| Functions orthletter | # Description 1 Key is found in the middle of the row | Sample Input Data row = "ABCDE", size = 5, key = "C" | Expected Output | Actual Output | P/F |
|-------------------------|--|--|--|--|-----|
| ar O I Catalo | 2 Key is not found in the row | row = "XYZPO", size = 5, key = 'A' | 4 | 4 | P |
| | 3 Key is found at the beginning of the row | row = "MNOPQ", size = 5, key = "N" | 0 | 0 | P |
| erateUniqueRandomLetter | 1 All letters except one are used | usedLetters = "ABCDEFGHUKLMNOPQRSTUVWXYZ" | Z Z | Z | P |
| | 2 No letters are used 3 Some letters are used, but the random letter is not one of them | usedLetters = () usedLetters = "FGHU" | any letter from A to Z any letter from A to E, or from K to Z | any letter from A to Z any letter from A to Z | P |
| ateBoard | 1 Create a 3x3 board | row = 3, col = 3 | 3x3 board (with unique letters per row) | 3x3 board (with unique letters per row) | P |
| | 2 Create a large board with maximum dimensions | row = 15, col = 15 | 15x15 board (with unique letters per row) | 15x15 board (with unique letters per row) | P |
| | 3 Create a board with non-square dimensions | row = 4, col = 6 | 4x6 board (with unique letters per row) | 4x6 board (with unique letters per row) | P |
| | | wordburklande has apple, baseva multifore = 2 a multifore = 3 a multifore = 0 a multifo | Display Velue: "I is a Pull." | Display trick: "It is a first." | |
| estionAnswerPhase | 1 Correctly answer the trivia | | Display finish. "It is a fluit." Function returns 1, indicating content enemes, and etter in board is now." Under the content and the conten | Display Prink "Til a Plut" Function returns ", find carrier, and entire " function returns ", find carrier, downed and entire in board is now " the carrier ", the carri | Р |
| | 2 Wrong answer to the trivia | Same as Sample Input Data 1, but the user enters "banana" instead of "apple" | | | P |
| | 3 Choose to exit the game | Same as Sample Input Data 1, but the user enters "0" to exit the game board * {\(\begin{array}{c} \cdot \cdo | Function returns -1, indicating the user wants to exit the game without answering the trivia | Function returns -1, indicating the user wants to exit the game without answering the trivia | P |
| | Player wins with at least one correct answer in each row | oil = 4 board = {(", *, *, *, *), *, *, *, *, *, **)} | 1 (Player wins) | 1 (Player wins) | P |
| | 2 Player does not win because one row has no correct answers | col = 4 Double ([(1,11,11,11,11,11,11,11,11,11,11,11,11,1 | 0 (Player does not win) | 0 (Player does not win) | Р |
| | 3 Player does not win because not all rows have been answered | | 0 (Player does not win) | 0 (Player does not win) | Р |
| | 1 Row has at least one *** | currentRow = (2, 2, 2, 2, 2, 2) | 1 | 1 | Р |
| | 2 Row does not have any *** | cal = 5 currentStow = (**) cal = 1 cal = 10 cal | 0 | 0 | P |
| | 3 Row has only one letter and it is *** | col = 1 | 1 | 1 | P |
| mePhase | 1 Player wins the game 2 Player loses the game | Assuming the data contactly attained and concease not by pay again. | "You WIN" then display "Do you want to play again?" "You LOSE" then display "Do you want to play again?" | "You WIN" then display "Do you want to play again?" "You LOSE" then display "Do you want to play again?" | P |
| | 2 Player loses the game 3 Play exits the game | Assuming the user chooses to exit the game during gameplay. | "You LOSE" then display "Do you want to play again?" "You LOSE" then display "Do you want to play again?" | "You LOSE" then display "Do you want to play again?" "You LOSE" then display "Do you want to play again?" | P |
| | | Assuming the user incorrectly arrivers all questions and chooses not to piny again. Assuming the user chooses to set the gene during gamegal, wordstbabbase = {"laptier", "I'll is a fixet."}}, "Charry", "I'll is red."}} unstricted: 3 "I'll in a fixet." "This yellow."}, "Charry", "I'll is red."}} | | | |
| | 1 Words in the database are already sorted alphabetically | nummords = 3 wordsDatabase = (Chanana*, Cit is vellow?)), Cacole*, Cit is a fruit.*)), Coherry*, Cit is red.*)) | wordsDatabase = {("apple", "It is a fruit.")}, ("banana", ("It is yellow.")}, ("cherry", ("It is red.")}} | wordsDatabase = {{"apple", "It is a fruit."}}, "banana", "It is yellow."}, {"cherny", "It is red."}}} | Р |
