ESTD:2001
An Institute with a Difference

# RNS Institute of Technology

**(AICTE** Approved, **VTU** Affiliated and **NAAC 'A'** Accredited)

**Department of Information Science and Engineering**

(**Accredited** by **NBA** for the *Academic Years 2018-19, 2019-20 and 2020-21*)

## Data structures Laboratory-18CSL38

# Voting System Using Linked List

**Staff in Charge: Mrs. Vinutha Lavanya**
**Designation**: **Assistant Professor**

**Carried out by:**
**Nidish G (1RN19IS94)**
**Phanish SN (1RN19IS100)**
**Hemanth Kumar M (1RN19IS063)**

# CONTENTS

- Abstract ( slide 3 )
- Introduction ( slide 4 )
- Objective of the project ( slide 5 )
- Data Structure ( Single Linked List ( slide 6))
- System requirements ( slide 7)
- Project Architecture

  (Flowchart / Block Diagram / DFDs representing overall working of project with explanation: 2 slides)
- Implementation modules ( slide 10 to 19)

  ( All functions prototypes (declarations) of the implemented code and explanations on how the function works)
- Results ( slide 20 to 25 )
- ( Snapshots showing all the steps by steps results )
- Conclusion & Future Enhancements - slide 26
- References ( Web-links )-slide 27

# Abstract

- It represents simulation of real time voting with multiple contestants and analyzing their results in different aspects.

- Our project uses basic functions to simulate real life voting system by calculating voter stats (randomly generated profiles),stats of the contestants and declaring the winner based on those calculated stats by considering appropriate conditions.

# Introduction

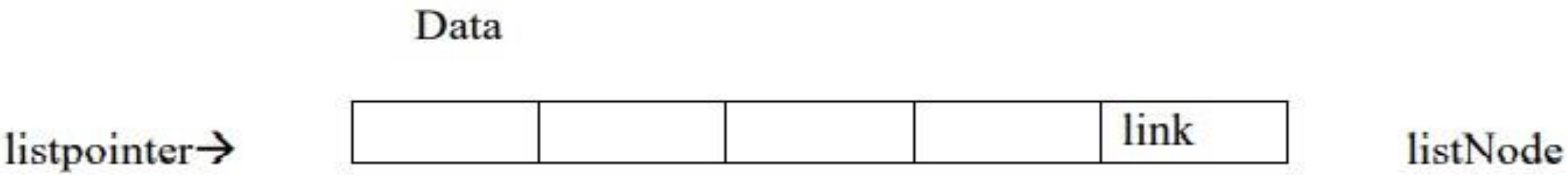
- Our project gives user a real time feel of the voting system.
- It allows user to choose between multiple contestants and vote according to their desired choice .
- It allows election commission(here the programmer) to check the results and declare the winner .
- It allows the programmer to generate numerous random multiple votes to simulate a large scale voting system without manual user input easily and effectively.
- User(Voter) can also enter manual inputs(Votes) .

# Objective of the project

- The main objective of our project is to show how electoral voting system works in real life.
- The main specifications of our project are:
- To generate random ‘N’ voters
- To count the votes
- To calculate the stats based on no. of votes
- To declare the winner based on calculated stats

# Data Structure

- The Data Structure used in this project is Single Linked List(SLL).



Data

listpointer→

link

listNode

# System Requirements

- This project was developed on an online compiler(repl.it).
- It requires a basic computer with an active internet connection.
- It requires all the basic C libraries.
- Visit our project at :

