Function: instructions()

1. Displays the rules of the game
2. Displays the game controls and commands
3. Wait for the user to press any key to continue
4. Returns the control to main menu.

Function: addcandidate()

1. Open the database file data.txt and addcandidate.txt in append model.
2. If any error reported in opening file, report error message.
3. Call function validate1() with structure can as parameter
4. Check the return value, if the return value is 0, return to main menu. If return value is 1, then update the files data.txt and addcandidate.txt with the user entered data.
5. Close file pointers.

Function: validate1()

1. Open the database file data.txt in read mode.
2. Take input of candidate name for validation.
3. Take input of the party name for validation.
4. Take input of the division name between range of 1-4.
5. Find the number of records already in the database. Increment by 1 and assign to the new candidate.
6. Check for the division number validity, if is greater than 0 and less than 5, accept the entry and move the file pointer to the beginning of the file.
7. Check the database for the existence of the candidate already in the database, if candidate already exist, display error, and return 0 the addcandidate() function. If there is entry found in the database, return the return code 1 to addcandidate() function.
8. Close the file pointers.

Function: viewcandidate()

1. Open the database file data.txt in read mode.
2. Display error if any error occurs in opening the file.
3. Display all the records stored in the file in a loop.
4. Close the file pointers and return the control to main menu.

Function: search()

1. Open the database file data.txt in read mode.
2. Take the search option input. Search options are search by Candidate Name or Division Name or by party name.
3. Based on the search option input take input of candidate name for search by candidate name, take input of division name for search by division name and take input of party name for search by party name.
4. For Search by Candidate name, within a loop compare the candidate’s name fetched from database and input name. if matches display the record.
5. For search by division number, within a loop compare the candidate’s division fetched from database and input division number. If matches display the record.
6. For search by party name, within a loop compare the candidate’s party assignment fetched from database and input party name. If matches display the record.
7. If the no records found for the provided input, display the error message.
8. Close the file pointers.

Function: check()

1. Open the database file data.txt in read mode
2. Calculate the number of candidates registered in the database
3. Display the number of registered candidate details
4. Close the file pointer.

Function: vote()

1. Open the database file data2.txt and votecandidate.txt in append mode.
2. Display error if any error occurs during opening the file
3. Call the function validate2() to validate the eligibility of the voter
4. If the return code is 1, all the voter to proceed with voter registration process
5. Take input of the voter’s name, division number
6. Record the vote in the database file data2.txt and votecandidate.txt
7. Close the record pointers

Function: validate2()

1. Open the database file data2.txt and votedvoters.txt in read mode
2. Display error if any error occurs during opening the file.
3. Take input of the voter id and voter age to validate
4. Check the eligibility of age. Voter should be above 18 years of age to get eligibility for voting. Also check for the registration of the voter, a voter cannot register multiple times
5. Check voting database to make sure the voter not voted already. If already voted, decline the registration.
6. If the candidate is eligible return to function vote with return code 1.
7. Close the file pointers.

Function: viewvoter()

1. Open the database file data2.txt in read mode
2. Display error if any error occurs during opening the file
3. Display all records from the data2.txt
4. Close the file pointer.

Function: votecasting()

1. Open the database file data.txt in read mode and data2.txt .txt in append mode
2. Display error if any error occurs during opening the file.
3. Take input of the voter’s name and division.
4. Check the database for registration of the voter. If registered proceed with voting process.
5. Take input of the candidate id for voting. Check the validity of the candidate, if candidate not found, display the error message, and take input again from the user. Iterate this loop until correct candidate id is entered.
6. If candidate id is correct, update the database accordingly.
7. Call the function view1reg() to record the voter casted his vote.
8. Call the function deletevoter() to delete the voter just completed from the registration table.
9. Display the error message if the voter tries to vote second time.
10. Call the function check2() to validate all the voter’s casted vote. If return code is 0, open the database data.txt in read mode.
11. Record the highest vote by each division (1-4)
12. Based the highest vote registered display the results.
13. If still some registered voters did not complete their voting, display message as election process still in open and wait for the results.
14. Close the file pointers.

Function: view1reg()

1. Open the database file data.txt in read mode and votedvoters.txt in append mode
2. Record the details of the voter who casted the vote in the database.
3. Close the file pointer and return to the function votecasting()

Function: deletevoter()

1. Open the database data2.txt in read mode and test1.txt in write mode
2. Record the voters not yet casted their vote in the file and skip the voters already casted their vote
3. After completing this process test1.txt will have only voters not yet casted the vote. Copy the content to database file data2.txt
4. Close the file pointers.

Function: check2()

1. Open the database file data2.txt in read mode
2. Check for the number of records in the database
3. If there is no records return to calling function votecasting with return code 0. There are pending records return with code 1.
4. Close the file pointers.

Function: sortbydivision()

1. Open the database file data.txt in read mode
2. Sort the database within the structure based on division number
3. Display the sorted records
4. Close the file pointers.

Function: viewcanvote()

1. Open the database file data.txt in read mode
2. Take the search option input. Search options are search by Candidate Name or Division Name or by party name.
3. Record the total number of votes by division number. If the input is based division number, results are displayed based on the input division number.
4. If the input is based in the candidate, take input of the candidate’s name, and display the records pertaining to the candidate.
5. If the input is base based on party name, take input of the party name and display the records pertaining to the party.
6. If any of the input does not match the records, display the error message.
7. Close the file pointers.

Function: viewcanvote2()

1. Open the database file data.txt in read mode, results file results.txt in append mode.
2. Results.txt file is updated with all result details to be displayed.
3. Print the winners by division wise, and also display candidates with lowest number of votes.
4. Close the file pointers.

Function: view2reg()

1. Open the voted voter database votedvoters.txt in read mode.
2. Display all the records from the database. This list the voters already casted their vote.
3. Close the file pointer

Function: viewbydiv()

1. Open the database file data.txt in read mode.
2. Take input of the division number whose details to be viewed
3. Display the details of the division as per the input provided. Details like winner by division wise is displayed.
4. Close the file pointer