| | 2 Words in the database are not sorted alphabetically | numWords = 3 | wordsDatabase = {("apple", "It is a fruit.")}, ("banana", {"It is yellow.")}, ("cherry", {"It is red.")}} | wordsDatabase = {{"apple", ("It is a fruit.")}, {"banana", {"It is yellow."}}, {"cherry", {"It is red."}}} | Р |
| | 3 Only one word in the database | woordstatease = (Teppin', Tis a flut,)), (Teanard', Tis yellow), (Teany', (Tis red.))) woordstatease = (Teanard', Tis yellow), (Teppin', Tis a flut,)), (Teany', (Tis red.))) woordstatease = (Teanard', Tis yellow), (Teppin', Tis a flut,)), (Teany', (Tis red.))) woordstatease = (Teopin', Tis a flut,))) woordstatease = (Teopin', Tis a flut,))) | wordsDatabase = {{"apple", {"it is a fruit."}}} | wordsDatabase = (("apple", ("It is a fruit."))) | Р |
| | | wordsDatabase = {{"apple", {"It is a fruit."}}, {"banana", {"It is yellow."}}, {"cherry", {"It is red."}}} numiforce = 3 %y = "banana" | | , | |
| | 1 Word exists in the database | key = "banana" wordsDelabase = ("apple", "It is a fruit."]), "banana", "It is yellow."]), "cherny", "It is red."]]) numifords = 3 key = "grape" key = "grape" | 1 (banana) | 1 (banana) | Р |
| | 2 Word does not exist in the database | numificats = 3 Key = "Grape" words@latabase = () numificats = 0 | -1 | 4 | Р |
| | 3 Database is empty | key = "approx" "approx" "His a facility" "Papagon" "His uniform" "His cod "III. | 4 | 4 | Р |
| nerwriteWord | 1 Overwrite with a unique word | numittide = 2 | wordsDatabase = {("apple", ("it is a fruit.")), ("grape", ("it is yellow.")), ("chemy", ("it is red."))} | wordsDatabase = {{"apple", {"It is a fault."}}, {"grape", {"It is yellow."}}, {"cherry", {"It is red."}}} | Р |
| | 2 Overwrite with a non-unique word | origWord = "banana" | "Word already exists in the database. Exiting" | "Word already exists in the database. Exiting" | Р |
| | 3 Overwrite when database is empty | nesWord = Pagistir socializations = 1 configura = 1 configura = 1 configura = Pagistir configura = Pagist | wordsDatabase = () | wordsDatabase = {} | |
| | | words/Database = {{"apple", {"H is a fruit."}}, {"banaina", {"H is yellow."}}, {"cherry", {"H is red."}}} words/Database = {{"apple", {"H is a fruit."}}, {"banaina", {"H is yellow."}}, {"cherry", {"H is red."}}} words/Database = {{"apple", {"H is sweet" and "it grows on trees!"}} | | | Ė |
| | 1 Add clues to a word with space for more clues | wordsDatabase a (Cannie", Cit is a finit "I), Chanana", Cit is vellow "I), Criberry", Cit is red "II). | Two clues ("It is sweet" and "It grows on trees") are added to the word "apple". | Two clues ("it is sweet" and "it grows on trees") are added to the word "apple". | Р |
| | 2 Add clues to a word when maximum limit is reached | woodnode + 0 Adding 11 class ("Tay is a first."), ("bannan", ("It is yellow")), ("dwny", ("It is red "))) Adding one clue. "It is a beny" Adding one clue. "It is a beny" Choosing not to add more clue. | "Clues maximum limit (10) reached." | "Clues maximum limit (10) reached." | Р |
| | 3 Add clues to a word without adding more clues | Adding one clue: "It is a beny" Choosing not to add more clues words/blabbase ("paple", "(fill is if vist", "It is sweet"), 2(), "banana", ("It is yellow", "It is long."), 2(), ("cherry", ("It is red.", "It is small."), 2()) | One clue ("It is a berry") is added to the word "banana". | One clue ("It is a berry") is added to the word "banana". | Р |
| | | wordsDatabase = {{"apple", {"It is a fruit", "It is sweet."], 2}}, ("banana", {{"It is yellow.", "It is long."], 2}}, {"cherry", {{"It is red.", "It is small."], 2}}} Viewing disus for the word "apple" | | | |
| wClues | 1 View clues of an existing word | | Clues for the word "apple" are displayed: "It is a fruit." and "It is sweet." | Clues for the word "apple" are displayed: "It is a fruit." and "It is sweet." | P |
| | 2 View clues but database is empty | wordsDatabase = {} numWords = 0 | "No words in the database. Please add words first." | "No words in the database. Please add words first." | P |
