Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
find_question	1	Search for existing question	Question arrQuestions[2] = { {"What is the capital of France?", "Paris", "London", "New York", "Tokyo", 1}, {"What is the highest mountain in the world?", "Mount Everest", "K2", "Makalu", "Cho Oyu", 1} }; int nQuestions = 2; char strQuestion[] = "What is the capital of France?";	0	0	Pass
find_question	2	Search for non-existing question	Question arrQuestions[2] = { {"What is the capital of France?", "Paris", "London", "New York", "Tokyo", 1}, {"What is the highest mountain in the world?", "Mount Everest", "K2", "Makalu", "Cho Oyu", 1} }; int nQuestions = 2; char strQuestion[] = "What is the capital of Spain?";	-1	-1	Pass
find_question	3	Search with empty question array	Question arrQuestions[0]; int nQuestions = 0; char strQuestion[] = "What is the capital of France?";	-1	-1	Pass

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F

add_record()	1	Adds a new question to the array	Question: "What is the capital of France?" Topic: "Geography " Choice 1: "Paris" Choice 2: "London" Choice 3: "Berlin" Correct choice: "Paris"	"Record added successfully!" is displayed The new question is added to the array	"Record added successfully!" is displayed The new question is added to the array	P
add_record()	2	Attempts to add a duplicate question to the array	Question: "What is the capital of France?" Topic: "Geography " Choice 1: "Paris" Choice 2: "London" Choice 3: "Berlin" Correct choice: "Paris"	"The question already exists:" is displayed, followed by the existing question and its details	"The question already exists:" is displayed, followed by the existing question and its details	P

add_record()	3	Adds a new question to a different topic	Question: "What is the capital of Australia?" Topic: "Geography " Choice 1: "Sydney" Choice 2: "Melbourne" Choice 3: "Canberra" Correct choice: "Canberra"	"Record added successfully!" is displayed The new question is added to the array	"Record added successfully!" is displayed The new question is added to the array	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
edit_record	1	Edit question successfully	"Question 1", ...	"Record updated successfully!"	"Record updated successfully!"	P
edit_record	2	Invalid index for topic selection	"Question 1", ...	"Error: Invalid Input"	"Error: Invalid Input"	P
edit_record	3	Edit correct choice successfully	"Question 1", ...	"Record updated successfully!"	"Record updated successfully!"	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
delete_record	1	Deletes a question with a specific topic and index from an array of questions	arrQuestions with 3 questions: {{"Math", "What is 2 + 2?"}, {"Science", "What is the boiling point of water?"}, {"Math", "What is 3 + 3?"}} nQuestions = 3 User selects topic "Math" and question #2 to delete	arrQuestions with 2 questions: {{"Math", "What is 2 + 2?"}, {"Math", "What is 3 + 3?"}} nQuestions = 2	arrQuestions with 2 questions: {{"Math", "What is 2 + 2?"}, {"Math", "What is 3 + 3?"}} nQuestions = 2	Pass

delete_record	2	Deletes a question with a specific topic and index from an array of questions	arrQuestions with 2 questions: {{"History", "Who was the first president of the United States?"}, {"Science", "What is the atomic number of oxygen?"}} nQuestions = 2 User selects topic "History" and question #1 to delete	arrQuestions with 1 question: {{"Science", "What is the atomic number of oxygen?"}} nQuestions = 1	arrQuestions with 1 question: {{"Science", "What is the atomic number of oxygen?"}} nQuestions = 1	Pass

delete_record	3	Cancels the deletion process and returns to main menu	arrQuestions with 1 question: {{"Geography", "What is the capital of France?"}} nQuestions = 1 User selects topic "Geography" and cancels the deletion process	arrQuestions with 1 question: {{"Geography" , "What is the capital of France?"}} nQuestions = 1	arrQuestions with 1 question: {{"Geography" , "What is the capital of France?"}} nQuestions = 1	Pass

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F

import_data	1	Importing data from a file with valid data	"test.txt" (file exists and contains valid data)	The function should read the contents of the file and store them in the array of Question structs passed as a parameter. The expected output is the success message "Data imported successfully!" printed to the console.	The function successfully reads the contents of the file and stores them in the array of Question structs passed as a parameter. The success message "Data imported successfully!" is printed to the console.	P

import_data	2	Importing data from a file with invalid data	"test.txt" (file exists but contains invalid data, e.g. incorrect format, missing fields, etc.)	The function should display an error message indicating that the file could not be imported and prompt the user to enter the filename again. This should continue until the user enters a valid filename or enters "0" to cancel. Once a valid filename is entered, the function should attempt to import the data again. The expected output is the success message "Data imported successfully!" printed to the console if the data is valid and successfully imported.	The function displays an error message indicating that the file could not be imported and prompts the user to enter the filename again. This continues until the user enters a valid filename or enters "0" to cancel. Once a valid filename is entered and the data is valid, the function successfully imports the data and prints the success message "Data imported successfully!" to the console.	P

