CSE 20 Spring 2024
Practice for Test 1 Solutions

1. Modeling, Sets and Functions, Algorithms In machine learning, clustering can be used to group similar data for prediction and recommendation. For example, each Netflix user's viewing history can be represented as a n-tuple indicating their preferences about movies in the database, where n is the number of movies in the database. Each element in the n-tuple indicate the user's rating of the corresponding movie: 1 indicates the person liked the movie, -1 that they didn't, and 0 that they didn't rate it one way or another.

Consider the following algorithm for determining if a user's viewing history represents strong opinions on many movies.

Determine if user's ratings tuple encode strong opinions

```
1  procedure opinion((r_1,...,r_n): a n–tuple of ratings; c: a nonnegative integer)
2  sum := 0
3  for i := 1 to n
4     if r_i ≠ 0
5        sum := sum + 1
6  return sum ≥ c
```

1. When $n = 3$, describe the set of all n-tuples representing user ratings
   (a) By the roster method.

   There are 27 3-tuples whose elements are each -1, 0, or 1. We can list them as follows.

   $\{(-1,-1,-1),(-1,-1,0),(-1,-1,1),(-1,0,-1),(-1,0,0),(-1,0,1),(-1,1,-1),(-1,1,$
   $(0,-1,-1),(0,-1,0),(0,-1,1),(0,0,-1),(0,0,0),(0,0,1),(0,1,-1),(0,1,0),(0,1,1),$
   $(1,-1,-1),(1,-1,0),(1,-1,1),(1,0,-1),(1,0,0),(1,0,1),(1,1,-1),(1,1,0),(1,1,1)\}$

   (b) With set builder notation.

   Using set builder notation, we can specify the constraints on each component of the 3-tuple:

   $$\{(x,y,z) \mid x \in \{-1,0,1\} \text{ and } y \in \{-1,0,1\} \text{ and } z \in \{-1,0,1\}\}$$

   (c) Using a recursive definition.

   Let's call the set of ratings of three movies R_3. We can define this set recursively by listing each of the finitely many tuples in R_3 in the basis step and then defining the recursive step to be: when $u \in R_3$, so is u.

2. What are the possible return (output) values of this algorithm?

   Since the algorithm returns the result of a comparison, the possible values are True or False.

3. **Trace** the computation of $opinion((-1,0,0,0,1,1,-1),2)$.

The computation of $opinion((-1,0,0,0,1,1,-1),2)$ first (in line 2) initializes the variable *sum* to 0 and then steps through each component of the tuple in the for loop in line 3, checking (in line 4) if the value is zero or not. The first component of $(-1,0,0,0,1,1,-1)$ is nonzero so line 5 increments *sum*, which now has value 1. When the for loop checks the components at positions 2,3,4, the *sum* variable is not updated. However it still does not increment the value in sum. When the for loop gets to each of positions 5,6,7, the nonzero value of the elements at these positions means the if branch of the condition is followed and the *sum* variable is incremented three times, to get to the value 4. In line 6, the comparison $sum \geq c$ evaluates to $4 \geq 2$, which is True. Thus, $opinion((-1,0,0,0,1,1,-1),2)$ returns True.

4. Give an example of a nonnegative integer c so that, no matter which n-tuple $(r_1,\ldots,r_n)$ we consider, $opinion((r_1,\ldots,r_n),c)$ will have the same value. What value is it?

The nonnegative integer $c = 0$ is an example. The value of *sum* is initialized to 0 in line 2. The values of the components n-tuple $(r_1,\ldots,r_n)$ determine whether the value of *sum* ever increases in the form loop lines 3-5; the value can never decrease. Thus, no matter the choice of n-tuple, the comparison $sum \geq 0$ will evaluate to True.

2. Sets and Functions

1. Give a recursive definition for the set $\mathbb{N}$.

The set of nonnegative integers can be defined recursively as follows:
Basis step: $0 \in \mathbb{N}$
Recursive step: If $x \in \mathbb{N}$ then $x + 1 \in \mathbb{N}$.