| | | numWords = 0 wordsDatabase = {\papele", {\papele", \text{ is a fruit.", "It is sweet."}, 2\}, \papele \text{"banana", {\papele \text{ is long."}, 2\}, \papele \text{"this perior.", "It is long."}, 2\}, \papele \text{"this red.", "It is small.", 2\}} numWords = 3 | | | |
| | 3 Go back to the menu | Choosing to go back to the menu (enter "0") wordsDatabase = {("apple", {("It is a fruit.", "It is sweet."), 2}}, {"banana", {("It is yellow.", "It is long."), 2}}, {"cherry", {("It is red.", "It is small."), 2}}} | Function returns without displaying any clues. | Function returns without displaying any clues. | P |
| | View words and clues, navigate through entries | numbrodes a 3 | Words and their corresponding clues are displayed in alphabetical order. The user can navigate through entries using 'N' and 'P', and exit using 'X'. | Words and their corresponding clues are displayed in alphabetical order. The user can navigate through entries using 'N' and 'P', and exit using 'X'. | Р |
| | 2 View words and clues when database is empty | monthly control of the second | "No words in the database. Please add words first." | "No words in the database. Please add words first." | Р |
| | 3 Wew words and clues, navigate through entries, then reach the end of the list | numWords = 3 User navigates through entries until reaching the last entry. words/batabase = {*rapple*, {*fit is a fault*, *fit is sweet*, 25}, {*panana*, {*fit is yellow*, *fit is long.*7, 25}} | "End of the list." and still display the same current word entry | "End of the list." and still display the same current word entry | Р |
| dWord | 1 Add a new word to the database | numWords = 2 | Word "grape" is successfully added to the database with its corresponding clues. | Word "grape" is successfully added to the database with its corresponding clues. | Р |
| | 2 Attempt to add an existing word | would get a new word garper wind class: words/Database = ("appler", ("It is a fault," "It is sweet."), 2]), ("banana", {{"It is yellow.", "It is long."), 2]}} numlfords = 2 Attempting to add an existing word "banana". | "Word already exists in the database, Exiting" | "Word already exists in the database. Exiting" | Р |
| | 3 Attempt to add a word when the database is full | wordsDatabase = {<150th word entry here>} numiVords = 150 Attendation to add an oxidation word "bacaca" | "Database is full. Cannot add more words." | "Database is full. Cannot add more words." | |
| | | Assumpting to add an externing word batterial . words/Database = ("apple", ("It is a brit.", "It is sweet."), 2)), ("banana", {("It is yellow.", "It is long."), 2)}) numifords = 2 Modifying the word "apple" to "avit". | | | r |
| dityEntry | 1 Modify the word of an entry in the database | Modifying the word "appek" of "appek", ("It is a fruit," "it is sweet"), 2]), ("banana", {("It is yellow.", "It is long."), 2])) numifrods = 2 Modifying the second due of the word "banana". | Word "apple" is successfully modified to "avt" in the database. | Wiord "apple" is successfully modified to "avit" in the database. | P |
| | 2 Modify a clue of an entry in the database | Modifying the second dise of the word "banana". wordsDatabase = {("apple", {("It is a fruit.", "It is sweet."), 2}}, ("banana", {("It is yellow.", "It is long."), 2}}) | Clue "it is long," of the word "banana" is successfully modified to a new clue. | Clue "It is long," of the word "banana" is successfully modified to a new clue. | P |
| | 3 Attempt to modify a non-existent word | Modifying the second due of the word 'bainsham'. wordshablased = ("[replect, ("I'll a sell," 1 is a seed; 1, 2)], ("banana", ("[I'll s yellon", "It is king "), 2]) Adamping to modify the word "cange" winth does not easis in the database. Adamping to modify the word "cange" winth does not easis in the database. Out of the database of the database of the database of the database. Adamping to modify the word "cange" winth does not easis in the database. | "Word does not exist in the database." | "Word does not exist in the database." | Р |
