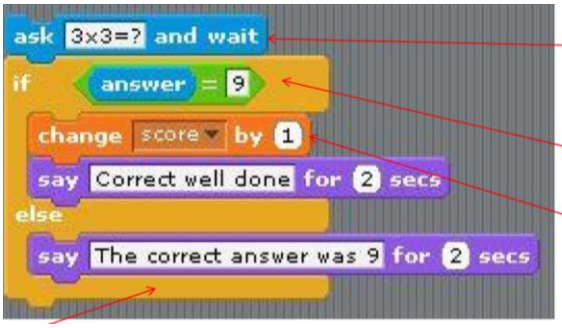
<u>Part I</u>: QUIZ SHOW – Q & A **Project 2 Computer Science**

Create a program which asks the user basic questions—or any other trivia question you like as long as they are all in the same subject of questions. If the user types the right answer they will be congratulated and their score will be increased by 1. If the user types the wrong answer they will be told the correct answer and will not receive an increase in score. After each answer is given the current score will be reported to the user.



Ask a question and wait for the user to answer
If user answer is = to 9 then increase score variable by 1 and say correct well done

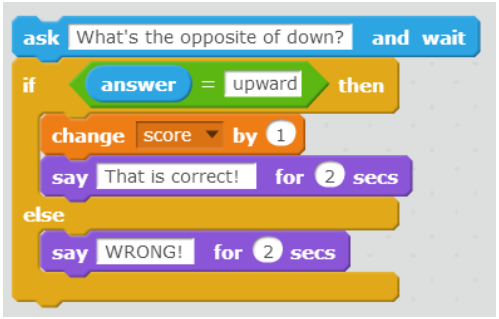
1. Choose a backdrop and a sprite that will be your game show host. The Q&A code you write should belong to this sprite.

2. Create the proper variable to keep score (this is your counter)

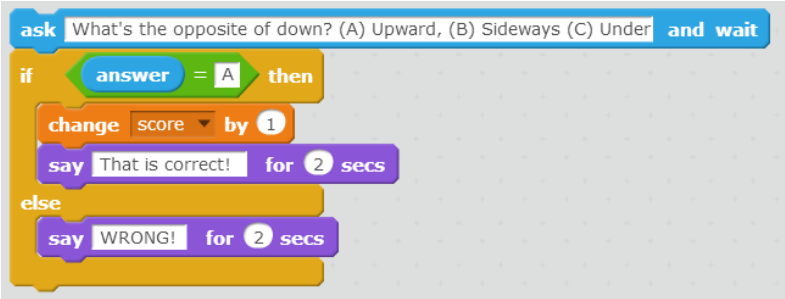
3. Have at least 5 questions prepared.

NOTE: It is *easiest* to have questions whose answers are *numerical*, since Scratch is very sensitive to the words being typed *exactly* as you intended for user input.

For example, say you ask "What is the opposite of down?" and the user answers "up." If you have the answer "upward" stored in the program, the program would tell the user that they are WRONG! This creates frustration for the user, and they won't want to play your game.

