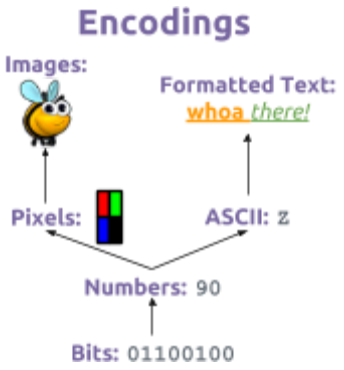
STUDENT EXAMPLE_____ Period _____ Date _____

Practice PT: Encode an Experience

Project Description

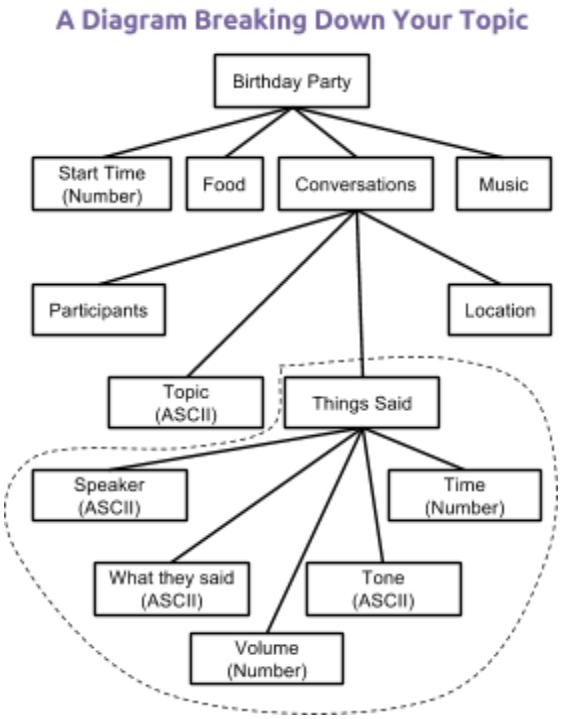


In this unit you have learned how bits are used to represent numbers, text, and images. In each case you were using a protocol or an **encoding** to interpret a sequence of 0s and 1s as meaningful information. Figuring out clever ways to encode information is one of the things computer scientists need to think about in many different contexts. There was a time before we had digital images, or music, or streaming television, or online shopping. Someone had to figure out how to encode those things in order get them onto and into computing devices. For this project the challenge is to **design an encoding for an experience of your choosing**.

Step 0: Choose Your Experience

You should feel free to choose any type of experience you wish. In this example we have encoded what it's like to go to a birthday party. Your challenge is to identify the critical aspects of that experience and determine how that can be translated into a binary protocol. Here are some guidelines for choosing your experience:

- **Pick a general category of experiences.** While you are basing your encoding off your experience, everyone should be able to use it. "Going to a baseball game" is much better than "the last baseball game I went to"
- **Pick something that isn't already "digital".** "Going to a concert" is a better choice than "making a playlist" since the second one is already primarily a digital process.
- **Make sure to pick an event you have a detailed understanding of.** You don't need to pick an experience you've done before. It's fine to pick "Walking on the moon" if you think you know enough to write that, but "Walking to school" probably also has a lot of interesting information to consider.



Step 1: Break Down Your Topic

You will be creating a diagram that breaks your topic into components, eventually getting down to the things you could represent with ASCII or numbers. This is sometimes called **Top-Down Design** - a design process that begins by specifying complex pieces and then dividing them into successively smaller pieces.

Begin by **writing down the experience you chose**. Then imagine you are telling your friend about the experience. What are the major categories of information you would want to tell your friend about so they could "relive" the experience? What will they want to know about? In this example we picked the start time, food, music, and conversations that were at the party, but there are certainly many others we could have added.

Once you have your list of components **consider whether any of them can be immediately encoded into ASCII text or numbers**. For a birthday party an example might be the *starting time* which could be encoded as a number. For these components write either *ASCII* or *Number* below them to indicate how they could be encoded.