2. Use the recursive definition from part (a) to give a recursive definition for the function with domain $\mathbb{N}$, codomain $\mathbb{N}$ which computes, for input i, the sum of the first i powers of 2. For example, on input 0, the function evaluates to 2^0, namely to 1. On input 1, the function evaluates to $2^0 + 2^1$, namely 3. On input 2, the function evaluates to $2^0 + 2^1 + 2^2$, namely 7.

Let's call this function $sumPow$. Its recursive definition will mirror the recursive definition of its domain, from part (a).
$sumPow : \mathbb{N} \to \mathbb{N}$
Basis step: $sumPow(0) = 1$.
Recursive step: If $x \in \mathbb{N}$ then $sumPow(x+1) = sumPow(x) + 2^{x+1}$.
In this function definition, we assumed that we have access to a definition of the power function to compute powers of two. In fact, we can define this function recursively as well:
$powTwo : \mathbb{N} \to \mathbb{N}$
Basis step: $powTwo(0) = 1$.
Recursive step: If $x \in \mathbb{N}$ then $powTwo(x+1) = 2powTwo(x) = powTwo(x) + powTwo(x)$.

3. Base expansions, Multiple representations

1. Compute the ternary (base 3) expansion of 28.

We can express 28 in terms of powers of 3 as

$$28 = 1 \cdot 27 + 1 = 1 \cdot 3^3 + 0 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0 = (1001)_3.$$

Therefore, by definition of base 3 expansion, $28 = (1001)_3$.

2. Compute the product of $(6A)_{16}$ and $(11)_{16}$, without converting either number to another base.

We use the usual algorithm for multiplication, except using hexadecimal symbols.

$$\begin{array}{r} 6A \\ \times\ 11 \\ \hline 6A \\ +6\ A0 \\ \hline 70A \end{array}$$

Thus, the product of $(6A)_{16}$ and $(11)_{16}$ is $(70A)_{16}$.

3. Confirm your answer for part (b) by converting $(6A)_{16}$ and $(11)_{16}$ to decimal, multiplying them, and converting the product to base 16.

Converting:

$$(6A)_{16} = 6 \cdot 16^1 + 10 \cdot 16^0 = 96 + 10 = 106$$

$$(11)_{16} = 1 \cdot 16^1 + 1 \cdot 16^0 = 16 + 1 = 17$$

Multiplying in decimal:

$$\begin{array}{r} 106 \\ \times\ \underline{17} \\ 742 \\ \underline{+1060} \\ 1802 \end{array}$$

Converting back to hexadecimal: by long division,

$$1802 = 112 \cdot 16 + 10$$
$$112 = 7 \cdot 16 + 0$$
$$7 = 0 \cdot 16 + 7$$

Thus, $1802 = (70A)_{16}$, agreeing with part (b).

4. How many bits will there be in the binary (base 2) expansion of 2020? Can you compute this without fully converting 2020 to base 2?

The leading coefficient will need to be in the column corresponding to the highest power of 2 less than or equal to 2020. Listing the powers of 2:

Exponent	0	1	2	3	4	5	6	7	8	9	10	11
Power	1	2	4	8	16	32	64	128	256	512	1024	2048

we see that the binary expansion of 2020 will be of the form $(1a_9 \cdots a_0)_2$ and will therefore have 11 bits.

5. Give an example of a number that can be represented in base 2 fixed-width 3, but not in base 2 expansion.

The number 0 can't be represented in binary (base 2) but is represented in base 2 fixed-width 3 $(000)_{2,3}$.

6. Give an example of a number that can be represented in base 2 expansion, but not in base 2, fixed-width 3 expansion.

The number 8 can be represented in base 2 as $(1000)_2$ but can't be represented in binary fixed-width 3 because it requires four bits.

7. Give the representation of -7 in sign-magnitude width 3 and 2s complement width 3. Then do the same for width 4.

The number -7 can't be represented in sign magnitude width 3 or in 2s complement width 3:

- The least number that can be represented in sign magnitude width 3 has representation $[111]_{s,3}$ and is -3.
- The least number that can be represented in 2s complement width 3 has representation $[100]_{2c,3}$ and is -4.