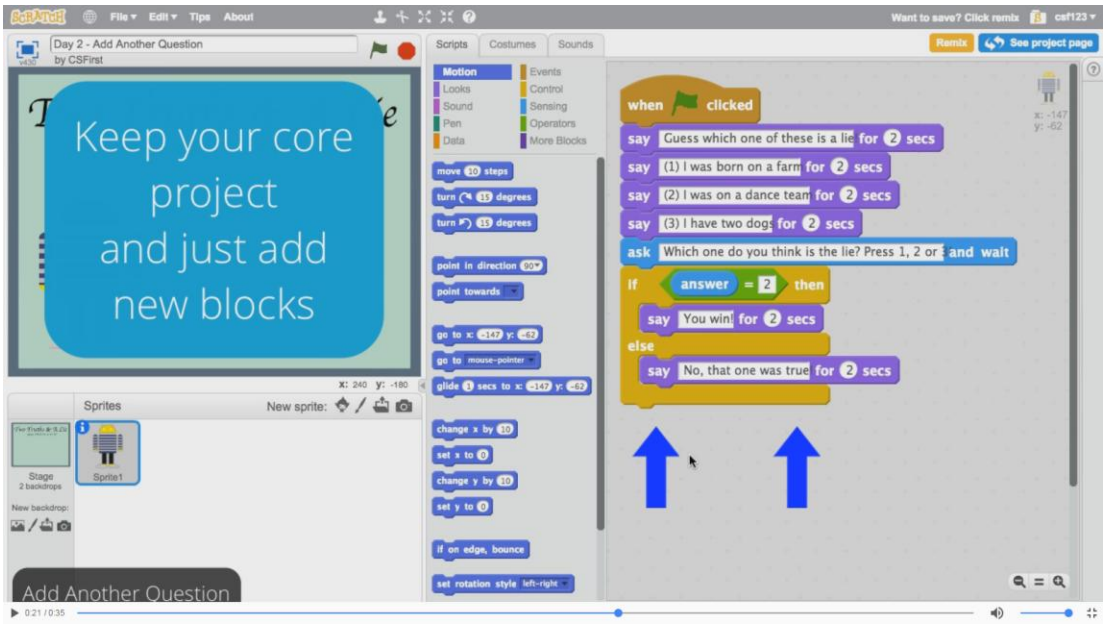


If you want non-numerical questions, a way around the problem described above is to make the questions multiple choice (A, B, or C):



Now it is more difficult for the user to get the wrong answer since they are selecting from a bunch of different options rather than typing it in.

To add more add questions:



4. If the player gets to a total score of 3—they have won the game. There should be a sound and visual effects to signal that the player has won the game.

5. If by the time all five questions have finished, the player has not achieved a score of at least 3, then a "game over, you lose" visual effect and sound must appear.

6. Done? Move onto the second part. At the end of Part II you will submit your work via email.

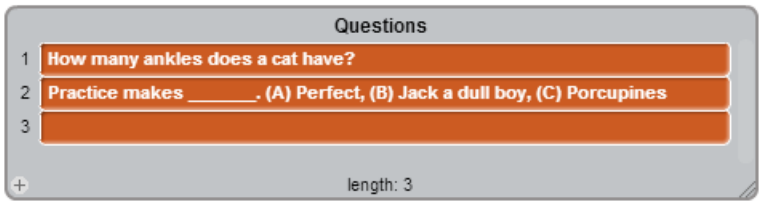
Part II: QUIZ SHOW – CHOOSING RANDOM QUESTIONS

For Part II we will be building on your previous program from part I. Instead of asking questions in a specific order, we are going to make two "**Lists**" (AKA "**Arrays**")—one *questions* list, and one *answers* list—and have the program randomly select a question from the list. Yay! Exciting.

1. Go to the data menu in the scripts palette and click "**Create a List**" (for all sprites). Just like with variables, you can call lists anything you want, but it's easiest to keep track of a list if you call it something relevant. In this context, "**Questions**" would be a logical choice since it will be a list of questions. As long as the "**Questions**" list has a check mark next to it, it will be visible on the stage. While working out the program it is helpful to have it there, but you will want to uncheck it when we are done so that the user cannot see the questions ahead of time.

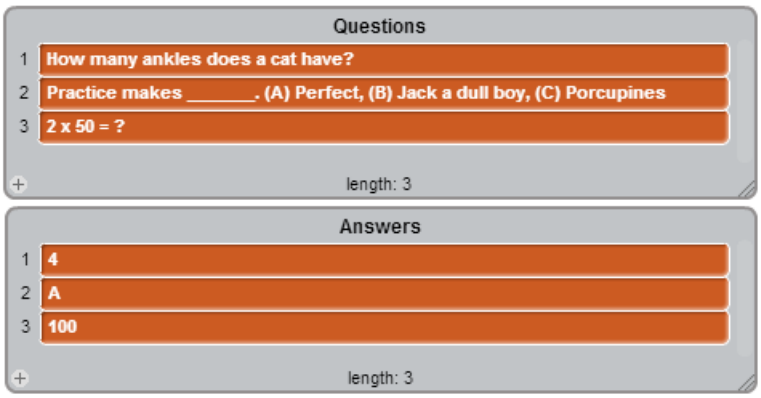
2. If you hit the '+' sign on the list displayed on the stage, you can start typing in questions one at a time. Add at least five questions.

   NOTE: You can click and drag the corner of the list box on the stage so that you can more easily view the questions you've added. **Example:**



3. Repeat this process by making another list called "**Answers**." Write the answers to the questions in the "**Questions**" list <u>in order</u> into the "**Answers**" list.

   Example (corresponding with the questions example above):



4. **<u>READ</u> the following:**

   You will see several new blocks appear under the **Data** menu in the scripts tab once the first list is created. One block we are particularly interested in is shown to the right (item ___ of [Answers]). You can use this block to refer to specific item in a given list. Referencing the examples from step II.3, "Item 1 of Answers" refers to the first item in the Answer list, which in this case would the number 4.

5. **<u>READ</u> the following:**

   At this point, we want the program to choose a **RANDOM** question from your list each time. We then want the program to check and see if the user got the question correct or not by seeing if their answer matches the corresponding answer stored in the "**Answer**" list.

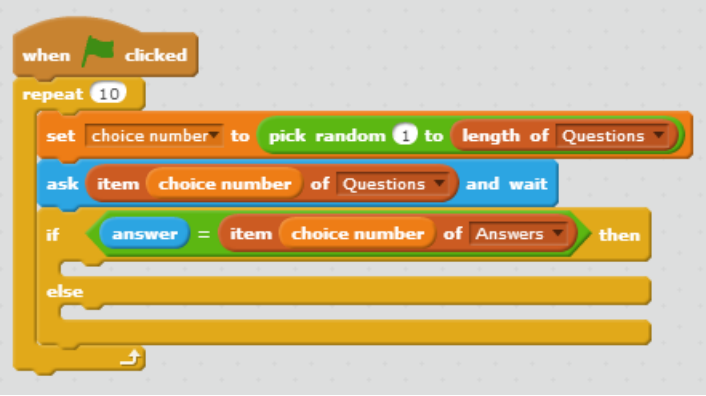
   One of the ways we can identify the question and the answer is by the number of the question/answer within the list. For the Q&A data structure we've built, the **third question** on the "**Question**" list corresponds with the **third answer** on the "**Answer**" list. The **fourth question** on the "**Question**" list corresponds with the **fourth answer** on the "**Answer**" list and so on.

6. **Create** a variable called "**choice number**." If the computer randomly chooses the number **3**, and assigns this value to the variable "**choice number,**" then the game show host should read off the **third *question*** on the list.

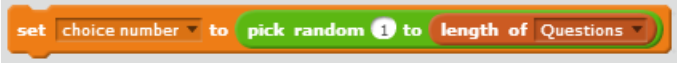
7. **<u>READ</u> the following:**

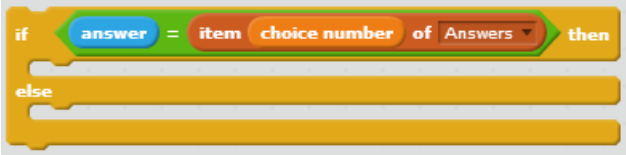
   We want the *answer* to be correct only if the person enters an answer that is equal to the **third *answer*** on the list. In "computer speak" (AKA, "**Pseudo code**") this would be, "***if the "choice number" item from the "Questions" list equals the "choice number" item of the "Answers" list, then the answer is correct."***

8. **<u>IN YOUR NOTEBOOK (1 point)</u>:** copy the following block of code, leaving space after the "if" and "else" to fill in code:



9. **IN YOUR NOTEBOOK (3 points):** Explain what the following three lines of code do – (**Hint:** the reading in Steps **II.4**, **II.5**, and **II.7** contains the information you need to be able to explain the code in detail).





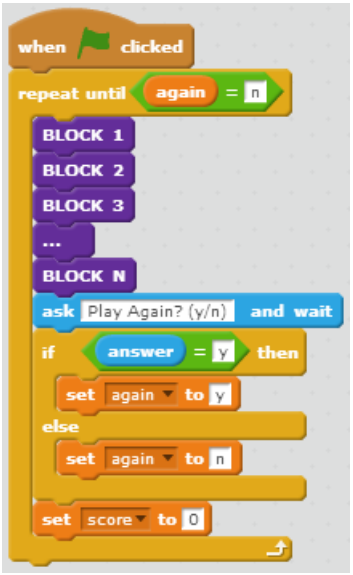
10. **IN YOUR NOTEBOOK (1 pt):** What is the "**repeat [10]**" block for in the example program? Explain what this block of code does.

11. **IN SCRATCH:** Fill in the code that should go after the "if" in Block 3, and after the "else" in Block 3. (**Hint:** remember the goal of the game, and what you did in Part I). Once you get the program working, fill in the missing code **IN YOUR NOTEBOOK (1 pt)**.

12. **IN YOUR NOTEBOOK (2 pts):** What would happen if you added **MORE** questions to your list? Would have to change your code? Why or why not?

13. **IN YOUR NOTEBOOK (2 pts):** Copy the program to the right, and reference the note below to explain what it does.

    NOTE: This is called a basic "Play Again? (y/n)" loop. **Blocks 1, 2, 3, …, N** can represent ***any*** program that you could write for a given character or backdrop. The "**repeat until**" block terminates once the variable "**again**" is set to the value "n". Therefore, at the end of the program, the character or backdrop asks the user what they would like to do, and updates the value of the variable "again" accordingly.



14. **IN SCRATCH**: Add "Play Again (y/n)" functionality to your Quiz Show project.

    NOTE: "Play Again (y/n)" functionality has nothing to do with lists, but nonetheless an important thing to learn since you will be using this type of functionality in many future programs.

QUIZ SHOW SUBMISSION PROTOCOL:

When you have completed your program, click "**SHARE**" in the upper-right your project! Due on Nov. 4TH!