**MoralMP.c Test Cases (Scrambled Words)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| inputPlayerDetails() | 1 | User inputs small letters for name | player.name=ally | Accepted name and proceed to enter date | Accepted name and proceed to enter date | P |
| 2 | User inputs capital letters for name | player.name=ALLY | Accepted name and proceed to enter date | Accepted name and proceed to enter date | P |
| 3 | User inputs a combination of both for name | player.name=Ally | Accepted name and proceed to enter date | Accepted name and proceed to enter date | P |
|  | 4 | User inputs a correct month, day, date for date | dateToday.month= 1  dateToday.day =24  dateToday.day= 2023 | Accepted and proceed to game | Accepted and proceed to game | P |
|  | 5 | User inputs a wrong month for date | dateToday.month= 13  dateToday.day =24  dateToday.day= 2023 | "Invalid date. Please enter a valid date."  Ask to enter date again | "Invalid date. Please enter a valid date."  Ask to enter date again | P |
|  | 6 | User inputs a wrong day for date | dateToday.month= 13  dateToday.day =60  dateToday.day= 2023 | "Invalid date. Please enter a valid date."  Ask to enter date again | "Invalid date. Please enter a valid date."  Ask to enter date again | P |
|  | 7 | User inputs a wrong year for date | dateToday.month= 13  dateToday.day =60  dateToday.day= 2026 | "Invalid date. Please enter a valid date."  Ask to enter date again | "Invalid date. Please enter a valid date."  Ask to enter date again | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| Shuffle\_Words() | 1 | This does not have an input but it shuffles the words to be given | i=0 | Random word from text file dictionary | Random word from text file dictionary | P |
| 2 | i=4 | Random word from text file dictionary | Random word from text file dictionary | P |
| 3 | i=51 | Loop stops | Loop stops | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| Load\_Words () | 1 | This does not have an input but it loads all the contents of the text file to the code | i=0 | ATTRACT | ATTRACT | P |
| 2 | i=50 | Loop stops | Loop stops | P |
| 3 | Text file does not exist | “Error opening dictionary file” | “Error opening dictionary file” | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| countWordLength () | 1 | This does not have an input but it computes the length of a word | word[]=ATTRACT | 7 | 7 | P |
| 2 | word[]=STRAWBERRY | 10 | 10 | P |
| 3 | word[]=BACHELOR | 8 | 8 | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| scrambleWord() | 1 | This does not have an input but it scrambles the letters from a word | word[]=STRAWBERRY | word[]=SRWBARTREY | word[]=SRWBARTREY | P |
| 2 | word[]=QUARRELING | word[]=QANEIURRLG | word[]=QANEIURRLG | P |
| 3 | word[]=DECLINE | word[]=DELNCIE | word[]=DELNCIE | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| guessAnswer() | 1 | This does not have an input but it checks if the answer of the player is correct or not | Correct guess  usedClues=0 | “CORRECT!”  Result=1  Points+200 | “CORRECT!”  Result=1  Points+200 | P |
|  | 2 | Wrong guess | “INCORRECT.TRY AGAIN”  Result=0  Points-300 | “INCORRECT.TRY AGAIN”  Result=0  Points-300 | P |
|  | 3 | Correct guess  usedClues=1 | “CORRECT!”  Result=1  Points+word length -10 | “CORRECT!”  Result=1  Points+word length -10 | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| processPlayerTurn() | 1 | User inputs a from the choices | nAction=1 | “First letter of the word is \_\_” | “First letter of the word is \_\_” | P |
| 2 | User inputs a negative number | nAction=-1 | “Invalid input. Please try again.” | “Invalid input. Please try again.” | P |
| 3 | User inputs number not on the choices | nAction=5 | “Invalid input. Please try again.” | “Invalid input. Please try again.” | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| askToPlayAgain() | 1 | User inputs from the choices | playAagin=1 | Game restarts | Game restarts | P |
| 2 | User inputs negative number | playAagin=-1 | “Only enter 1 or 2!” | “Only enter 1 or 2!” | P |
| 3 | User inputs number not on the choices | playAagin=3 | “Only enter 1 or 2!” | “Only enter 1 or 2!” | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| loadHighScores() | 1 | This does not have an input but it loads the top 5 high scores from the text file | highScores[i].name= ALLY  highScores[i].score= 5000  highScores[i].dateAchieved.month=1  highScores[i].dateAchieved.day=24  highScores[i].dateAchieved.year=2023 | Content will be loaded into the code | Content will be loaded into the code | P |
| 2 | highScores[i].name= ALLY  highScores[i].score= 0  highScores[i].dateAchieved.month=1  highScores[i].dateAchieved.day=24  highScores[i].dateAchieved.year=2023 | Will not reach the code | Will not reach the code | P |
| 3 | highScores[i].name= ALLY  highScores[i].score= -100  highScores[i].dateAchieved.month=1  highScores[i].dateAchieved.day=24  highScores[i].dateAchieved.year=2023 | Will not reach the code | Will not reach the code | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| updateAndSort() | 1 | This does not have an input but it updates the current high scores | player.score=2000  highScores[1-5].score=0 | Will be the first place | Will be the first place | P |
| 2 | player.score=5000  highScores[1].score=2000 | Will become the first place | Will become the first place | P |
| 3 | player.score=0  highScores[1-5].score=0 | Will not reach the code |  | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function Name | # | Test Description | Sample Input | Expected Result | Actual Result | P/F |
| saveHighScores() | 1 | This does not have an input but it saves the top 5 high scores in a text file | highScores[i].name= ALLY  highScores[i].score= 5000  highScores[i].dateAchieved.month=1  highScores[i].dateAchieved.day=24  highScores[i].dateAchieved.year=2023 | Content will be saved into the text file | Content will be saved into the text file | P |
| 2 | highScores[i].name= ALLY  highScores[i].score= 0  highScores[i].dateAchieved.month=1  highScores[i].dateAchieved.day=24  highScores[i].dateAchieved.year=2023 | Will not reach the code | Will not reach the code | P |
| 3 | highScores[i].name= ALLY  highScores[i].score= -100  highScores[i].dateAchieved.month=1  highScores[i].dateAchieved.day=24  highScores[i].dateAchieved.year=2023 | Will not reach the code | Will not reach the code | P |