With width 4 we can use the following representations:

- In sign magnitude width 4, -7 has representation $[1111]_{s,4}$ because the leftmost bit is the sign bit and the magnitude of -7 has binary fixed width 3 representation $7 = (111)_{2,3}$.
- In 2s complement width 4, -7 has representation $[1001]_{2c,4}$ because the leftmost bit is the sign bit and rightmost bits need to represent $2^{4-1} - 7 = 1$ in binary fixed width 3, $(001)_{2,3}$.

8. Give the representation of 10.5625 in fixed-width base-2 expansion with integer width 4 and fractional width 9.

We need to write 10.5625 as

$$(a_3a_2a_1a_0.c_1c_2c_3c_4c_5c_6c_7c_8c_9)_{2,4,9}$$

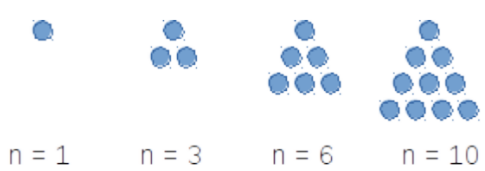
For the integer part: $10 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = (1010)_2$
For the fractional part: $0.5625 = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4}$ where

Exponent	-1	-2	-3	-4	-5	-6	-7
Power	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125

Thus,

$$10.5625 = (1010.100100000)_{2,4,9}$$

4. Circuits A triangular number (or triangle number) counts the objects that can form an equilateral triangle, as in the diagram below. The n^{th} triangular number is the sum of the first n integers, as shown in the following figure illustrating the first four triangular numbers (what is the fifth one?):



Design a circuit that takes four inputs x_0, x_1, x_2, x_3 and outputs True (T or 1) if the integer value $(x_3x_2x_1x_0)_{2,4}$ is a triangular number, and False (F or 0) otherwise. You

may assume that 0 is not a triangular number. (*Credit: UBC Department of Computer Science*)

Using the definition, we can write a table of values for the circuit we need to build, since the output will be one when $(x_3x_2x_1x_0)_{2,4}$ will be one of $1, 3, 6, 10, 15$.

x_3	x_2	x_1	x_0	Output
1	1	1	1	1
1	1	1	0	0
1	1	0	1	0
1	1	0	0	0
1	0	1	1	0
1	0	1	0	1
1	0	0	1	0
1	0	0	0	0
0	1	1	1	0
0	1	1	0	1
0	1	0	1	0
0	1	0	0	0
0	0	1	1	1
0	0	1	0	0
0	0	0	1	1
0	0	0	0	0

The output can be described using a compound proposition in DNF with five clauses (because there are five rows with output 1):

$$(x_3 \wedge x_2 \wedge x_1 \wedge x_0) \vee (x_3 \wedge \neg x_2 \wedge x_1 \wedge \neg x_0) \vee (\neg x_3 \wedge x_2 \wedge x_1 \wedge \neg x_0) \vee (\neg x_3 \wedge \neg x_2 \wedge x_1 \wedge x_0) \vee (\neg x_3 \wedge \neg x_2 \wedge \neg x_1 \wedge x_0)$$

We can use this DNF to draw a circuit that takes the four inputs and whose output is described.
Alternatively, a logically equivalent compound proposition is

$$output = [x_1 \wedge (x_0 \oplus x_2) \oplus x_3] \vee [\neg x_3 \wedge \neg x_2 \wedge \neg x_1 \wedge x_0]$$

Using this compound proposition, we can draw a circuit with fewer gates than the one described above.
Note: picture of circuits are omitted from sample solutions; please come to office hours or post to Piazza if you'd like to get feedback on a circuit you draw.

5. Circuits, Compound Propositions, Logical Equivalence

1. Draw a logic circuit that uses **exactly three** gates and is logically equivalent to

$$q \leftrightarrow (p \wedge r)$$

You may (only) use AND, OR, NOT, and XOR gates.

Since XOR gates are available, we can use a convenient relationship between XOR and $\leftrightarrow$:

$$q \leftrightarrow (p \wedge r) \equiv \neg(q \oplus (p \wedge r))$$

Implementing this compound proposition as a circuit, we will use one XOR gate, one AND gate, and one NOT gate, as required.
Note: picture of circuits are omitted from sample solutions; please come to office hours or post to Piazza if you'd like to get feedback on a circuit you draw.