import_data	3	Importing data from a file that does not exist	"not_a_file.txt" (file does not exist)	The function should display an error message indicating that the file could not be found and prompt the user to enter the filename again. This should continue until the user enters a valid filename or enters "0" to cancel. Once a valid filename is entered, the function should attempt to import the data again. The expected output is the success message "Data imported successfully!" printed to the console if the data is valid and successfully imported.	The function displays an error message indicating that the file could not be found and prompts the user to enter the filename again. This continues until the user enters a valid filename or enters "0" to cancel. Once a valid filename is entered and the data is valid, the function successfully imports the data and prints the success message "Data imported successfully!" to the console.	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
export_data	1	Valid input	arrQuestions={{"Science", 1, "What is the color of the sky?", {"blue", "green", "yellow"}, "blue"}}, nQuestions=1	File "output.txt" is created and contains the following content: Science 1 What is the color of the sky? blue green yellow blue 	File "output.txt" is created and contains the expected content	P
export_data	2	Invalid filename	arrQuestions={{"History", 1, "Who is the first president of the United States?", {"George Washington", "Thomas Jefferson", "John Adams"}, "George Washington"}}, nQuestions=1	Error message "Error: File cannot be found." is displayed	Error message "Error: File cannot be found." is displayed	P
export_data	3	Empty array	arrQuestions={}, nQuestions=0	Error message "Error: No data to export." is displayed	Error message "Error: No data to export." is displayed	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
load_scores	1	File exists and contains data	scores.txt with "Alice\n100\n\nBob\n200\n\n"	arrScores contains two PlayerScore structs with names "Alice" and "Bob" and scores 100 and 200 respectively, and nScores is set to 2	arrScores contains two PlayerScore structs with expected values and nScores is set to 2	P
load_scores	2	File exists but is empty	empty scores.txt file	arrScores remains unchanged and nScores is set to 0	arrScores remains unchanged and nScores is set to 0	P
load_scores	3	File does not exist	no file named scores.txt in directory	arrScores remains unchanged and nScores is set to 0	arrScores remains unchanged and nScores is set to 0	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F

save_scores	1	Saves a single score	arrScores: [{"John", 10}], nScores: 1	scores.txt: "John\n10\n\n"	scores.txt: "John\n10\n\n"	P
save_scores	2	Overwrites an existing score with a higher one	arrScores: [{"John", 15}, {"Mary", 20}, {"John", 5}], nScores: 3	scores.txt: "John\n15\n\nMary\n20\n\n"	scores.txt: "John\n15\n\nMary\n20\n\n"	P
save_scores	3	Appends new scores to the file	arrScores: [{"John", 10}, {"Mary", 20}], nScores: 2	scores.txt: "John\n10\n\nMary\n20\n\n"	scores.txt: "John\n10\n\nMary\n20\n\n"	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F

add_score	1	Adds a new score to the arrScores array when the name is not found	arrScores: [{strName: "John", nScore: 20}, {strName: "Jane", nScore: 30}], nScores: 2, strName: "Bob", nAdd: 10	arrScores: [{strName: "John", nScore: 20}, {strName: "Jane", nScore: 30}, {strName: "Bob", nScore: 10}], nScores: 3	arrScores: [{strName: "John", nScore: 20}, {strName: "Jane", nScore: 30}, {strName: "Bob", nScore: 10}], nScores: 3	P
add_score	2	Updates the score of an existing name in the arrScores array	arrScores: [{strName: "John", nScore: 20}, {strName: "Jane", nScore: 30}], nScores: 2, strName: "John", nAdd: 10	arrScores: [{strName: "John", nScore: 30}, {strName: "Jane", nScore: 30}], nScores: 2	arrScores: [{strName: "John", nScore: 30}, {strName: "Jane", nScore: 30}], nScores: 2	P
add_score	3	Does not add a new score to the arrScores array when the maximum number of scores has been reached	arrScores: [{strName: "John", nScore: 20}, {strName: "Jane", nScore: 30}, ..., {strName: "Player50", nScore: 50}], nScores: 50, strName: "Bob", nAdd: 10	No change to arrScores or nScores, and the message "Unable to add score - maximum number of scores reached!" is printed	No change to arrScores or nScores, and the message "Unable to add score - maximum number of scores reached!" is printed	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F