For the rest of your components you'll need to **keep splitting them up**. In our example we couldn't directly encode a conversation, so we kept breaking it down, dividing it into *participants*, a conversation *topic*, the *things said*, and a *location*. We even decided that the *things said* would need to be further divided to truly capture the experience. **Keep**

dividing until every path in your design tree ends in ASCII or a number.

This example is incomplete, and there's still a lot of interesting questions left. How would you go about encoding the food at a party? Where's the information about the guest list, or when everyone arrived? **You'll need to decide what's important to include and how to encode it.**

Step 2: Consult with a Partner

Once you think you have broken down your topic successfully, share your design diagram with a partner. Look over their work and consider the following questions:

- Is every component either split into more components or marked as ASCII / number?
- Do you agree with the choices your partner has made? Is there anything missing?
- Is there anything that's particularly clever or gives you ideas for your own encoding?

Share your thoughts with your partner and use the results of this consultation to finalize your encoding.

Step 3: Develop a Detailed Encoding

Pick **one** aspect of your experience to encode that shows one of the bottom levels of your design tree. We've circled a group like this in our example - the *Things Said* group of information. You will develop a detailed protocol explaining precisely how that portion of your diagram would be encoded.

Place the name of the high level component at the top and briefly explain what it is and how it is used. Then for each component list the following information:

- **Component:** name of the component
- **Type:** either ASCII or number
- **Number of bits:** Depending on the type there are different ways to complete this
    - **Numbers**: how many bits long this number will be and the range of values this provides
    - **ASCII**: you will always need 8 bits per character for text, but you need to think about how big or small this text needs to be. This can be tricky in some cases since you're effectively picking an upper limit on the size of the text. Then include the range of characters that could be included.
- **Description / Comments:** Explain a bit more what this component is and justify your choices for encoding it the way you did. Provide any other clarifying comments necessary.

Things Said: Every time a person says something in a conversation their comment, question, etc. is encoded. By combining all of the "Things Said" it is possible to recreate the conversation.

Component	Type	Number of bits/range	Description / Comments
Speaker	ASCII	256 bits (0 - 32 chars)	Name of the person speaking. 32 characters should be enough for most names. It is important to know who said what.
What they said	ASCII	8000 bits (0 - 1000 chars)	The text of what the speaker said, used to know the actual contents of the conversation. I figure that a person wouldn't speak for more than 1 minute at a time. I found that a typical American speaks at most 150 words per minute, and on average words have 5 characters. With spaces that is 900 characters so 1000 is sufficient.
Volume	number	7 bits (0 - 127 dB)	Measured in decibels. A rock concert is 125 decibels so the range of a 7 bit number (0-127) is enough. Helps gauge how loud the party was.
Tone	ASCII	128 bits (0 - 16 chars)	How the person said something e.g. "laughing" or "serious". This will only be a short one or two word description to help reconstruct the tone of the conversation.
Time	number	16 bits (0-65,535 secs) (0 to 18 hours)	When the thing was said, measured in seconds since the beginning of the party. 16 bits provides (0-65,535) seconds or a little more than 18 hours (long party). This is important for knowing the order of comments.

Name(s)_____ Period _____ Date _____

Reflection Questions

Written Reflection After completing your project respond to each of the following reflection questions. Your responses to each prompt should be **no longer than 300 words.**