2. Write a compound proposition which is logically equivalent to

$$(p \oplus q) \leftrightarrow r$$

You may only use the logical operators negation ($\neg$), conjunction ($\wedge$), and disjunction ($\vee$).

We apply several logical equivalences in turn: first, we can rewrite $\leftrightarrow$ in terms of $\wedge$ and $\rightarrow$:

$$(p \oplus q) \leftrightarrow r \equiv (\ (p \oplus q) \rightarrow r\ ) \wedge (\ r \rightarrow (p \oplus q)\ )$$

Now, we replace $\rightarrow$ by its equivalent disjunctive form: $HYP \rightarrow CONC \equiv (\neg HYP) \vee CONC$. Thus,

$$(\ (p \oplus q) \rightarrow r\ ) \wedge (\ r \rightarrow (p \oplus q)\ ) \equiv (\ \neg(p \oplus q) \vee r\ ) \wedge (\ \neg r \vee (p \oplus q)\ )$$

Last, we replace the XOR with equivalent compound propositions in terms of the other connectives: the XOR is true means one of its inputs is T and the other F; the XOR is false means its inputs are both T or both F.

$$(\ \neg(p \oplus q) \vee r\ ) \wedge (\ \neg r \vee (p \oplus q)\ ) \equiv$$
$$(\ ((p \wedge q) \vee (\neg p \wedge \neg q)) \vee r\ ) \wedge (\ \neg r \vee ((p \wedge \neg q) \vee (\neg p \wedge q))\ )$$

3. Find a compound proposition that is in DNF (disjunctive normal form) and is logically equivalent to

$$(p \vee q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (\neg p \vee q \vee r)$$

The given compound proposition is in CNF and indicates that the rows in the truth table with $p = F, q = F, r = T$, $p = F, q = T, r = F$, and $p = T, q = F, r = F$ are set to F. Thus, the truth table is

p	q	r	Output
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	T

To find a logically equivalence compound proposition in DNF, we represent the disjunction of landing in each of the rows that evaluate to T: the rows where $p = T, q = T, r = T$, or $p = T, q = T, r = F$, or $p = T, q = F, r = T$, or $p = F, q = T, r = T$, or $p = F, q = F, r = F$.

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$$
$$\vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$$

6. Translating Propositional Logic

p is "The display is 13.3-inch"
q is "The processor is 2.2 GHz"
r is "There is at least 128GB of flash storage"
s is "There is at least 256GB of flash storag
u is "There is at least 512GB of flash storage"

1. Are the statements

$$p \to (r \vee s \vee u) \quad , \quad q \to (s \vee u) \quad , \quad p \leftrightarrow q \quad , \quad \neg u$$

consistent? If so, translate to English a possible assignment of truth values to the input propositions that makes all four statements true simultaneously.

Yes, these statements are consistent. Consider the assignment of truth values

$$p = T \qquad q = T \qquad r = T \qquad s = T \qquad u = F$$

We evaluate the four compound propositions at this assignment to confirm that it makes all of them true simulatneously:

$$\begin{aligned} p \to (r \vee s \vee u) &= T \to (T \vee T \vee F) = T \to T = T \\ q \to (s \vee u) &= T \to (T \vee F) = T \to T = T \\ p \leftrightarrow q &= T \leftrightarrow T = T \\ \neg u &= \neg F = T \end{aligned}$$

Translating to English, we get "The display is 13.3-inch, the processor is 2.2GHz, there is at least 128GB of flash storage, there is at least 256GB of flash storage, but there is not at least 512GB of flash storage."

2. Consider this statement in English:

   > It's not the case that both the display is 13.3-inch and the processor is 2.2 GHz.

   Determine whether each of the compound propositions below is equivalent to the negation of that statement, and justify your answers using either truth tables or other equivalences.

