**PROJECT NAME:**

Electoral Commission Project- E-Systems

**BY:**

S. KAVIYARASU



DEPARTMENT OF AI AND DS

COIMBATORE INSTITUTE OF TECHNOLOGY

COIMBATORE

JANUARY 2022

**ELECTORAL COMMISSION PROJECT- E-SYSTEMS**

**AN INTRODUCTION:**

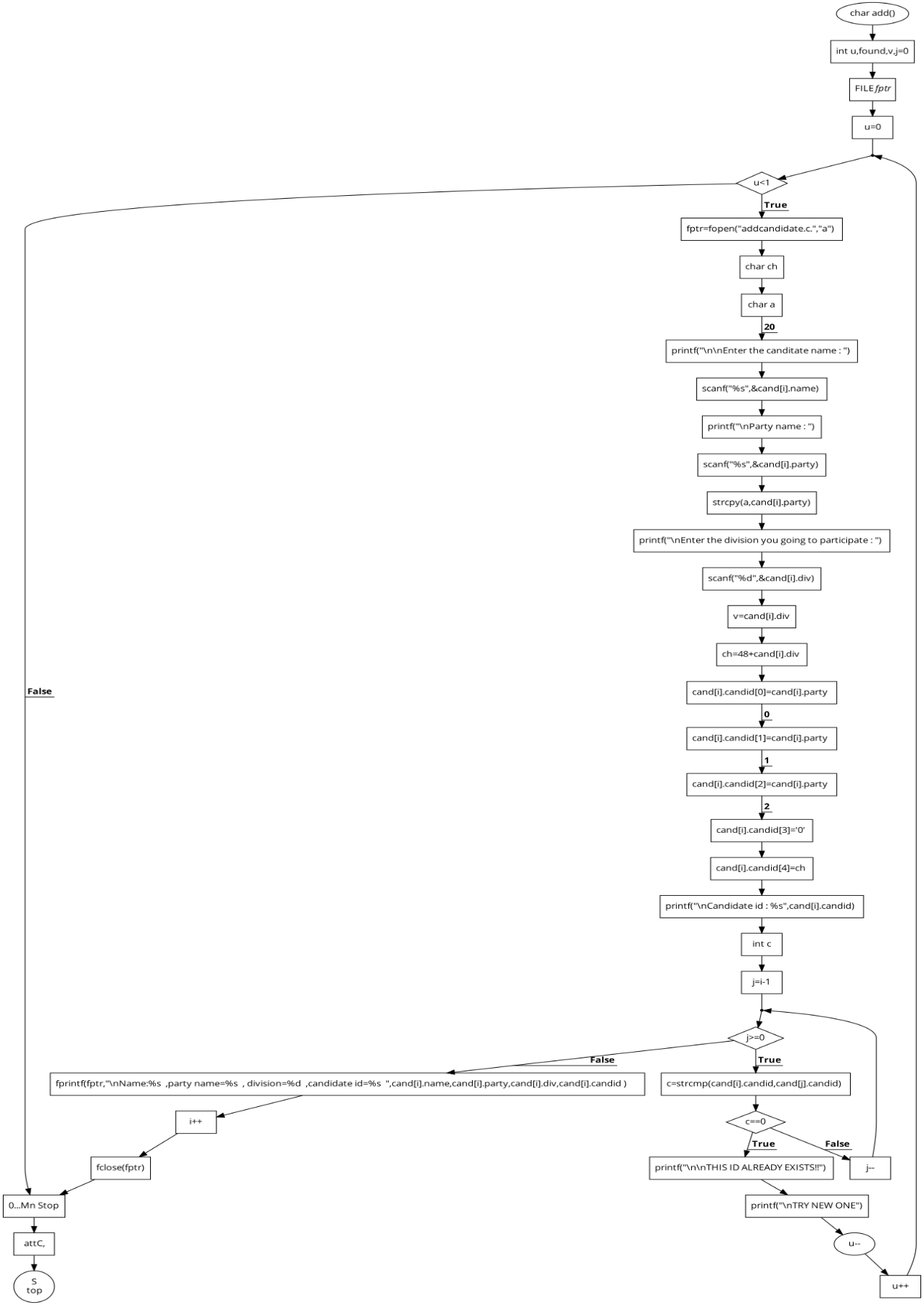
An electoral system or voting system is a set of rules that determine how elections and [referendums](https://en.wikipedia.org/wiki/Referendum) are conducted and how their results are determined. Political electoral systems are organized by governments, while non-political elections may take place in business, [non-profit organisations](https://en.wikipedia.org/wiki/Nonprofit_organization) and informal organisations. These rules govern all aspects of the voting process: when elections occur, [who is allowed to vote](https://en.wikipedia.org/wiki/Suffrage), who can stand as a [candidate](https://en.wikipedia.org/wiki/Candidate), [how ballots are marked and cast](https://en.wikipedia.org/wiki/Voting_method), how the ballots are counted, how votes translate into the election outcome, limits on [campaign spending](https://en.wikipedia.org/wiki/Campaign_finance), and other factors that can affect the result. Political electoral systems are defined by constitutions and electoral laws, are typically conducted by [election commissions](https://en.wikipedia.org/wiki/Election_commission), and can use multiple types of elections for different offices.

Some electoral systems elect a single winner to a unique position, such as prime minister, president or governor, while others elect multiple winners, such as members of parliament or boards of directors. When electing a [legislature](https://en.wikipedia.org/wiki/Legislature), voters may be divided into constituencies with one or more representatives, and may vote directly for individual candidates or for a list of candidates put forward by a [political party](https://en.wikipedia.org/wiki/Political_party) or [alliance](https://en.wikipedia.org/wiki/Electoral_alliance). There are many variations in electoral systems, with the most common systems being [first-past-the-post voting](https://en.wikipedia.org/wiki/First-past-the-post_voting), [block voting](https://en.wikipedia.org/wiki/Multiple_non-transferable_vote), the [two-round (runoff) system](https://en.wikipedia.org/wiki/Two-round_system), [proportional representation](https://en.wikipedia.org/wiki/Proportional_representation) and [ranked voting](https://en.wikipedia.org/wiki/Ranked_voting). Some electoral systems, such as [mixed systems](https://en.wikipedia.org/wiki/Mixed_electoral_system), attempt to combine the benefits of non-proportional and proportional systems.

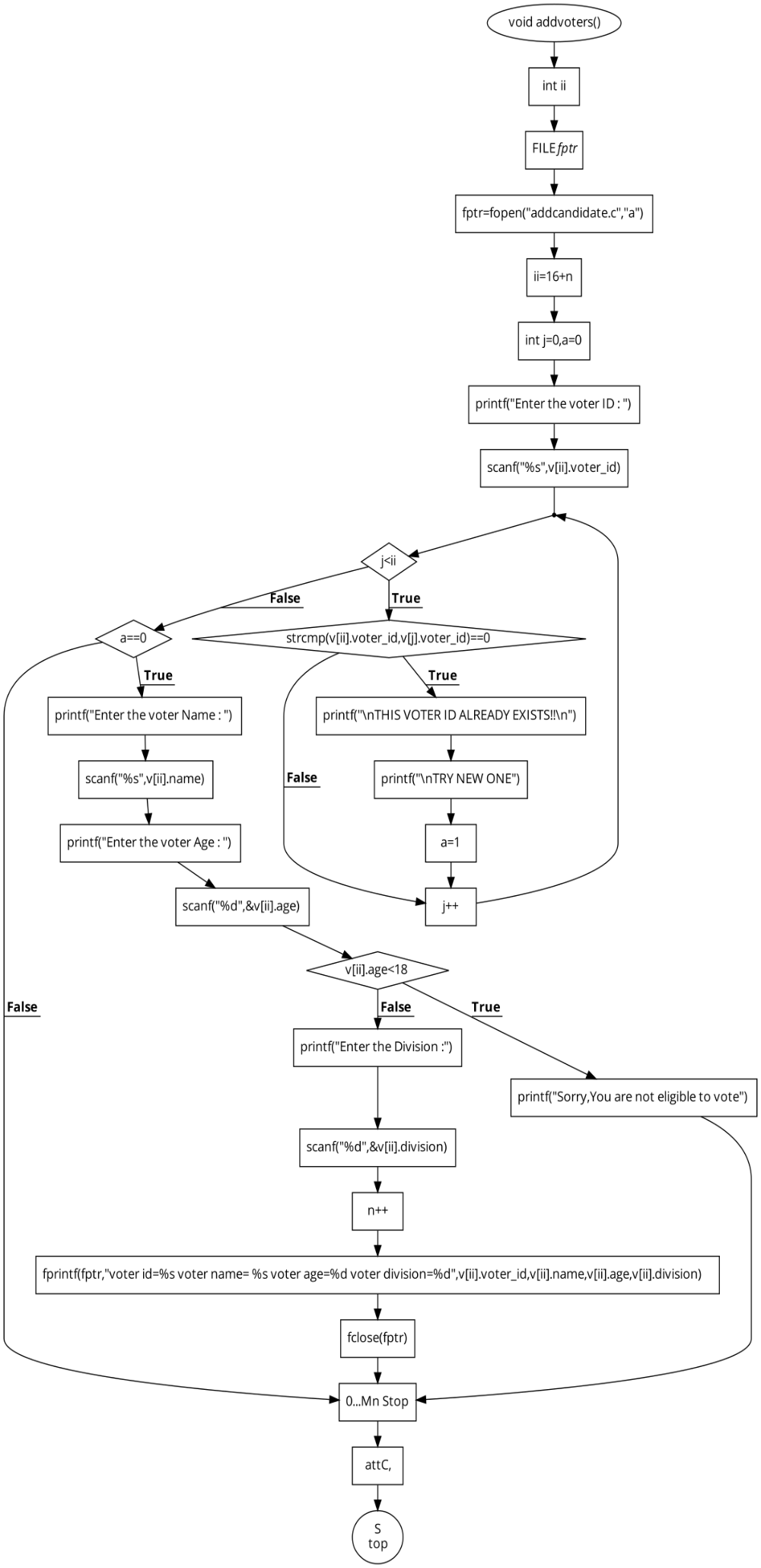
The study of formally defined electoral methods is called [social choice theory](https://en.wikipedia.org/wiki/Social_choice_theory) or voting theory, and this study can take place within the field of [political science](https://en.wikipedia.org/wiki/Political_science), [economics](https://en.wikipedia.org/wiki/Economics), or [mathematics](https://en.wikipedia.org/wiki/Mathematics), and specifically within the subfields of [game theory](https://en.wikipedia.org/wiki/Game_theory) and [mechanism design](https://en.wikipedia.org/wiki/Mechanism_design). Impossibility proofs such as [Arrow's impossibility theorem](https://en.wikipedia.org/wiki/Arrow%27s_impossibility_theorem) demonstrate that when voters have three or more alternatives, no [preferential voting system](https://en.wikipedia.org/wiki/Ranked_voting) can guarantee the race between two candidates remains unaffected when an irrelevant candidate participates or drops out of the election.

**FLOWCHARTS:**

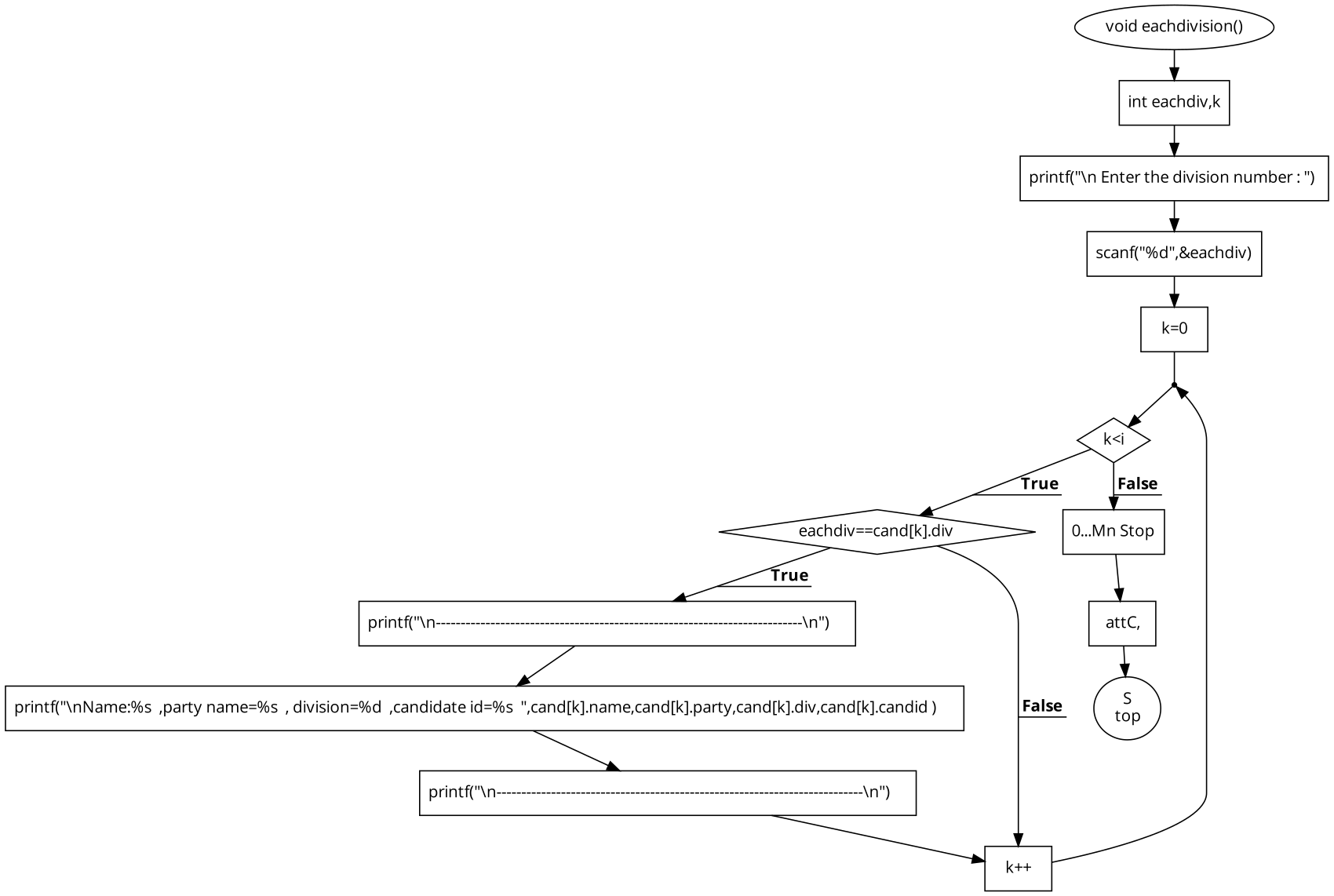
Addcandidates():



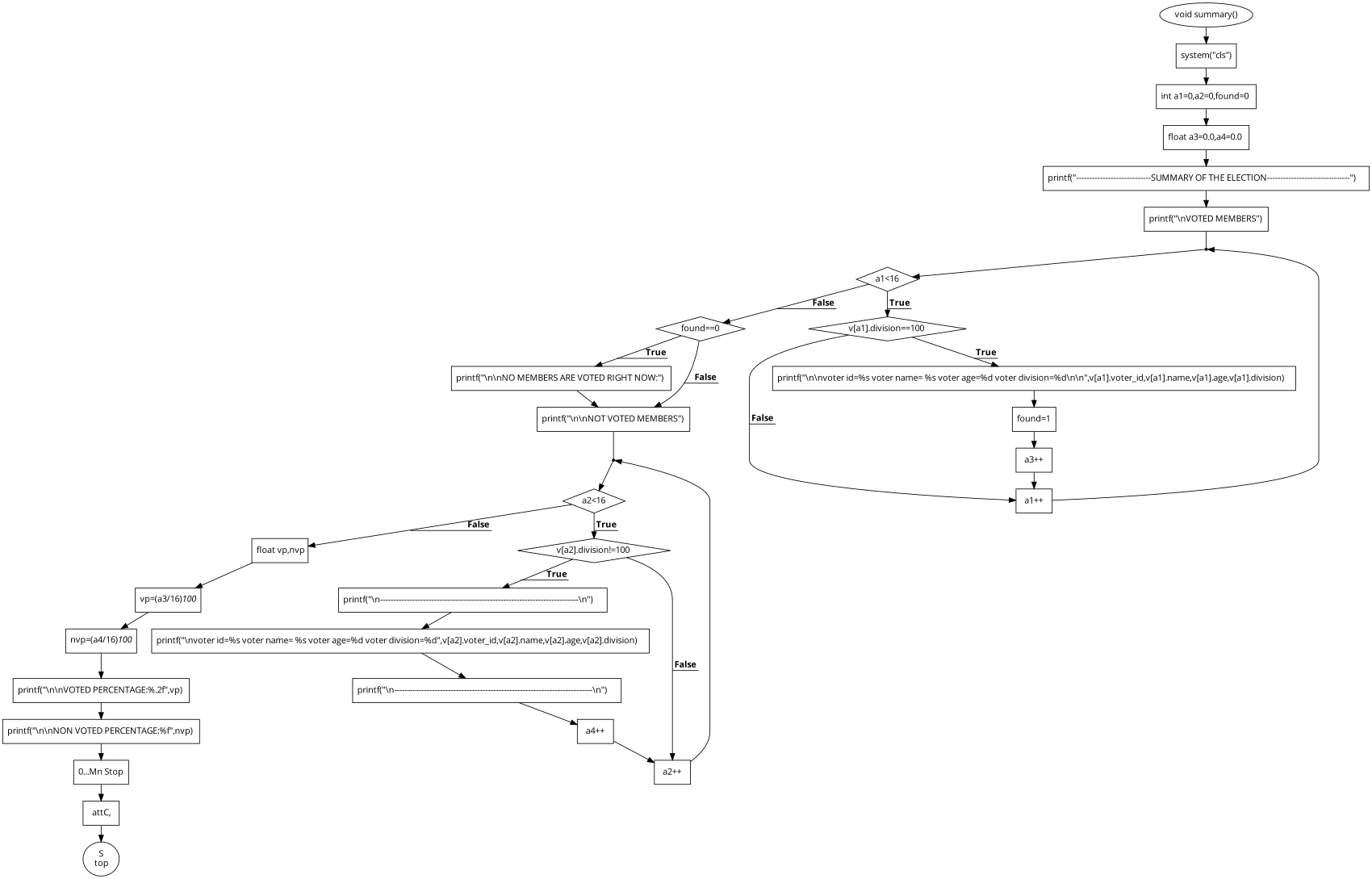
Addvoters():



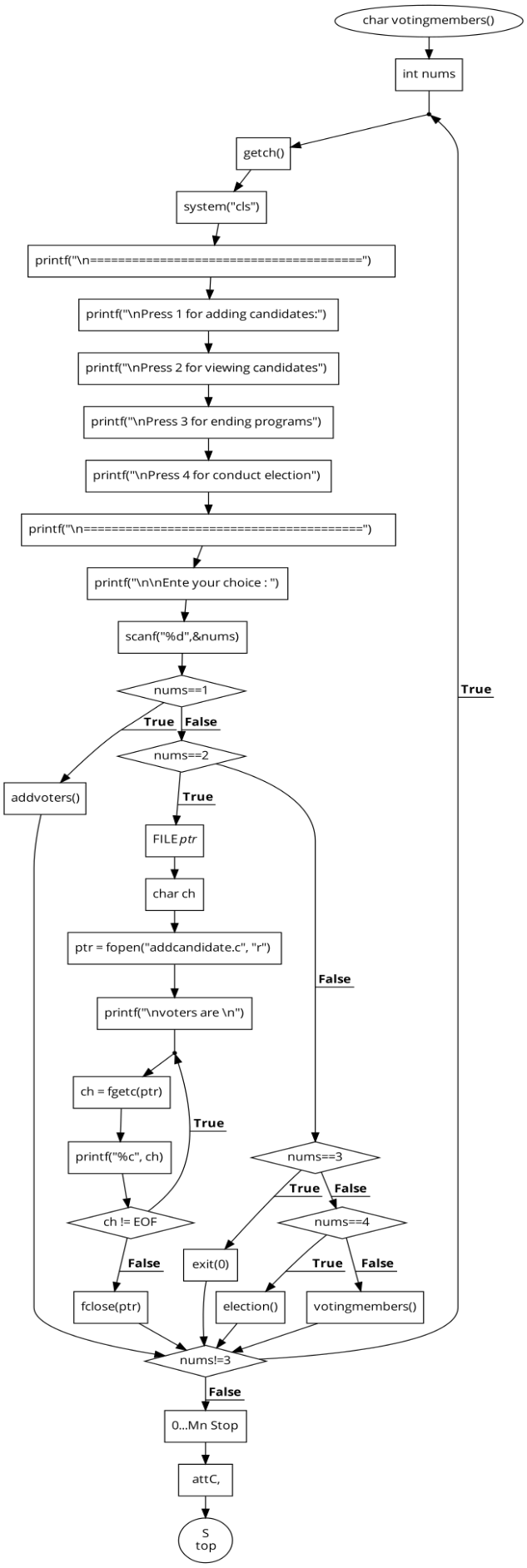
Eachdivision():



Summary():

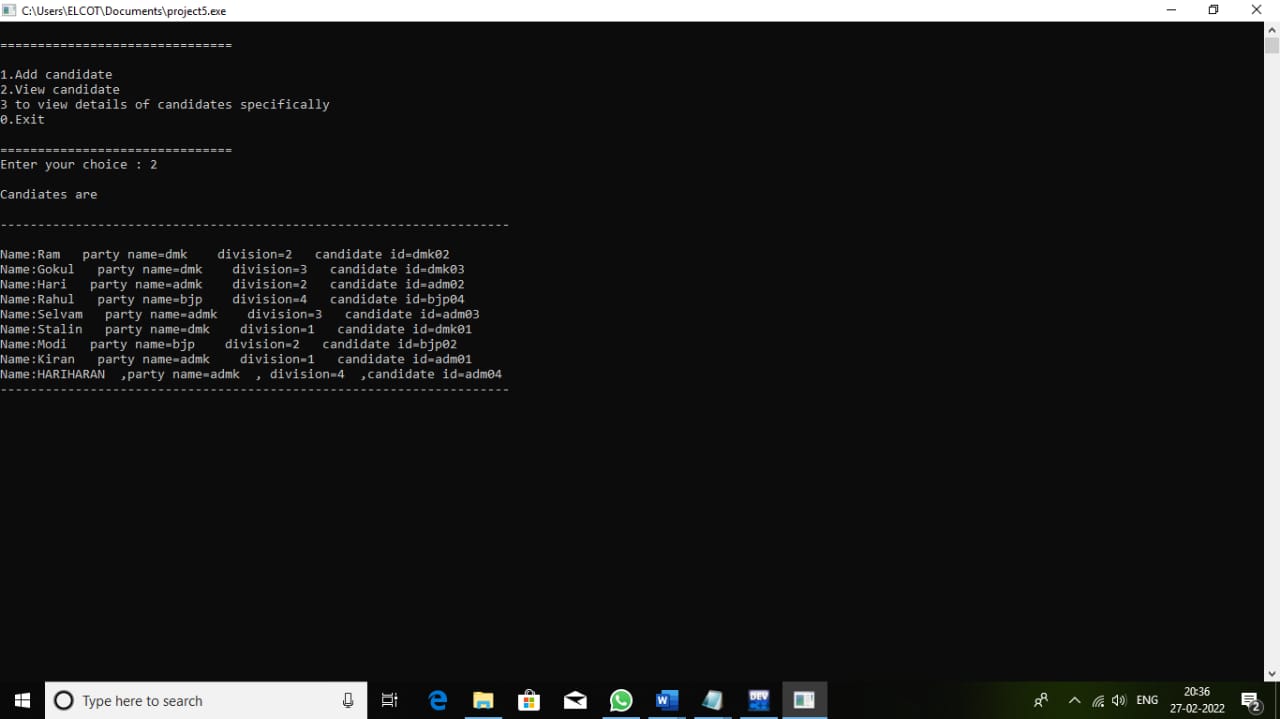


Votingmembers():



**SCREEN SHOTS:**

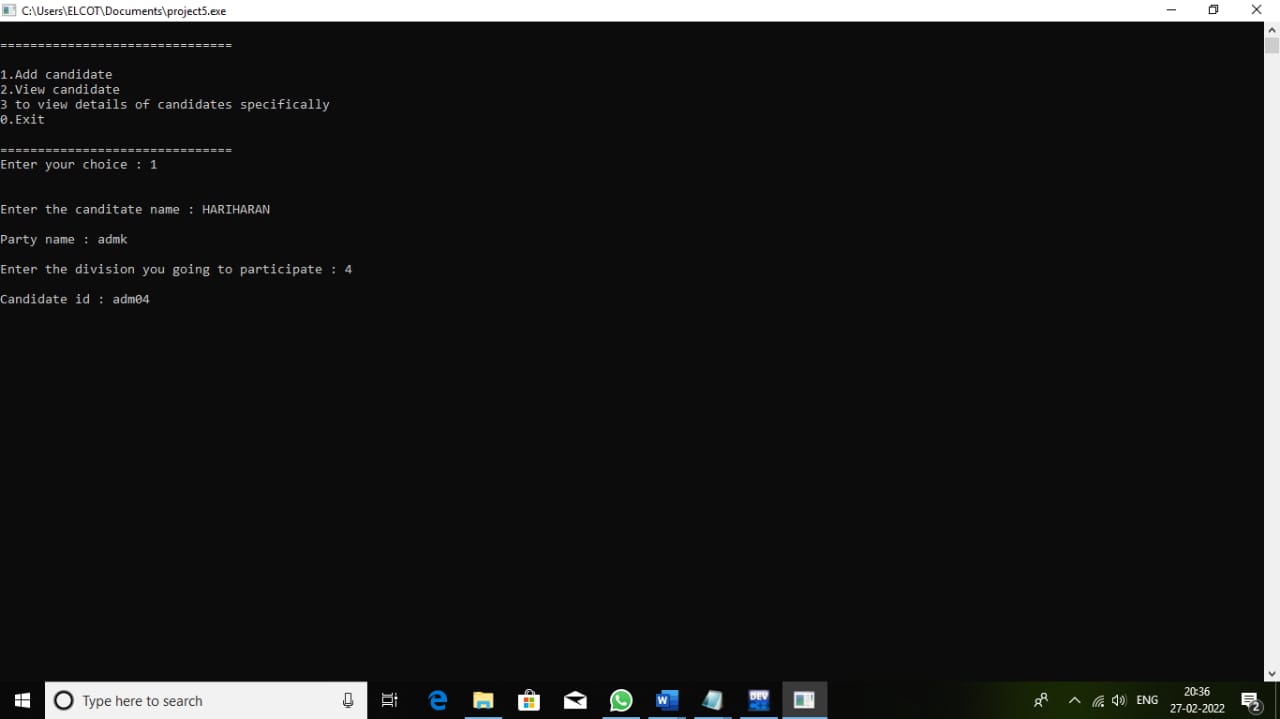
Displaying of candidates:

****

viewcandidates():

This function is for displaying all the candidates standing in the election. The voters can be able to view the candidates according to their party and division.

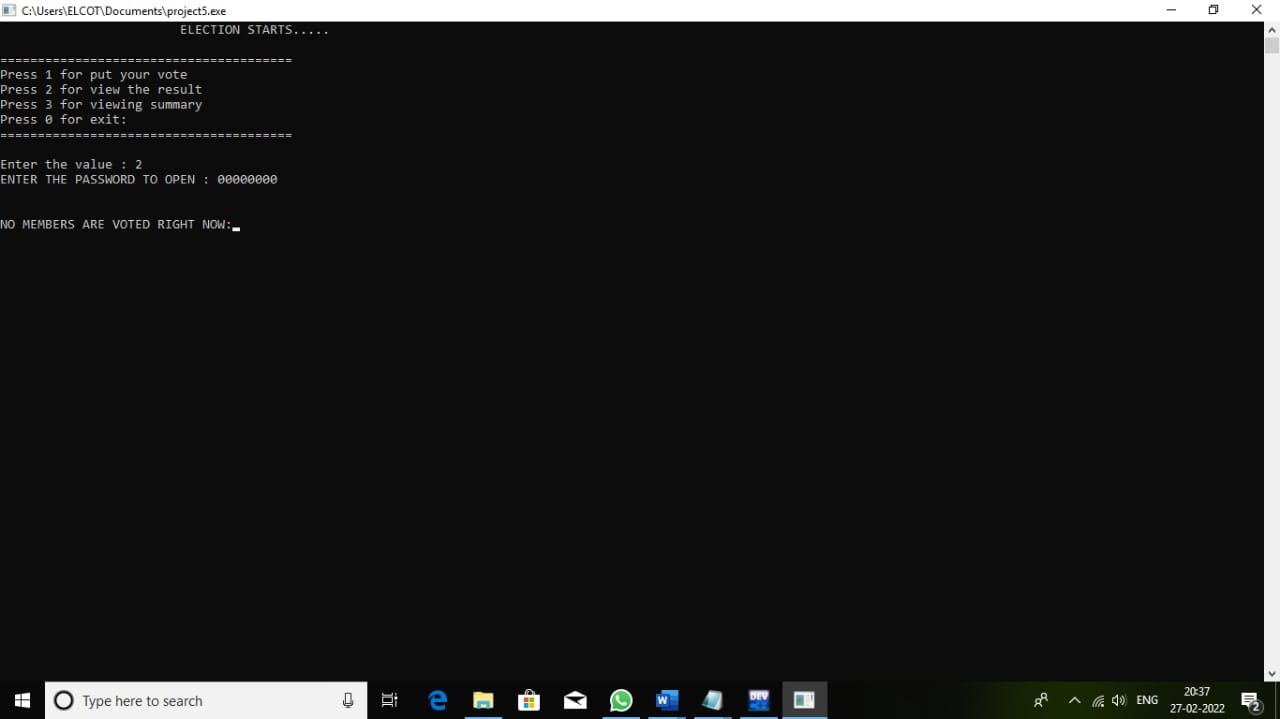
Adding candidates:

****

add():

This function is given for adding the new candidates to the candidate file by getting their details like name, party and division. The candidate id for them will be auto generated.

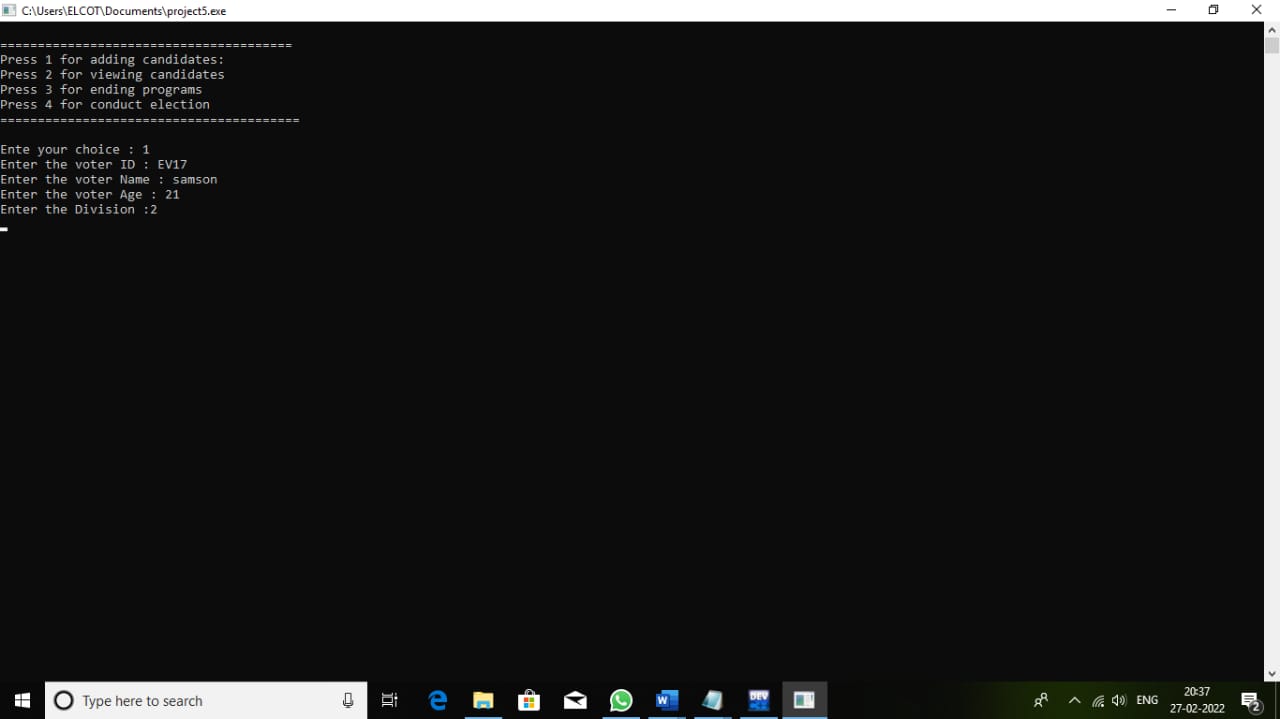
Voting options:

****

votingmembers():

This function is to the voting options, where user can add and view candidate.

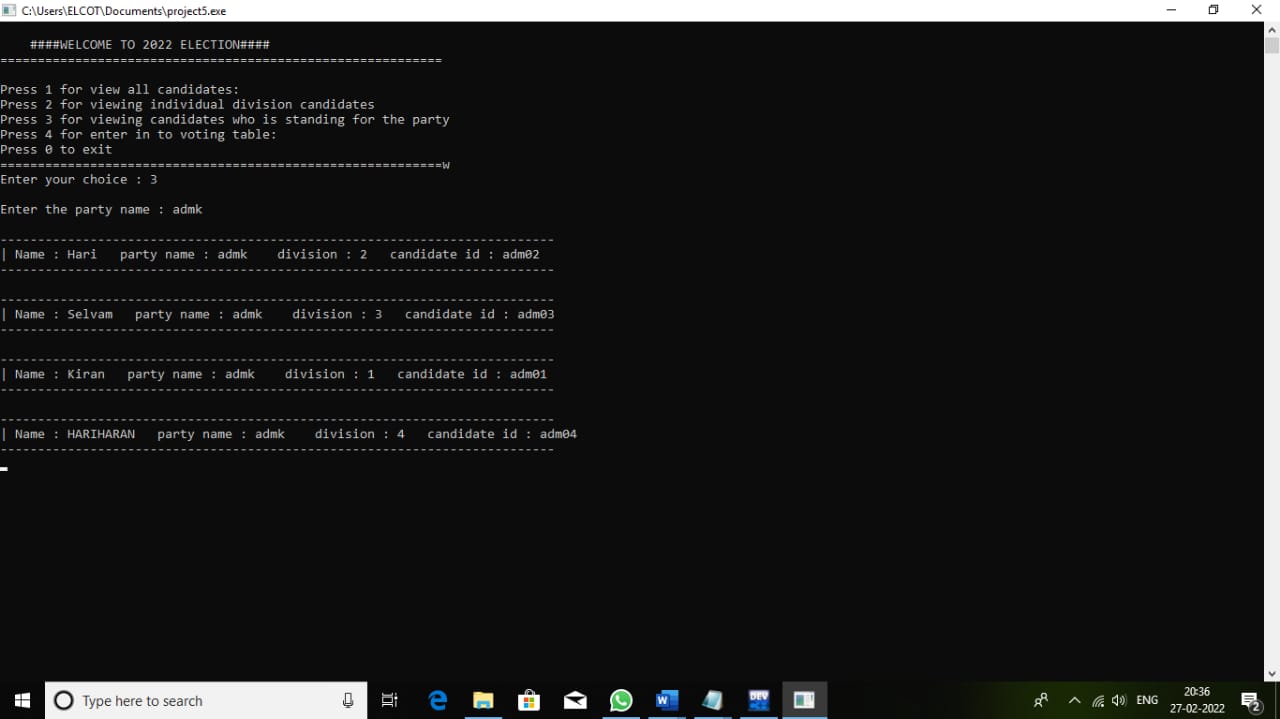
Adding voters:

****

addvoters():

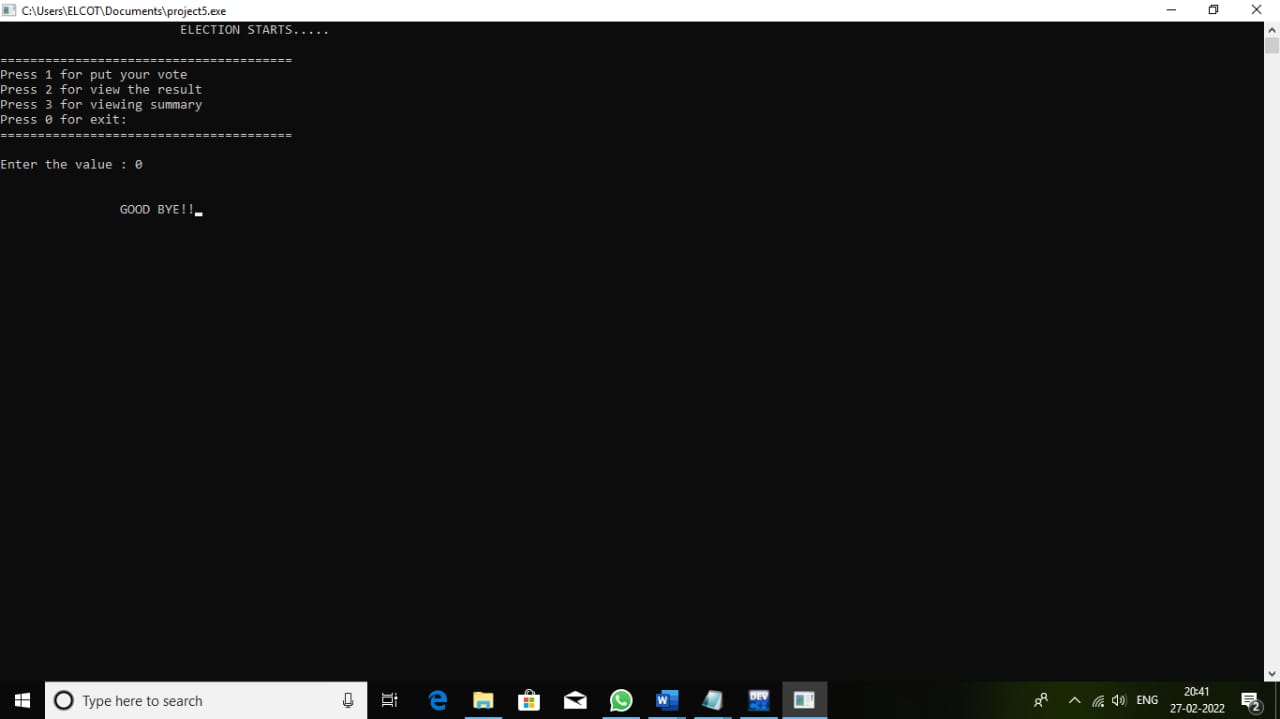
It is to add the voters into the voters file by getting their personal details like voter id, name, age and division where they are going to cast their vote.

Candidates by party wise:

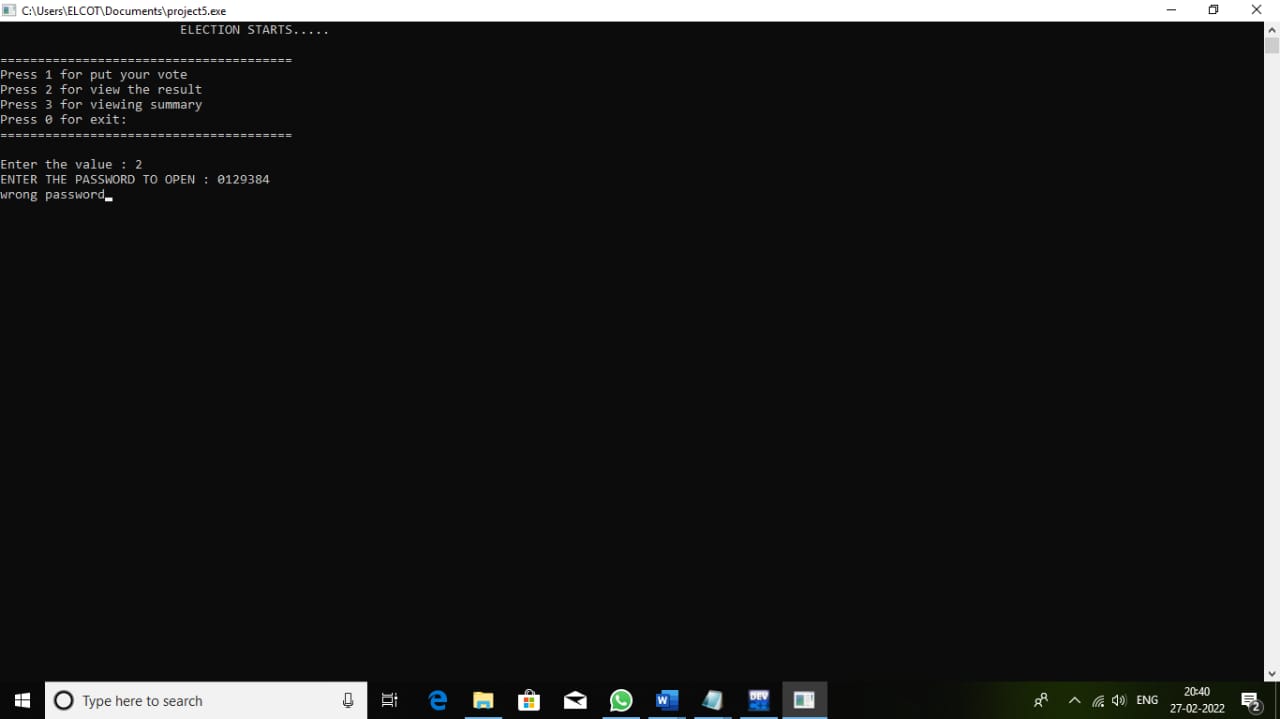
****

Here the user can be able to view the candidates based on the party. User have to give the party, in which the user wants to view the candidates.

Exit system:

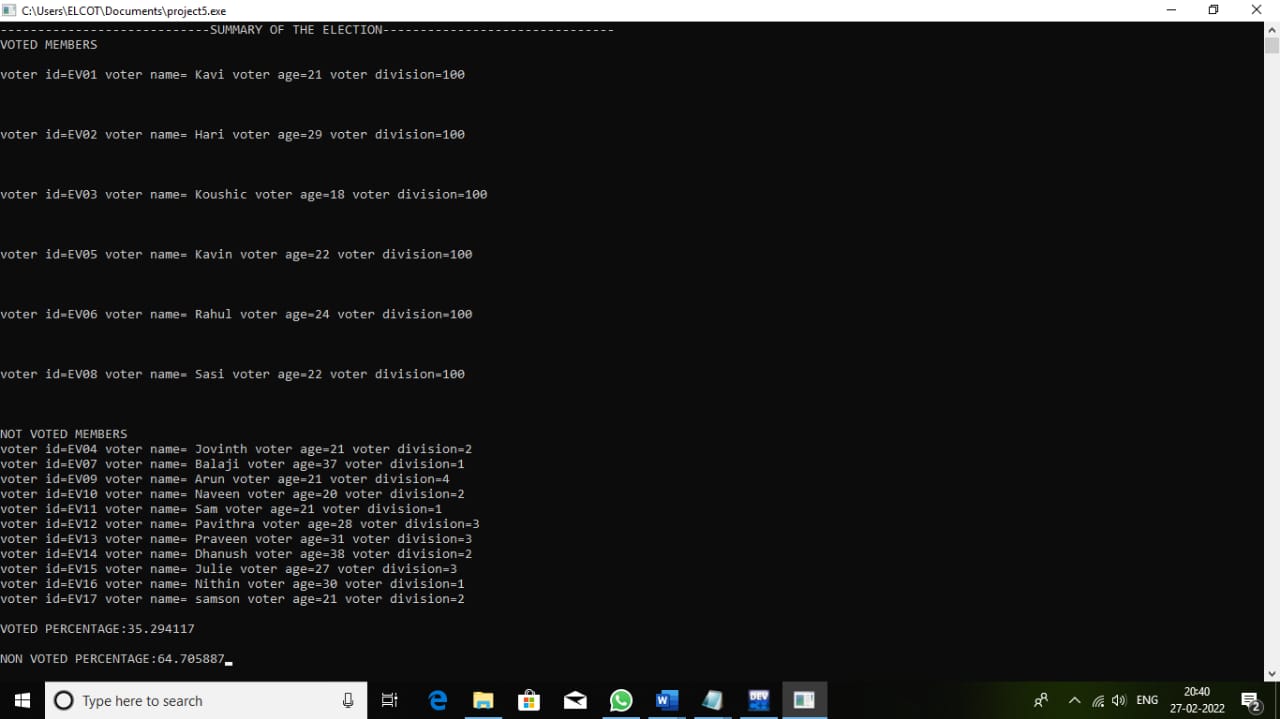
****

There is no special function for exiting, simply pressing 0 will exit from the program.

Opening voters table to count the votes: ****

Here we have given a security lock for the voting result system. The administrator can only open the result by giving a password.

Vote summary:

****

After voting every user can able to see the voting summary as showed above.

**SAMPLE CODES:**

**struct candidates**

**{**

**char candid[max];**

**char name[max];**

**char party[max];**

**int div;**

**int count;**

**}cand[20];**

**struct candidates cand[]={**

**{"dmk02","Ram","dmk",2},**

**{"dmk03","Gokul","dmk",3},**

**{"adm02","Hari","admk",2},**

**{"bjp04","Rahul","bjp",4},**

**{"adm03","Selvam","admk",3},**

**{"dmk01","Stalin","dmk",1},**

**{"bjp02","Modi","bjp",2},**

**{"adm01","Kiran","admk",1}};**

**struct voters**

**{**

**char voter\_id[10];**

**char name[20];**

**int age;**

**int division;**

**char status[1];**

**}v[25];**

**struct voters v[]={**

**{"EV01","Kavi",21,4},**

**{"EV02","Hari",29,1},**

**{"EV03","Koushic",18,1},**

**{"EV04","Jovinth",21,2},**

**{"EV05","Kavin",22,4},**

**{"EV06","Rahul",24,2},**

**{"EV07","Balaji",37,1},**

**{"EV08","Sasi",22,3},**

**{"EV09","Arun",21,4},**

**{"EV10","Naveen",20,2},**

**{"EV11","Sam",21,1},**

**{"EV12","Pavithra",28,3},**

**{"EV13","Praveen",31,3},**

**{"EV14","Dhanush",38,2},**

**{"EV15","Julie",27,3},**

**{"EV16","Nithin",30,1}};**

**char votings();**

**void eachdivision();**

**char voters();**

**char add();**

**char viewcandidates();**

**char votingmembers();**

**void addvoters();**

**void viewvoters();**

**char election();**

**void votes();**

**void go();**

**void results();**

**void summary();**

**char viewcandidates()**

**{**

**FILE\* ptr;**

**char ch;**

**ptr = fopen("addcandidate.c.", "r");**

**printf("\nCandiates are \n");**

**printf("\n--------------------------------------------------------------------\n");**

**do {**

**ch = fgetc(ptr);**

**printf("%c", ch);**

**} while (ch != EOF);**

**printf("\n--------------------------------------------------------------------\n");**

**fclose(ptr);**

**}**

**TASKS PERFORMED BY TEAM MEMBERS:**

|  |  |
| --- | --- |
| TEAM MEMBERS | TASKS |
| HARIHARAN | Created addcandidates and viewcandidates files.  All Menu options.  Work done:   * votings() * eachdivision() * viewcandidates * votingmembers() * election() * votes() * results() |
| KAVIYARASU | Validation of both candidate’s and voter’s record.  Register voter option.  Flowchart.  Work done:   * voters() * add() * addvoters() * viewvoters() * go() |
| KOUSHIC | Summary of voting  Comment lines. |
| JOVINTH | Added default voters and candidates to the files. |

**CONCLUSION:**

By doing this project, we get an opportunity to learn many things related to C programming language and coding knowledge. And we came to know about the voting process and the voting machine. In this program we have included many functions for the voting system to make it realistic.