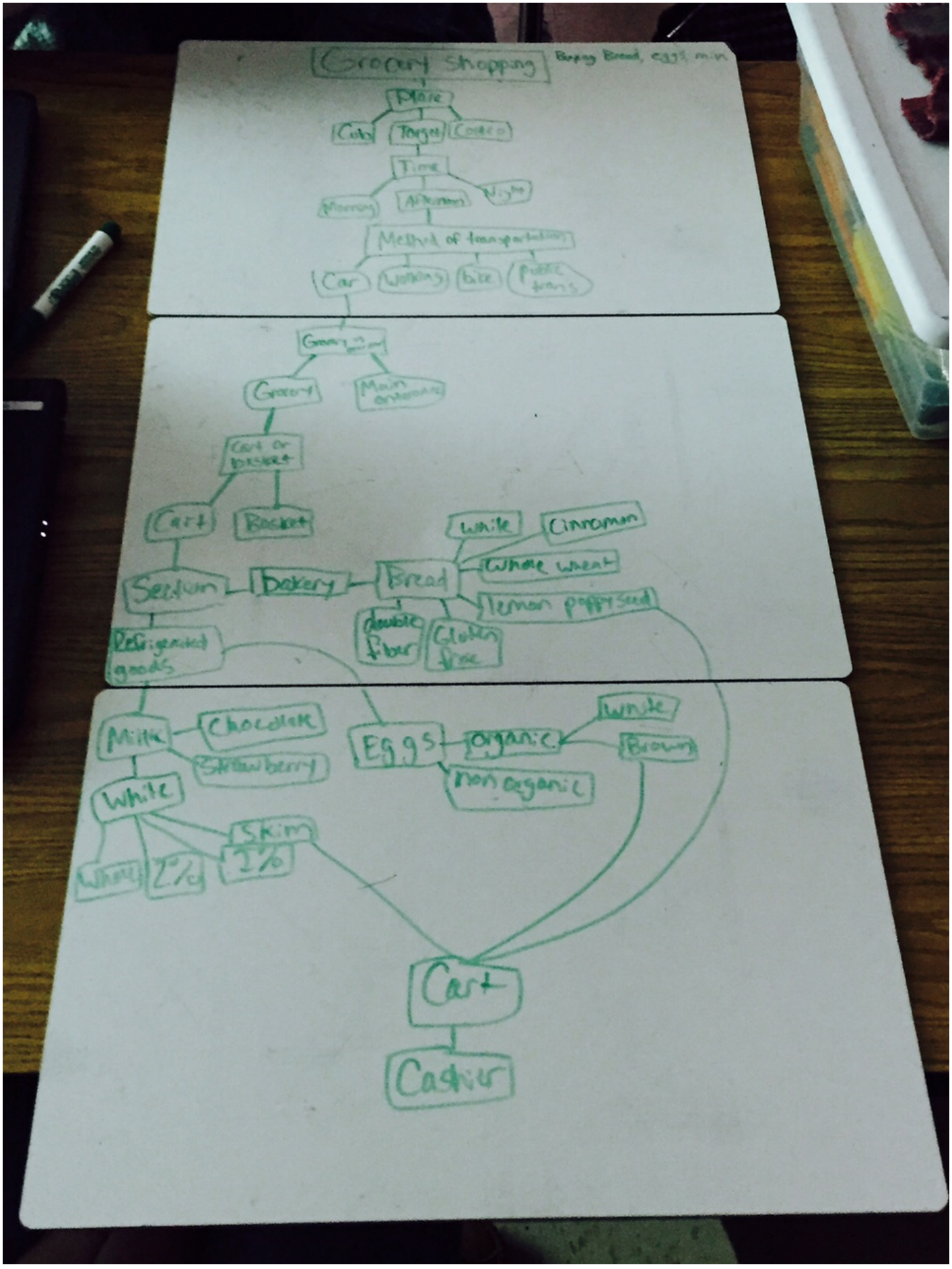
1. Why is it necessary to break down experiences (or other complex information in general) into their component parts if we want to represent them on a computer?