   Possible compound propositions:

   - $\neg p \vee \neg q$
   - $\neg(p \to \neg q)$
   - $\neg(p \wedge q)$
   - $(\neg p \leftrightarrow \neg q) \wedge p$

The sentence can be translated to $\neg(p \wedge q)$. Its negation is then

$$\neg\neg(p \wedge q) \;\equiv\; (p \wedge q) \;\equiv\; \neg(p \to \neg q)$$

because $A \wedge \neg B \equiv \neg(A \to B)$ so $p \wedge q \equiv \neg(A \to \neg B)$.
This is also equivalent to $(\neg p \leftrightarrow \neg q) \wedge p$, as we can see from the last two columns of the the truth table below

p	q	$(\neg p \leftrightarrow \neg q)$	$(\neg p \leftrightarrow \neg q) \wedge p$	$p \wedge q$
T	T	T	T	T
T	F	F	F	F
F	T	F	F	F
F	F	T	F	F

3. Consider the compound proposition

$$(p \wedge q) \to (r \vee s \vee u)$$

Express the **contrapositive** of this conditional as a compound proposition. Then, give an assignment of truth values to each of the input propositional variables for which the original compound proposition is True but its **converse** is False.

The contrapositive of this conditional is

$$\neg(r \vee s \vee u) \rightarrow \neg(p \wedge q)$$

The converse of the original conditional is

$$(r \vee s \vee u) \rightarrow (p \wedge q)$$

Consider the assignment of truth values

$$p = T \qquad q = F \qquad r = T \qquad s = F \qquad u = F$$

We evaluate the original conditional and its converse with this assignment of truth values:

$$\text{Original} : (p \wedge q) \rightarrow (r \vee s \vee u) = (T \wedge F) \rightarrow (T \vee F \vee F) = F \rightarrow T = T$$
$$\text{Converse} : (r \vee s \vee u) \rightarrow (p \wedge q) = (T \vee F \vee F) \rightarrow (T \wedge F) = T \rightarrow F = F$$

7. Evaluating predicates, Evaluating nested predicates

1. Over the domain $\{1, 2, 3, 4, 5\}$ give an example of predicates $P(x), Q(x)$ which demonstrate that

$$(\forall x P(x)) \vee (\forall x Q(x)) \qquad \text{is not logically equivalent to} \qquad \forall x(\ P(x) \vee Q(x)\ )$$

We will define predicates which make one of the statements False and the other True. Consider

x	$P(x)$	$Q(x)$
1	T	F
2	F	T
3	T	F
4	F	T
5	T	F

Evaluating the statements: $\forall x(P(x) \vee Q(x))$ is True because each row in the table has at least one of $P(x)$, $Q(x)$ evaluate to True. On the other hand, neither $\forall x P(x)$ nor $\forall x Q(x)$ is True, because each predicate at least one domain element where it evaluates to False (for counterexamples: $P(2)$ is False and $Q(1)$ is False). Since $F \vee F = F$, $(\forall x P(x)) \vee (\forall x Q(x))$ is False. Since there is some definition of the predicates P and Q where the two statements

$$(\forall x P(x)) \vee (\forall x Q(x)) \quad , \quad \forall x(\ P(x) \vee Q(x)\ )$$

evaluate to different truth values, they are not logically equivalent

2. Over the domain $\{0, 1, 2\}$, give an example of predicates $P(x), Q(x)$ for which all of these statements are true:

$$\forall x(P(x) \to Q(x))$$

$$\exists x\, P(x)$$

$$\exists x\, \neg P(x)$$

$$\exists x\, \neg Q(x)$$

We will define predicates using their tables of values:

x	$P(x)$	$Q(x)$
0	T	T
1	F	T
2	F	F

We now prove that each of the statements is True:

- To prove the universal statement $\forall x(P(x) \to Q(x))$, we evaluate the conditional at each element in the domain: At 0, $P(0) \to Q(0) = T \to T = T$; at 1, $P(1) \to Q(1) = F \to T = T$; at 2, $P(2) \to Q(2) = F \to F = T$. Since the body of the quantification is True at all element in the domain, $\forall x(P(x) \to Q(x))$ is True.
- To prove the existential statement $\exists x P(x)$, we need a witness in the domain where $P(x)$ evaluates to True. The witness is 0, since the first row in the definition table gives $P(0) = T$.
- To prove the existential statement $\exists x \neg P(x)$, we need a witness in the domain where $\neg P(x)$ evaluates to True. The witness is 1, since the second row in the definition table gives $P(1) = F$ so $\neg P(1) = T$.
- To prove the existential statement $\exists x \neg Q(x)$, we need a witness in the domain where $\neg Q(x)$ evaluates to True. The witness is 2, since the third row in the definition table gives $Q(2) = F$ so $\neg Q(2) = T$.