| eteWord | 1 Delete an existing word from the database | numWords = 2 Deleting the word "apple". | Word "apple" is successfully deleted from the database. | Word "apple" is successfully deleted from the database. | Р |
| | Attempt to delete a non-existent word from the database | wordsDatabase = {("apple", {("It is a fruit.", "It is sweet."), 2}), ("banana", {("It is yellow.", "It is long."), 2}}} numlfolds = 2 Attempting to delete the word "orange" which does not exist in the database. | "World does not exist in the database." | "Word does not exist in the database." | Р |
| | 3 Delete the last word from the database | Additional to the second of th | Word "apple" is successfully deleted from the database, and the database becomes empty. | Word "apple" is successfully deleted from the database, and the database becomes empty. | Р |
| | | words/Database = ("Apple", ("It is a fruit ", "It is sweet."). 2))) numifords = 1 Deleting the clus "It is sweet." from the word "apple". | | | Ė |
| eteclue | Delete an existing clue from a word in the database | wordsDatabase = {("apple", {("It is a fruit.", "It is sweet."), 2})) numWords = 1 | Clue "it is sweet." is successfully deleted from the word "apple" in the database. | Clue "it is sweet." is successfully deleted from the word "apple" in the database. | P |
| | 2 Attempt to delete a clue from a non-existent word in the database | Attempting to delete a cute from the word 'congrey "which does not exist in the database. wordsDatabase ("apple", ("It is a "hut", "It is sweet"), 2()) numification a 1 Attempting to delete a non-assistent clue from the word "apple". | "Word does not exist in the database." | "Word does not exist in the database." | Р |
| | Attempt to delete a clue that does not exist in the word in the database | Attempting to delete a non-existent clue from the word "apple", wordSchildsbase = 0 in multifolds = 0 in following the control of the control | "Clue does not exist in the distalhase." | "Che does not exist in the distribuse." | P |
| | | It is sender ved Chject hannama It is a full Ved It is yellow Yes | | | |
| ort | 1 Import a file with valid data | wordsDatabase = {} ournWoods = 0 | Words "apple" and "banana" with their corresponding cluss are successfully imported into the wordsDatabase. | Words "apple" and "banana" with their corresponding clues are successfully imported into the wordsDatabase. | Р |
| | 2 Attempt to import a non-existent file | Text file name: non_existent_file.txt wordShatsbase = {\(\begin{array}{c} \arrappi \) \text{("It is a fruit.", "It is sweet."}, 2)}\) \[\text{numMirror is a fruit."} \] | File does not exist.* | File does not exist." | P |
| | 3 Import a file with duplicate words | Test the name con_existent_file.td swortDuttablese_file_file_file_file_file_file_file_fil | Prompt "apple already exists in the database. Overwrite Entry? (jiri): "is displayed, and upon entering "y", "the word "apple" is successfully overwritten with the new data. | Brown Tools desired sold to the delibers Councils State (1) to desired and use added 5° the word Tools's a council to the delibers (1) to the state of the state | |
| | □ Impans arend Will Culphouse Worlds | It is sour: Yes wordsDatabase = ([*apple", ([*H is a fruit.", 'R is sweet."), 2]), (*Banana", ([*H is a fruit.", 'R is yellow "), 2])) | The content of the file "export, test full should be: Object apple It is a fluit. It is sweet. | The content of the file "export, set but should be: Open: apple It is a thut: It is sweet. | , |
| port | 1 Export data to a valid file | numWords = 2 | Object: banana It is a fruit: It is yellow. | Object: banana It is a fruit: It is yellow. | Р |
| | Attempt to export data when no words are in the database | Fax mant: upper_lead.ox words/Database = {} numifolds = 0 final country final | "No words in the database. Please add words first." | "No words in the database. Please add words first." | P |
| | 2 Assempt to export data when no words are in the database | | | | |

words/ballables contains 151 entries
full/filtrate 151
3 Attempt to export more than 150 data entries
Firence Cannot export data entries greater than 150."
Firence Cannot export data entries greater than 150."
Prince Cannot export data entries greater than 150."