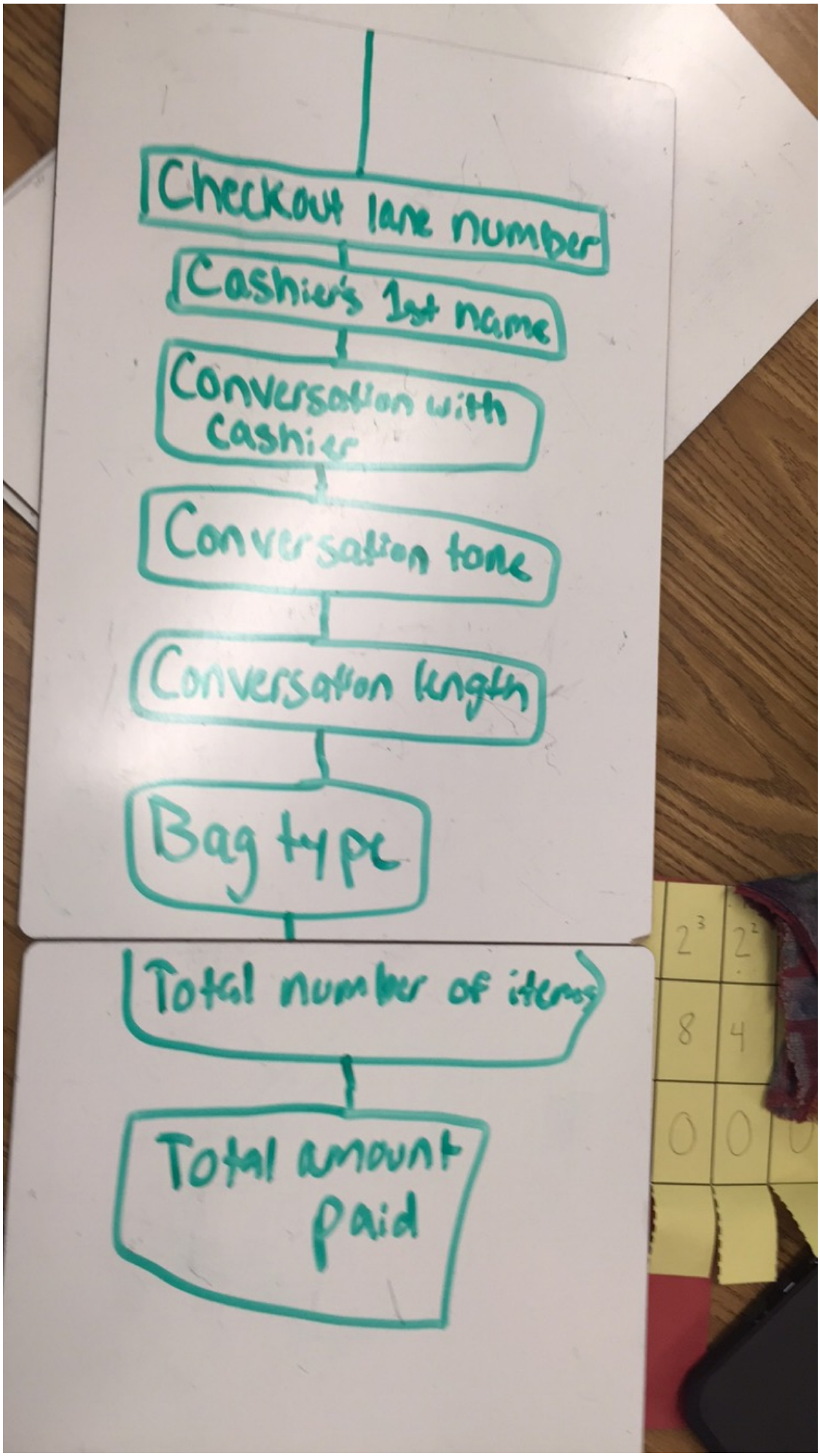
To fully give someone the same experience that you had, it is necessary to give very exact and specific details. The further an experience can be broken down into its component parts, the more specific the details about it can be. This gives the most accurate rendition of the experience that can be represented on a computer.
<3 THE GROCERY GALS.

2. In your detailed encoding you decided how many bits should be used to represent each component, a choice which impacts the maximum number or amount of text you are able to encode. Choose a component for which this decision was difficult to make. Justify your final decision regarding the number of bits to use for encoding this component, and explain the implications of your choice.

Choosing the right bit size for length of conversation with the cashier was slightly difficult. We all are people who don't really enjoy making conversation with cashiers, so none of us immediately knew how long a conversation typically lasts with a cashier. We reasoned that 15 minutes would be the max length of a conversation with a cashier, since the person buying groceries needs to get their groceries home to be refrigerated, and the cashier would most likely have other customers to help. However, we increased the bit count to 5 bits (up to 31) to reduce the chance of having overflow issues.
<3 THE GROCERY GALS.

3. Describe how getting feedback from a classmate impacted the structure or complexity of your encoding. As part of your explanation specify at least one aspect of your encoding that was directly influenced by this collaboration.