display_scores	1	Display top scores	arrScores = [{"John", 50}, {"Alice", 60}, {"Bob", 70}], nScores = 3	Top Scores: ---------- 1. Bob - 70 2. Alice - 60 3. John - 50	Top Scores: ---------- 1. Bob - 70 2. Alice - 60 3. John - 50	P
display_scores	2	Display top scores with new scores loaded	arrScores = [{"John", 50}, {"Alice", 60}], nScores = 2, scores.txt = "Bob\n80\n\n"	Top Scores: ---------- 1. Bob - 80 2. Alice - 60 3. John - 50	Top Scores: ---------- 1. Bob - 80 2. Alice - 60 3. John - 50	P
display_scores	3	Display top scores with empty scores	arrScores = {}, nScores = 0	Top Scores: ----------	Top Scores: ----------	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
play_game	1	Correctly answer a question	arrQuestions, nQuestions, arrScores, nScores	"2\nanswer" printed to console, score = 1	"2\nanswer" printed to console, score = 1	P
play_game	2	Incorrectly answer a question	arrQuestions, nQuestions, arrScores, nScores	"3\nanswer" printed to console, score = 0	"3\nanswer" printed to console, score = 0	P

play_game	3	End game when prompted	arrQuestions, nQuestions, arrScores, nScores	"0\n" printed to console, final score displayed	"0\n" printed to console, final score displayed	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
admin_menu	1	Should add a record to the array of questions	arrQuestions = [{1, "What is the capital of France?", "Paris", 1}, {2, "What is the largest planet in our solar system?", "Jupiter", 1}] nQuestions = 2	nChoice = 1 Enter question ID: 3 Enter question text: What is the boiling point of water? Enter answer: 100 Enter difficulty level (1-3): 2	arrQuestions = [{1, "What is the capital of France?", "Paris", 1}, {2, "What is the largest planet in our solar system?", "Jupiter", 1}, {3, "What is the boiling point of water?", "100", 2}] nQuestions = 3	P

admin_menu	2	Should edit a record in the array of questions	arrQuestions = [{1, "What is the capital of France?", "Paris", 1}, {2, "What is the largest planet in our solar system?", "Jupiter", 1}, {3, "What is the boiling point of water?", "100", 2}] nQuestions = 3	nChoice = 2 Enter question ID to edit: 1 Enter new question text: What is the capital of Italy? Enter new answer: Rome Enter new difficulty level (1-3): 1	arrQuestions = [{1, "What is the capital of Italy?", "Rome", 1}, {2, "What is the largest planet in our solar system?", "Jupiter", 1}, {3, "What is the boiling point of water?", "100", 2}] nQuestions = 3	P
admin_menu	3	Should delete a record from the array of questions	arrQuestions = [{1, "What is the capital of Italy?", "Rome", 1}, {2, "What is the largest planet in our solar system?", "Jupiter", 1}, {3, "What is the boiling point of water?", "100", 2}] nQuestions = 3	nChoice = 3 Enter question ID to delete: 2	arrQuestions = [{1, "What is the capital of Italy?", "Rome", 1}, {3, "What is the boiling point of water?", "100", 2}] nQuestions = 2	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F

game_menu	1	Play game (case 1)	arrQuestions: {...}, nQuestions: 10, arrScores: {...}, nScores: 5	Starts game and updates scores	Starts game and updates scores	P
game_menu	2	View scores (case 2)	arrQuestions: {...}, nQuestions: 10, arrScores: {{"Player 1", 5}, {"Player 2", 10}}, nScores: 2	Displays scores: Player 2: 10, Player 1: 5	Displays scores: Player 2: 10, Player 1: 5	P
game_menu	3	Exit game (case 0)	arrQuestions: {...}, nQuestions: 10, arrScores: {...}, nScores: 5	Saves scores and exits game	Saves scores and exits game	P

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
main_menu	1	User inputs 1 for admin menu, enters correct password, and adds a new question to arrQuestions	arrQuestions initially contains 5 questions, user inputs "1", enters correct password "adminpassword", inputs data for new question (Question: "What is the capital of France?", Answer: "Paris"), and inputs "0" to sign out	arrQuestions should now contain 6 questions, with the new question added to the end of the array	arrQuestions contains 6 questions, with the new question added to the end of the array	P

main_menu	2	User inputs 2 for game menu, plays game and achieves a score of 5, then exits game menu	arrScores initially contains 3 scores, user inputs "2", plays game and achieves a score of 5, then inputs "0" to exit game menu	arrScores should now contain 4 scores, with the new score added to the end of the array	arrScores contains 4 scores, with the new score added to the end of the array	P
main_menu	3	User inputs 0 to exit program	User inputs "0" to exit program	Program should exit	Program exits	P