8. Evaluating predicates, Evaluating nested predicates Recall that S is defined as the set of all RNA strands, strings made of the bases in $B = \{\mathtt{A}, \mathtt{U}, \mathtt{G}, \mathtt{C}\}$. Recall the definition of the following predicates: $F_{\mathtt{A}}$ with domain S is defined recursively by:

Basis step: $F_{\mathtt{A}}(\mathtt{A}) = T$, $F_{\mathtt{A}}(\mathtt{C}) = F_{\mathtt{A}}(\mathtt{G}) = F_{\mathtt{A}}(\mathtt{U}) = F$

Recursive step: If $s \in S$ and $b \in B$, then $F_{\mathtt{A}}(sb) = F_{\mathtt{A}}(s)$

$P_{\mathtt{AUC}}$ with domain S is defined as the predicate whose truth set is the collection of RNA strands where the string $\mathtt{AUC}$ is a substring (appears inside s, in order and consecutively)

L with domain $S \times \mathbb{Z}^+$ is defined by, for $s \in S$ and $n \in \mathbb{Z}^+$,

$$L(s,n) = \begin{cases} T & \text{if } rnalen(s) = n \\ F & \text{otherwise} \end{cases}$$

1. Give a witness for $\exists s_1 \, \exists s_2 \, (L(s_1, 4) \wedge L(s_2, 4) \wedge \neg(s_1 = s_2))$ where S is the set of RNA strands, $L(s,n)$ is a predicate with domain $S \times \mathbb{Z}^+$ that is true when s has length n

In English, the statement can be translated to "There are strands s_1, s_2 where each has length 4 and the strands are not equal." A witness is $s_1 = \texttt{CCCC}$, $s_2 = \texttt{GGGG}$. Since each of s_1, s_2 is an element of S, s_1 has length 4, s_2 has length 4 and these two strings are not the same.

2. Give a counterexample that disproves $\forall i\,(\neg L(\texttt{ACU}, i))$

In English, the statement can be translated to "No positive integer is the length of the strand ACU."
A counterexample that helps us disprove this statement is 3. This is a positive integer and is the length of ACU so $L(\texttt{ACU}, 3)$ is True, hence $\neg L(\texttt{ACU}, 3)$ is False.

3. Determine which of the following statements is True or False (you do not need to prove your answer).

   (a) $\exists n \in \mathbb{Z}^+\ \forall s \in S\ (P_{\texttt{AUC}}(s) \to \neg L(s, n))$

   In English, this statement is saying that there is a positive integer that is not the length of every RNA strand with AUC. This is **true**, as we can see with the witness $n = 1$.

   (b) $\exists n \in \mathbb{Z}^+\ \forall s \in S\ (P_{\texttt{AUC}}(s) \wedge \neg L(s, n))$

   In English, this statement is saying that there is a positive integer that is not the length of any RNA strand and that each RNA strand has AUC as a substring. This is **false**, because when we consider any positive integer, we'll be able to find a counterexample strand that does not have AUC as a substring and so makes the conjunction false.

   (c) $\forall s \in S\ \exists n \in \mathbb{Z}^+\ (P_{\texttt{AUC}}(s) \vee \neg L(s, n))$

   In English, this statement is saying that for each RNA strand we can find some positive integer so that at least one of "the strand has AUC as a substring" and "this number is not the length of this strand" is true. This is **true**, because given any RNA strand, we can pick a witness positive integer by choosing a number that is not its length.

   (d) $\forall s \in S\ \exists n \in \mathbb{Z}^+\ (L(s, n) \to \neg P_{\texttt{AUC}}(s))$

   In English, this statement is saying that for each RNA strand, there is a witness positive integer which makes the conditional statement "if the RNA strand has this length then it does not have AUCas a substring" true. This is **true**, because given any RNA strand, we can pick a witness positive integer by choosing a number that is not its length which will make the hypothesis of the conditional statement false and therefor the conditional will be true.