Grocery shopping
Buying Bread, eggs, milk
Store
Cub
Target
Costco
Time
Morning
Afternoon
Night
Method of transportation
Car
Walking
Bike
Public trans
Grocery
Main entrance
Cart or basket
Cart
Basket
Section
Bakery
Refrigerated goods
Bread
White
Cinnamon
Whole wheat
lemon poppyseed
double fiber
Gluten free
Milk
Chocolate
Strawberry
White
Skim
Whole
2%
1%
Eggs
organic
non organic
White
Brown
Cart
Cashier


Checkout lane number
Cashier's 1st name
Conversation with cashier
Conversation tone
Conversation length
Bag type
Total number of items
Total amount paid

Name(s)_____ Period _____ Date _____

Detailed Protocol

Checking out at the Grocery Store

Component	Type	Number of bits/range	Description / Comments
Checkout lane number	number	5 bits (1-30)	Number of the checkout lane selected. Target has about 30 checkout lanes, so 5 bits should be sufficient.
Conversation with cashier	ASCII	8000 bits (up to 1000 characters)	Text of what was said in the conversation. Conversation with a cashier is typically shorter than a normal conversation, so 500 characters should be sufficient
Bag type	ASCII	56 bits (up to 7 characters)	Which type of bag is chosen for groceries. Plastic is 7 characters long and paper is 5. Allowing 7 characters covers both choices.
Cashier's first name	ASCII	80 bits (up to 10 characters)	The cashier's first name, as displayed on their name tag. Using 10 characters should be enough for most first names.
Tone of conversation	ASCII	80 bits (up to 10 characters)	Adjective that most accurately describes the tone of conversation with the cashier. Maximum length of adjective is 10 characters, as this will cover adjectives.
Length of conversation	numbers	5 bits (up to 31 minutes)	How long the conversation was in minutes. 15 minutes should be enough for a single conversation, as cashiers are working and don't have time for a long conversation.
Total number of items	numbers	4 bits (up to 15 items in total)	Total number of items purchased. 5 of each item seems reasonable for one trip.
Total amount paid	numbers	7 bits(up to $127)	Total price, in dollars, of all the items bought. Since we are only buying milk, eggs, and bread, the total should (hopefully) not be more than $100.

Name(s)_____ Period _____ Date _____

Rubric

Component	1	2	3	Score
Diagram: Complete	Few aspects of the experience have been accounted for or the diagram is incomplete.	The experience has been generally broken down but some key elements may have been missed.	The experience has been thoroughly broken down into its component parts. All critical elements have been accounted for.	3
Diagram: Proper Encodings	Many components have been improperly assigned number / ASCII or should have been further subdivided.	Most components are subdivided when necessary and are reasonably assigned ASCII / number.	Components are consistently subdivided when necessary and are reasonably assigned ASCII / number.	3
Detailed Protocol: Number of Bits	The protocol frequently uses an unreasonable number of bits to encode each component.	The protocol generally uses a reasonable number of bits to encode each component.	The protocol consistently uses a reasonable number of bits to encode each component.	3
Detailed Protocol: Description / Comments	Most components require more explanation of how they will be interpreted or why they are included.	Some components could benefit from further explanation of how they will be interpreted or why they are included.	Nearly all components provide rich explanations of how they are interpreted and why they are included.	3
Reflection: Why Split Up Components	The response reflects an incomplete understanding of why topics must be broken into their components in order to be represented by a computer.	The response reflects a partial understanding of why topics must be broken into their components in order to be represented by a computer.	The response reflects a deep understanding of why topics must be broken into their components in order to be represented by a computer.	3
Reflection: Most Difficult Component	The response lacks details describing how the decision of the number of bits was made and does not justify the final decision.	The response provides some detail on how the chosen number of bits was selected or provides a partial justification for the decision made.	The response provides rich details for how the chosen number of bits was selected and provides justifications for the decision made.	3
Reflection: Collaboration	The response demonstrates little or no exchange of feedback between partners.	The response explains that partners shared feedback. Little attention is given either to identifying the details of that feedback or how it was incorporated into the encoding.	The response provides details of feedback exchanged between partners and ties it to specific features of the encoding.	3

Feedback Summary and Response

Summarize your feedback here:

Respond to your feedback here: