| **Quiz Overview** | |
| --- | --- |
| Describe the purpose and topic of your quiz.  Include a list of questions your quiz will have and what inputs the user will need to enter to answer the questions. | |
| "Addition Quiz"  "What is 11+13: a:22 b:24 c:26 d:25",  "What is 3+12: a:16 b:14 c:15 d:17",  "What is 13+8: a:17 b:21 c:19 d:20",  "What is 24+14: a:32 b:34 c:35 d:38",  "What is 50+50: a:84 b:96 c:80 d:100"  answers "b", "c", "b", "d", "d" | "Subtraction Quiz"  "What is 1-0: a:1 b:0 c:-1 d:2",  "What is 2-7: a:-6 b:5 c:-5 d:-4",  "What is 3-1: a:2 b:4 c:-3 d:5",  "What is 4-5: a:4 b:-1 c:3 d:-2",  "What is 5-3: a:-4 b:4 c:7 d:2"  answers "a", "c", "a", "b", "d" |
| The user has a choice of two quizzes to pick from, the addition quiz and the subtraction quiz.  The user will receive the questions in a random order. | |

| **Program Overview** |
| --- |
| Explain how to run your program.  Describe the variables you have used and what they do for your program. |
| The user should start by running the function: menu(). This will take the user to a sort of menu screen where they have 3 choices  1: Exit  The user will exit the program  2: Addition quiz  The program will run the function: addition\_quiz()  The function addition\_quiz() will define the questions, answers and quiz\_name variables and will be taken to the function: quiz\_body  3: subtraction quiz  The program will run the function: subtraction\_quiz()  The function subtraction\_quiz() will define the questions, answers and quiz\_name variables and will be taken to the function: quiz\_body  After the quiz\_body function has finished the user will be taken back to the menu to either pick another quiz or exit the program  user\_input: collects the users input into a variable  quiz\_name: a string the provides the name of the chosen quiz to be displayed at the start  questions: an array holding the questions and their options (a, b, c, or d)  answers: an array holding the answer to each question  initial\_question\_len: stores the initial length of the questions array upon starting the quiz (also gets rid of a constant)  questions\_correct: stores the number of questions the user has gotten correct  current\_question: stores the number of the question in the array (also used for the answers)  questions\_correct\_p: the percentage of questions the user has gotten correct (divided by initial\_question\_len then multiplied by 100 for percentage) |

| **Testing and Debugging Procedures - Expected** |
| --- |
| Document expected inputs and outputs of the program.  Add more rows as needed. |

| **Version** | **Expected Input** | **Expected Output** | **Actual Output**  **(Screenshot Evidence or copied from console)** |
| --- | --- | --- | --- |
|  | uppercase or lowercase a, b, c, d | incorrect/correct. The .lower() function ensures that any answer the user inputs is converted to a lower case letter for the program to use. |  |
|  | ONLY a, b, c or d | otherwise it should tell the user their input is invalid    this block of code ensures that the user can only input the letters a, b, c, d | After an invalid input the program will automatically pick a new question for the user without removing the question the user did not answer correctly (a valid input). |

| **Testing and Debugging Procedures - Boundary/Edge/Invalid** |
| --- |
| Document the rest. Make sure to note what type it is.. |

| **Version** | **Expected Input** | **Expected Output** | **Actual Output**  **(Screenshot Evidence or copied from console)** |
| --- | --- | --- | --- |
| Session 1 | Boundary case: array attempts to grab a piece of data outside of the arrays range because of this code:  current\_question = random.randint(0 , len(questions))  this happens because the len() function gives five which is in fact the number of items in the array however the array will start counting from zero making it give us an error after it attempts to grab questions[5]  (which does not exist) | this was fixed by changing the aforementioned code to: current\_question = random.randint(0 , len(questions)-1)  this takes one off the number given by the len function which prevents it from giving us a value above the number of items in our array |  |
| Session 2-8 | Invalid case: Quiz body does not check to see if the answers and questions arrays align (have the same number of list items) and this can cause problems if the program tries to grab a piece of data in an array that does not exist (similar to the other issue) | this was fixed by adding an if statement to the beginning of the quiz\_body function to make sure that both the answers and questions arrays had the same amount of data |  |