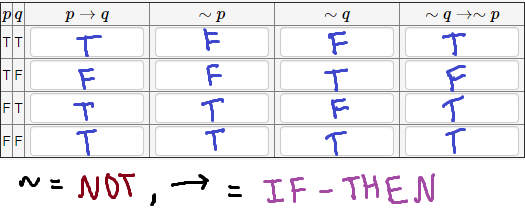
“” and “” logically equivalent, not logically equivalent, or not logically comparable?

Remember the arrow is “if-then”. The only way a “if-then” statement can be false is when the “if” (or the first input) is true and the second input “then” is false.

* Two statements are *logically equivalent* if and only if their resulting forms are logically equivalent when identical statement variables are used to represent component statements. Ex:



“” and “” are **logically equivalent**.

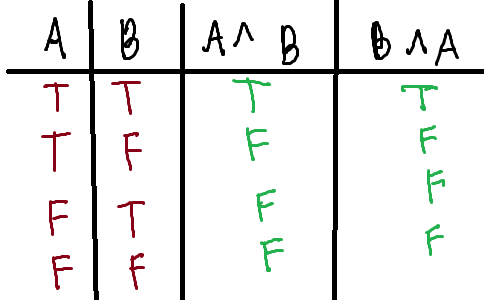


**16.** True or false?

1. “A and B” is logically equivalent to “B and A”
2. “A or B” is logically equivalent to “B or A”
3. “A iff B” is logically equivalent to “B iff A”
4. “AB” is logically equivalent to “BA”

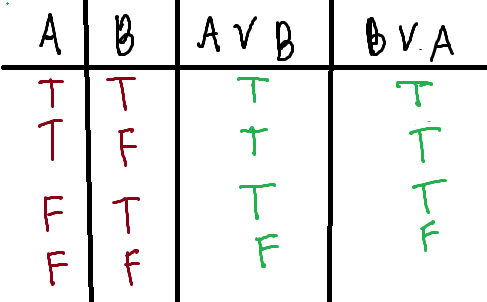
Part A

To solve this, you should draw truth tables for each and include all possibilities.



You can solve this with logic as well. The operation “and” requires both A and B to be true (which is only one possibility in this case, considering 2 variables to compare). This has to be **true** because firstly, both sides are the same and secondly, both have to be true for true true.

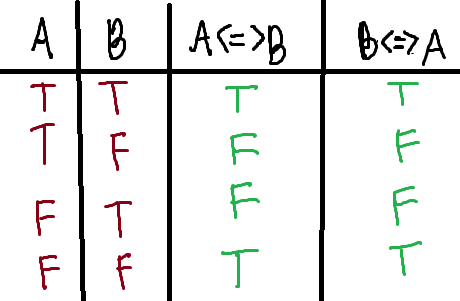
Part B



Since both sides are the same, this is **true**. Also notice that as long as one of the boolean values for A and B is true, the output of the statements is true.

Part C

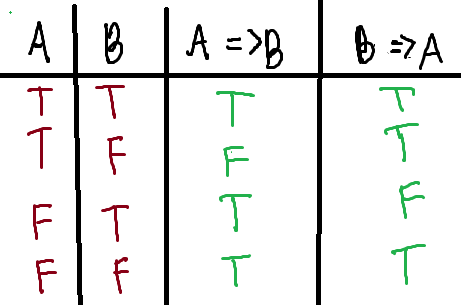
Note that “iff”, or if and only if, is the symbol.



This is **true**. Remember that “iff” statements are only true if both inputs match. For example, true true and false false both have an output of true. Think, “A is true if and only if B is true”. AKA, the A and B values can be exchanged in that sentence and it will still be true.

Part D

Remember, the signals that this is an “if-then” statement.



If a condition doesn’t hold, the statement doesn’t apply, so technically it’s a true statement. If a condition does hold and the statement doesn’t match the “then”, it’s a false statement.

For example, if a person makes a statement, “if this card is a heart, then it is a queen”. He’s only lying when the card is a heart (the condition met = true) and the card is a king (the statement doesn’t match = false), thus representing that order DOES matter in this, so this one is **false**, also because the values on both sides don’t match.

**17.** True or false?

1. If , then
2. If , then
3. If , then
4. If , then

Part A

You prove by making a counterexample. Think, if the statement was true, then ALL values of *x* must be valid for that equation. For an if-then statement like this, you assume the if part to be true (and choose a value of *x* to make it true) and see if it proves true for the then part.

However, this statement is **false**. Why?

Let , then , but in this case, this makes NOT true because if you plug in -6, you’ll get and that is not true, indicating that NOT ALL values of *x* are valid for the equation, this making the entire statement true.

Part B

This is **true**. Do a direct proof. Let . Multiply both sides as follows:

and then

By the transitive property of inequalities, you can assume is true.

Part C

Remember, this is NOT the same as Part B. It’s saying that if , then . This is different because you’re assuming the “if” part to be true. This is **false**, and you can prove this with a counterexample like Part A:

Let’s say . It fits the condition because , which is true. However, -6 is NOT greater than 5.

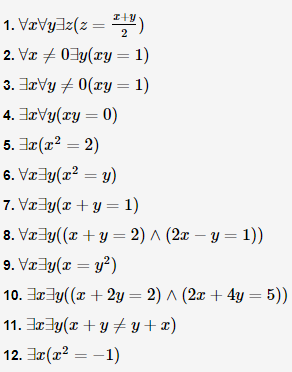
Part D

This is **true**, because when doing these, you always assume the “if” statement to be true. Think about it, the only values of *x* squared that are less than 25 are -4, -3, -2, -1, 0, 1, 2, 3, and 4. Since the range of possible values is finite, it’s possible to list them all out, then prove at least 0, 1, 2, 3, and 4 and saying that their negative counterparts have the same outputs, thus showing that a, they’re all under 5.

Probably, the accepted proof would be using the proof of contrapositive.

First, you must rewrite the statement accordingly (so from to ). The new rewritten statement is if , then . You already proved this in Part B (even though the sign is greater than or equal to, it still applies if you think about it).

**18.** Determine the truth value of the following statements if the universe of discourse of each variable is the set of real numbers (ℝ).



Note that ℝ includes both rational and irrational numbers.

Part 1

For all () real number values of *x* and *y*, there exists () *z* where *z* = .

Let . You can assume that because they both represent ALL real numbers. So, you can assume that the new predicate (*P(x)*) that . Since they’re both from the set of real numbers, there definitely exists a value of *z* that is the same (equals to) *x*. For example, let . Since *z* is from the same set, it can also equal 6. So, this is **true**.

Also true because either way, adding real numbers together and dividing them by 2 always results in a real number.

Part 2

For all *x* where , there exists *y* where .

Set to set *y* in terms of *x*. Then, . Thus, this is **true**.

(If this was a set of integers, this would not be possible since real numbers includes rational numbers, which includes fractions.)

Proof: If , , which is correct, where ⅕ is the *y* value to make the expression true. This statement basically means for all values of *x*, there is a *y* to match each of them to make the expression true. You could also say if , which is also true.

Part 3

There exists *x* for all of where .

This is **false**. What this statement really means is that there is at least 1 value of *x* that can multiply with ANY number of *y* to equal 1, in which this is impossible for this. You can prove this with a counterexample:

Let . If you multiply that with any real number (*y*), you won’t get 1 BECAUSE there only exists ONE solution for *y*, not all, and that is ¼.

Part 4

There exists *x* for all of *y* where .

This is **true** because if *y* is ANY real number ever, multiplying it by 0 will definitely get 0, no matter if *y* is 5 or pi. For example, and so on and so forth. Or, another interpretation: there exists ONE value of *x* that when multiplied with ANY VALUE of *y* to make it equal 0.

Part 5

There exists *x* where .

You can solve this equation for *x* to get , which proves there is at least one value of *x* that works, thus making thus **true**.

Part 6

For all of *x*, there exists *y* where .

This means, for every real number *x*, does there exist some other number that is equal to ? This is **true** because you can put any real number for *x*, square it, and get a real number *y* back.

Part 7

For all real numbers *x*, there exists a real number *y* where .

This is **true** because you can put any real number for *x*, then find a value of *y* that makes it equal to 1. Let , so , which comes out to be .

Part 8

For all real numbers *x*, there exists a real number *y* where AND .

For this, you must test BOTH expressions to be true before the whole thing can evaluate to true. You need to have the same value of *x* and *y* in the first equation to apply to the second.

Let’s try out real numbers for *x* and *y* that satisfies the first equation. First, choose any real number *x*, like 0. , so in this case, *y* has to be 2. Does this work for the second equation? No, because .

So, this is **false**.

Part 9

For all real numbers *x*, there exists a real number *y* where .

This is **false** because not every real number has a square root *y*. A counterexample of this is to replace *x* with any negative real number.

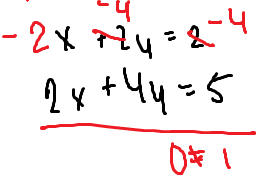
Let’s say, . So, the equation would be . The statement states that a real number *y* must exist that leads to the value of *x*. So, if you solve this for *y*, there is no value for *y* that can lead to a -1 because of the square. Squaring something always results in a positive number.

Part 10

There exists a real number *x* and a real number *y* where AND .

Having two “there exists” basically means showing that there are values for *x* and *y* that make this true (an example). The only time this would be false is if there are NO examples with *x* and *y* being valid or the values for one equation don’t satisfy the other equation.

For this one here, you create a system of equations and try to solve for one variable by cancelling the other. For this one, it’s impossible to find values for *x* and *y* because they will cancel each other out easily:



So, this is **false**.

Part 11

There exists real numbers *x* and *y* where .

Addition is commutative, where . This immediately disproves the above statement, because whatever values for *x* and *y* are, they will equal each other all the same, so this is **false**.

Part 12

There exists a real number *x* where .

This is **false** because any number squared NEVER equals a negative.

**19.** Negate the following statement: If Mary fails her classes, then she cannot graduate.

p: Mary fails her classes q: Mary can graduate

1. Write the statement in formal logic.
2. Negate the logic.
3. Rewrite the negated logic in English.

Part A

It’s . The statement is the sentence in the problem written in English. Since *p* and *q* are defined, you can see that *q* is the opposite of the *q* in the statement, so you would use the ~ as the not for *q* (thus showing that it’s the opposite).

Part B

The negation of a conditional statement is NOT another conditional statement, but an AND statement. The typical negation of . For example, the following negate:

If you finish your dinner, then you can play outside.

You finished your dinner but (another word for and) you cannot play outside.

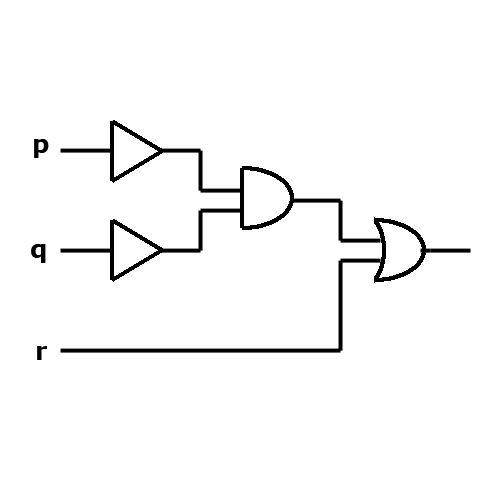
The first statement is a promise. The second is a lie. If you say the first one, the second one can’t be true. Refer to the lie example in the notes. Satisfy the assumption (the first part) and negate the conclusion (the second part), basically.

So for this, leave *p* as is. Convert the if-then operation to and, then negate the *q*. Negating the already-negated *q* would leave *q* by itself, getting the answer***p* ^ *q***.

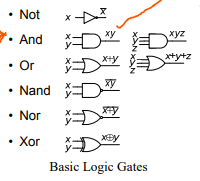
Part C

To rewrite the negated logic in English, simply write it using the logic and replace the variables with the English provided in the problem: **Mary fails her classes and she can graduate**.

**20.** Determine the correct logical statement based on the following circuit diagram. Which of the following is the correct logical statement?



To do this, refer to the following logic gates key:



The answer is **(*pq*)*r***. CHECK LATER WHY