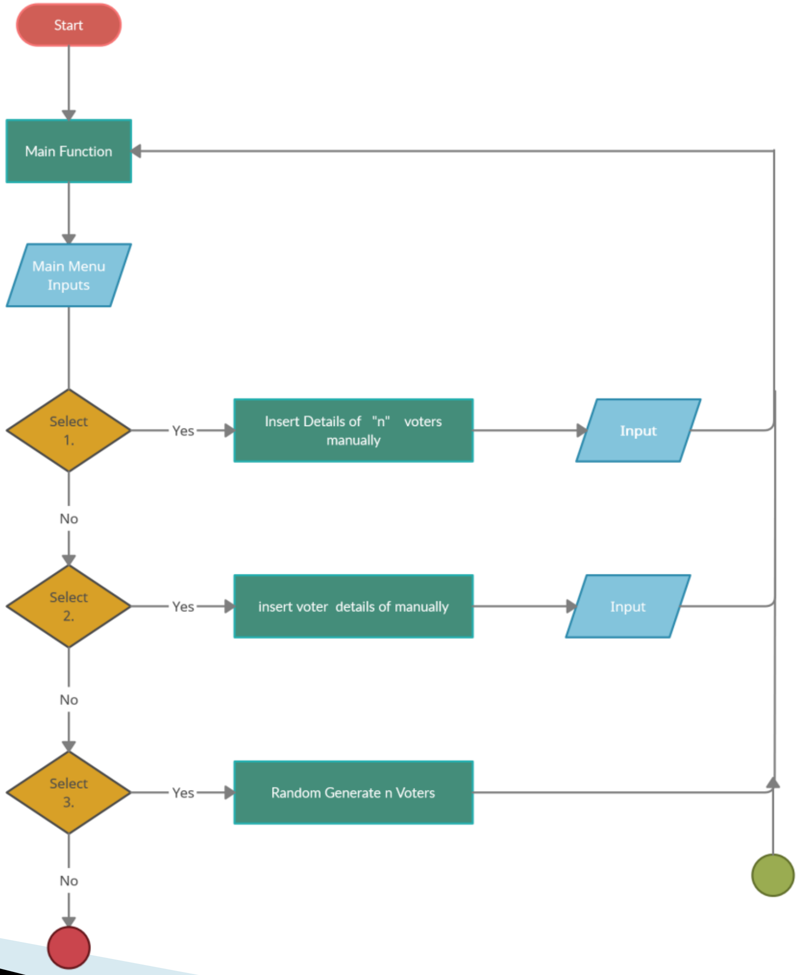
https://repl.it/@NidPlays/DSA-Project-Work#main.c

# System Requirements

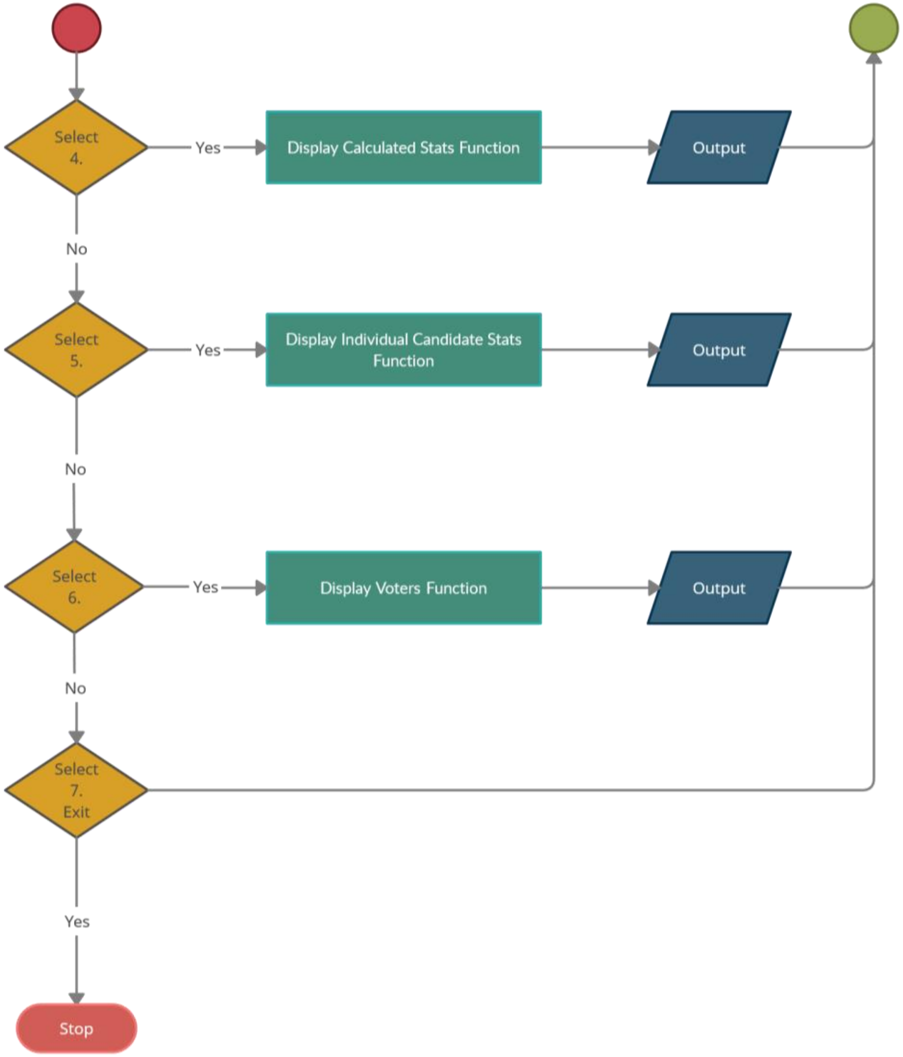
- PC Specs
- Processor: Intel atom or higher
- Online Internet connection
- Ram: 4GB
- Os: windows,linux,macos
- Online compiler: https://repl.it

# Project Architecture

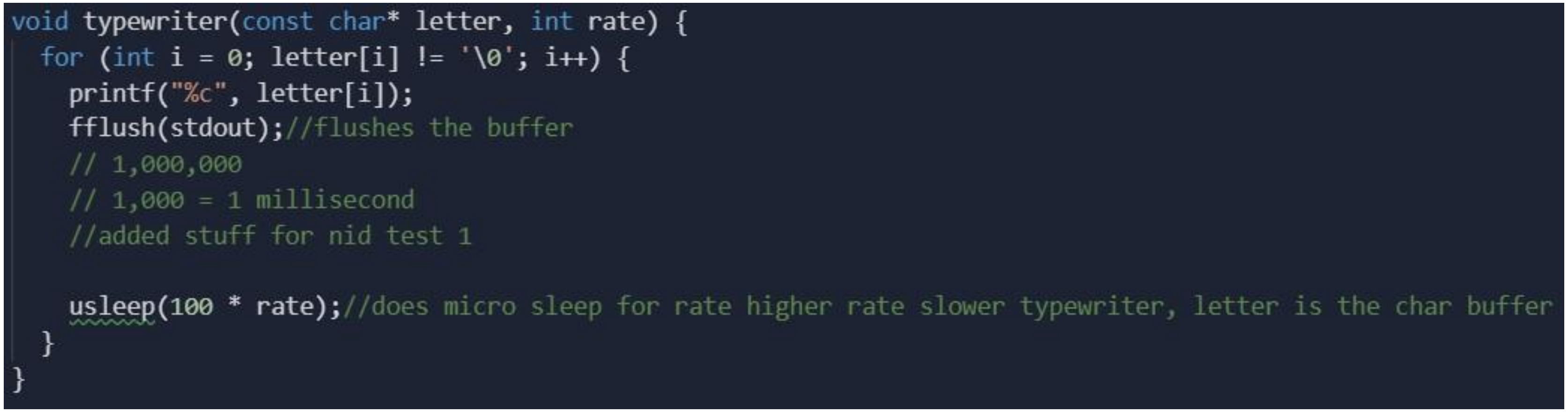
- The following flowchart shows step by step procedure of how our project works:



# Project Architecture



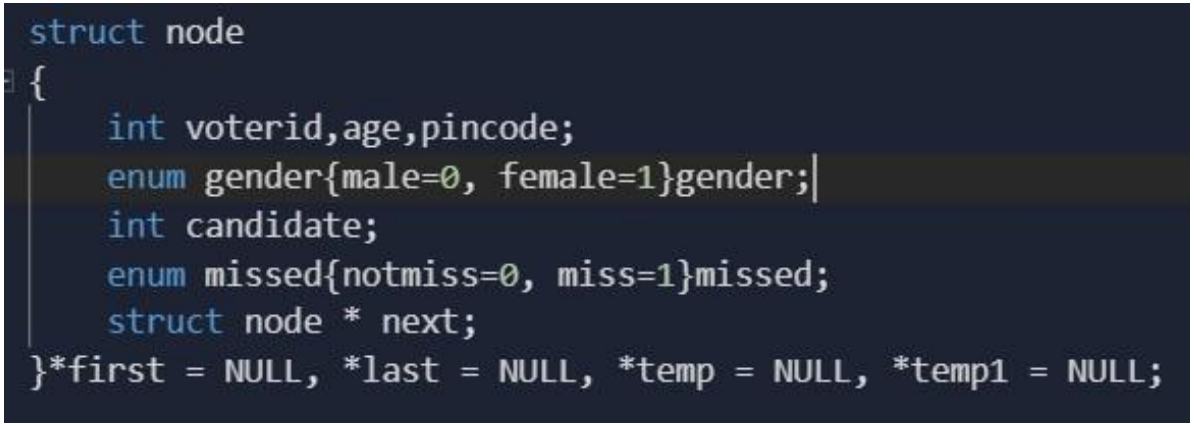
# Implementation modules



```
void typewriter(const char* letter, int rate) {
  for (int i = 0; letter[i] != '\0'; i++) {
    printf("%c", letter[i]);
    fflush(stdout);//flushes the buffer
    // 1,000,000
    // 1,000 = 1 millisecond
    //added stuff for nid test 1

    usleep(100 * rate);//does micro sleep for rate higher rate slower typewriter, letter is the char buffer
  }
}
```

- This function(Typewriter) simulates the typing experience of typewriter.
- In the above function, 'usleep' stops the execution of the program for the given time(micro seconds).

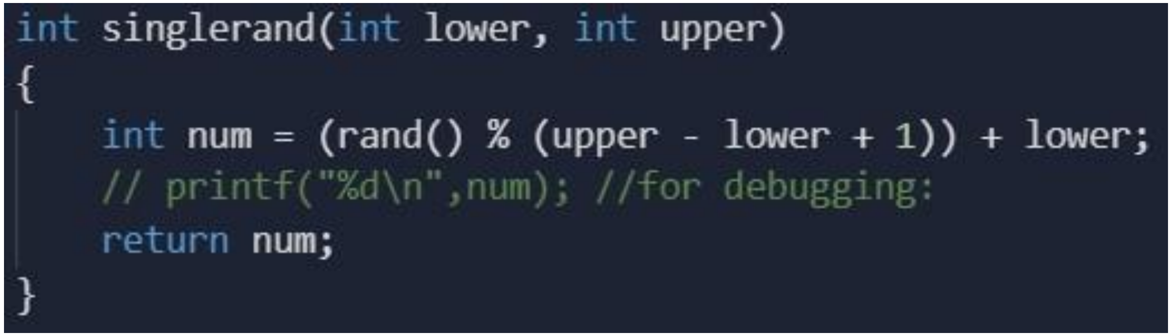
# Implementation modules



```c
struct node
{
    int voterid,age,pincode;
    enum gender{male=0, female=1}gender;
    int candidate;
    enum missed{notmiss=0, miss=1}missed;
    struct node * next;
}*first = NULL, *last = NULL, *temp = NULL, *temp1 = NULL;
```

- This is the main structure which
- is used to store data

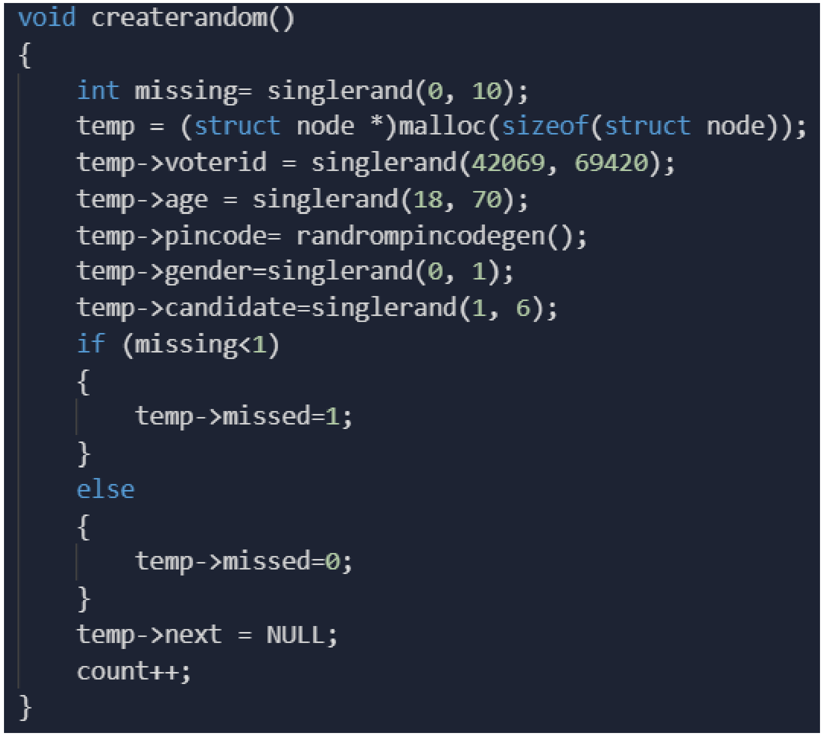
# Implementation modules



```c
int singlerand(int lower, int upper)
{
    int num = (rand() % (upper - lower + 1)) + lower;
    // printf("%d\n",num); //for debugging:
    return num;
}
```

- This function is used to generate random inputs in the program.
- The function randomly generates the inputs between the upper and the lower limits(range).

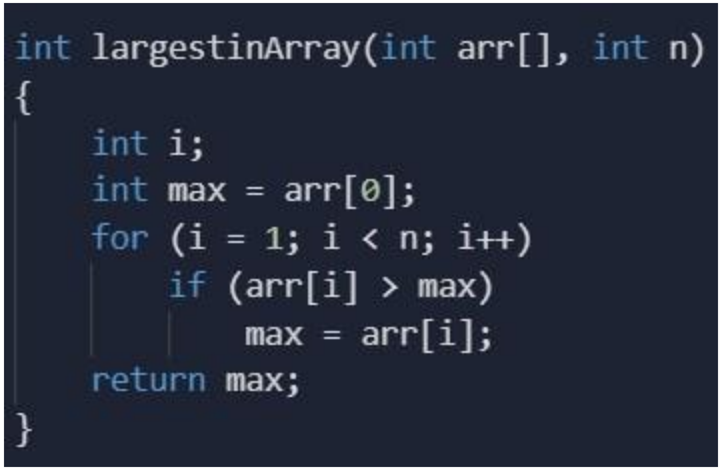
# Implementation modules



```c
void createrandom()
{
    int missing= singlerand(0, 10);
    temp = (struct node *)malloc(sizeof(struct node));
    temp->voterid = singlerand(42069, 69420);
    temp->age = singlerand(18, 70);
    temp->pincode= randrompincodegen();
    temp->gender=singlerand(0, 1);
    temp->candidate=singlerand(1, 6);
    if (missing<1)
    {
        temp->missed=1;
    }
    else
    {
        temp->missed=0;
    }
    temp->next = NULL;
    count++;
}
```

- This function is used when we insert new voters using random generation

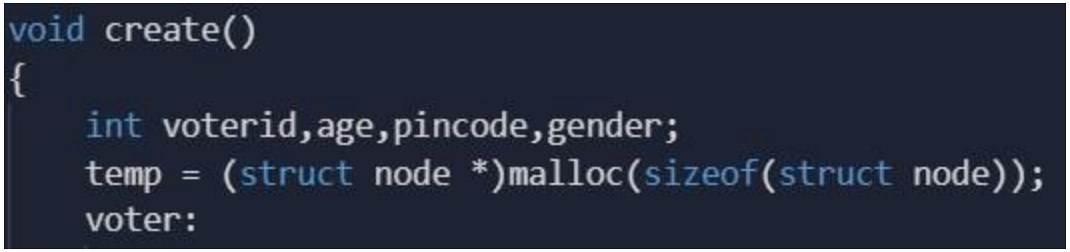
# Implementation modules



```
int largestinArray(int arr[], int n)
{
    int i;
    int max = arr[0];
    for (i = 1; i < n; i++)
        if (arr[i] > max)
            max = arr[i];
    return max;
}
```

- This function is used to find the largest element in an array

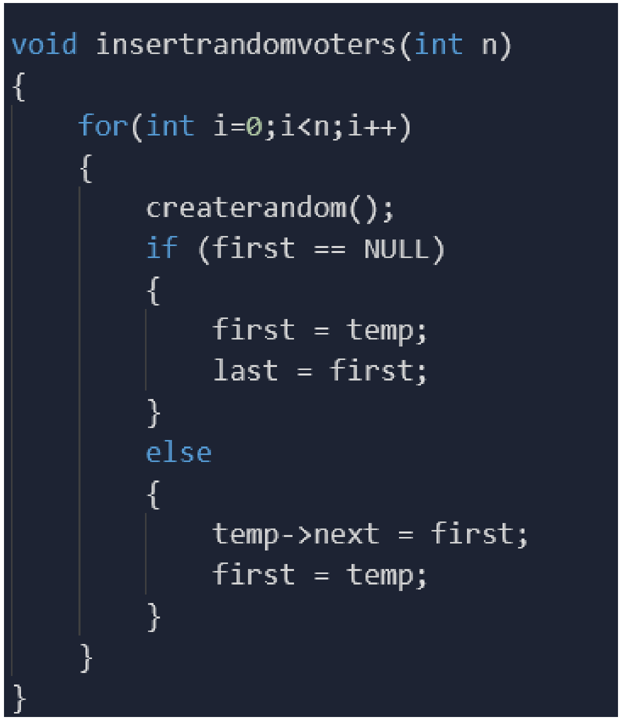
# Implementation modules



```
void create()
{
    int voterid,age,pincode,gender;
    temp = (struct node *)malloc(sizeof(struct node));
    voter:
```

- This function is used when we insert voter details manually and we need to choose the candidate

# Implementation modules



```
void insertrandomvoters(int n)
{
    for(int i=0;i<n;i++)
    {
        createrandom();
        if (first == NULL)
        {
            first = temp;
            last = first;
        }
        else
        {
            temp->next = first;
            first = temp;
        }
    }
}
```

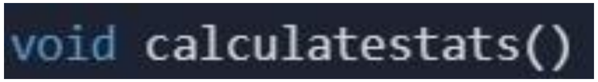
- ➢ This function is used to insert random voters into the linked list

# Implementation modules

```
void individualStats()
```

- This function counts the number of voters, missed voters and nota and displays who won the election and candidate specific statistics

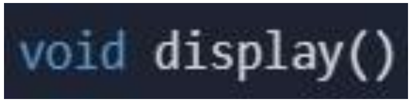
# Implementation modules



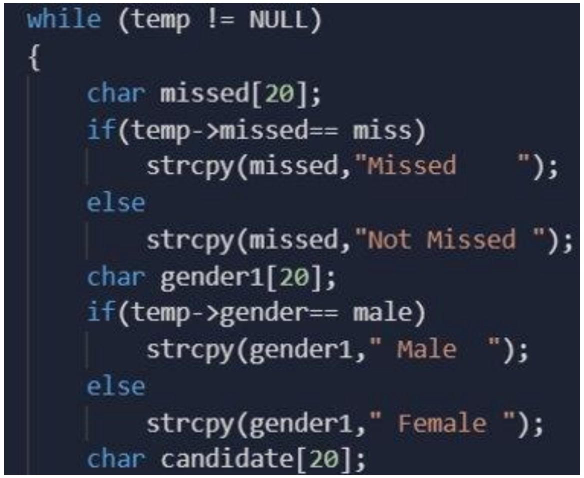
```
void calculatestats()
```

- This function is used to calculate statistics based on the various parameters that is given in the struct, i.e. voters in specific age group, gender of voters, total number of voters, nota voters and missed voters

# Implementation modules



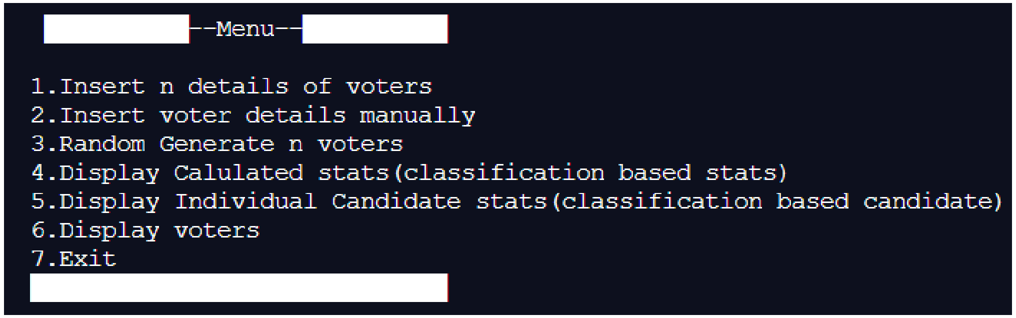
```
void display()
```



```
while (temp != NULL)
{
    char missed[20];
    if(temp->missed== miss)
        strcpy(missed,"Missed    ");
    else
        strcpy(missed,"Not Missed ");
    char gender1[20];
    if(temp->gender== male)
        strcpy(gender1," Male  ");
    else
        strcpy(gender1," Female ");
    char candidate[20];
```

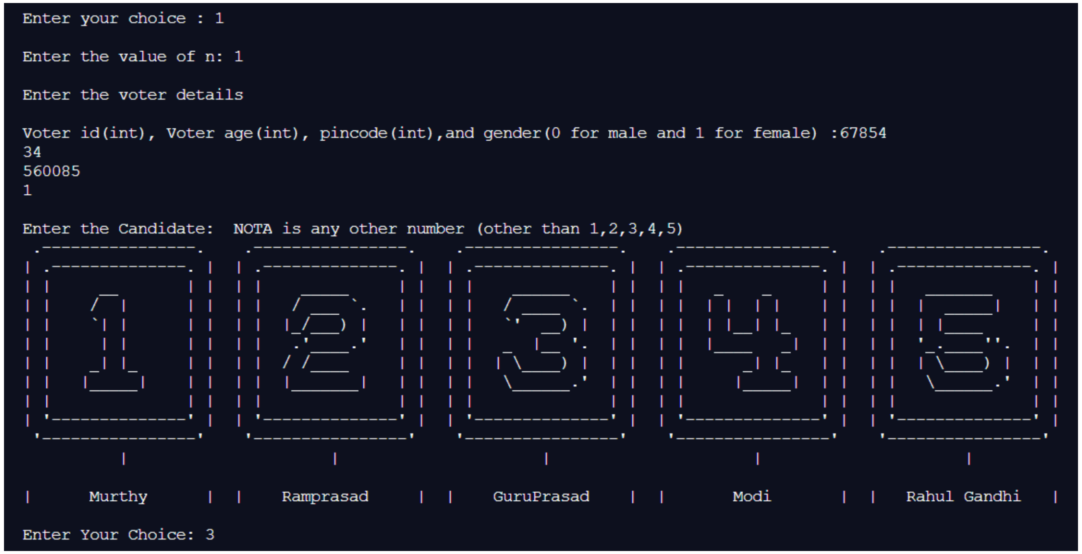
- ➢ This function is the display function which displays all the candidates at once

# Results



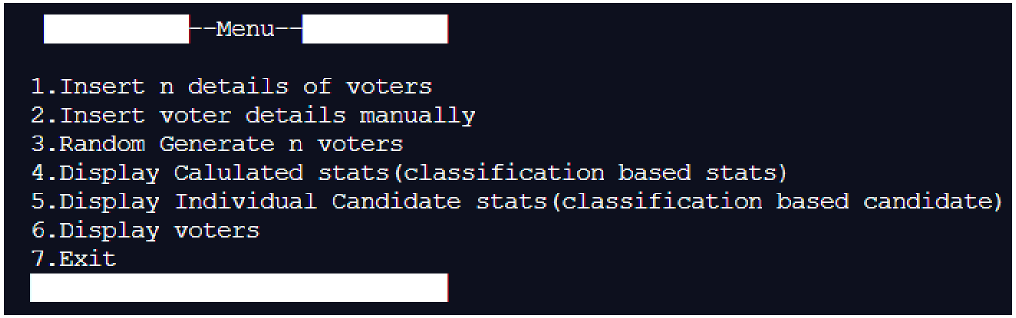
```
--Menu--

1.Insert n details of voters
2.Insert voter details manually
3.Random Generate n voters
4.Display Calulated stats(classification based stats)
5.Display Individual Candidate stats(classification based candidate)
6.Display voters
7.Exit
```

- Output when the first(1.) option is selected



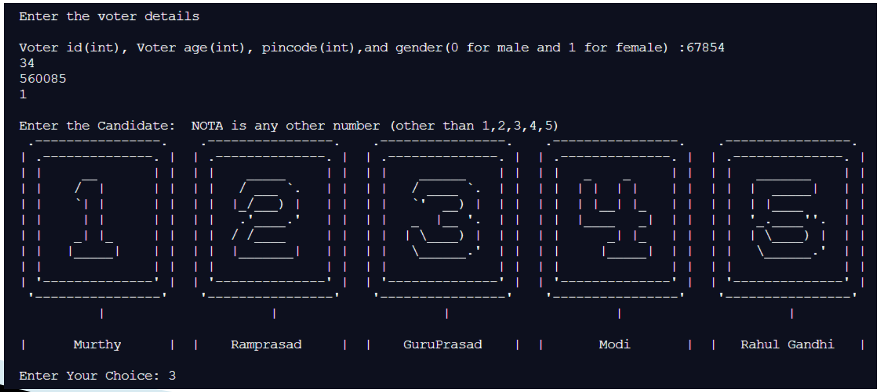
```
Enter your choice : 1

Enter the value of n: 1

Enter the voter details

Voter id(int), Voter age(int), pincode(int),and gender(0 for male and 1 for female) :67854
34
560085
1

Enter the Candidate:  NOTA is any other number (other than 1,2,3,4,5)
 .----------------.   .----------------.   .----------------.   .----------------.   .----------------. 
| .--------------. | | .--------------. | | .--------------. | | .--------------. | | .--------------. |
| |              | | | |              | | | |              | | | |              | | | |              | |
| |     __       | | | |    _____     | | | |    ______    | | | |   _    _     | | | |   _______    | |
| |    /  |      | | | |   / ___ `.   | | | |   / ____ `.  | | | |  | |  | |    | | | |  |  _____|   | |
| |    `| |      | | | |  |_/___) |   | | | |   `'  __) |  | | | |  | |__| |_   | | | |  | |____     | |
| |     | |      | | | |   .'____.'   | | | |   _  |__ '.  | | | |  |____   _|  | | | |  '_.____''.  | |
| |    _| |_     | | | |  / /____     | | | |  | \____) |  | | | |      _| |_   | | | |  | \____) |  | |
| |   |_____|    | | | |  |_______|   | | | |   \______.'  | | | |     |_____|  | | | |   \______.'  | |
| |              | | | |              | | | |              | | | |              | | | |              | |
| '--------------' | | '--------------' | | '--------------' | | '--------------' | | '--------------' |
 '----------------'   '----------------'   '----------------'   '----------------'   '----------------' 
         |                    |                    |                    |                    |

|      Murthy      | |    Ramprasad     | |    GuruPrasad    | |       Modi       | |   Rahul Gandhi   |

Enter Your Choice: 3
```

# Results



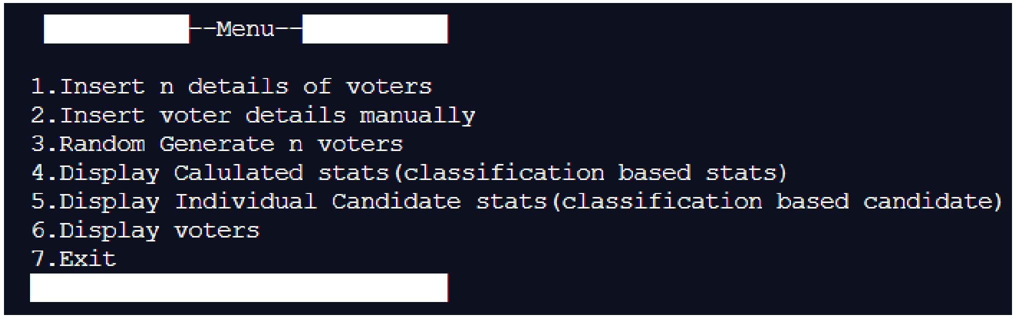
```
--Menu--

1.Insert n details of voters
2.Insert voter details manually
3.Random Generate n voters
4.Display Calulated stats(classification based stats)
5.Display Individual Candidate stats(classification based candidate)
6.Display voters
7.Exit
```

- ➢Output when the second(2.) option is selected.
- ➢It will be same as the first option but we will be allowed to insert the details of only one voter.



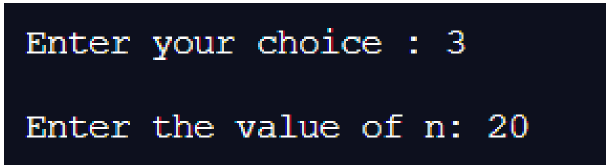
```
Enter the voter details

Voter id(int), Voter age(int), pincode(int),and gender(0 for male and 1 for female) :67854
34
560085
1

Enter the Candidate:  NOTA is any other number (other than 1,2,3,4,5)

|      Murthy      | |    Ramprasad     | |    GuruPrasad    | |       Modi       | |   Rahul Gandhi   |

Enter Your Choice: 3
```

# Results



```
 ██████████--Menu--██████████

1.Insert n details of voters
2.Insert voter details manually
3.Random Generate n voters
4.Display Calulated stats(classification based stats)
5.Display Individual Candidate stats(classification based candidate)
6.Display voters
7.Exit
█████████████████████████████
```

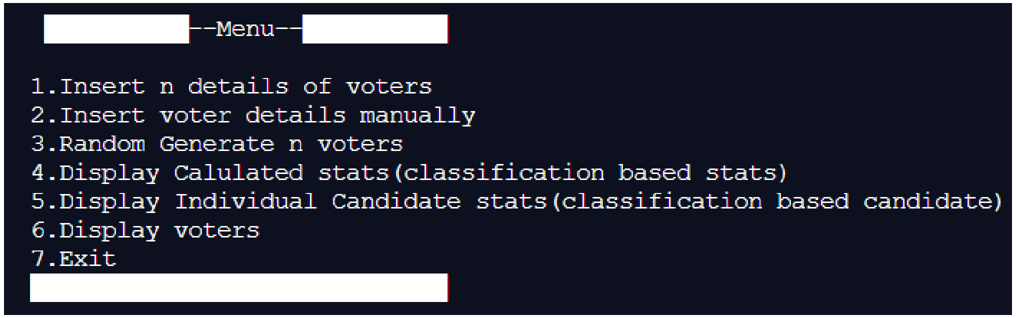
- ➢ Output when the third(3.) option is selected.



```
Enter your choice : 3

Enter the value of n: 20
```

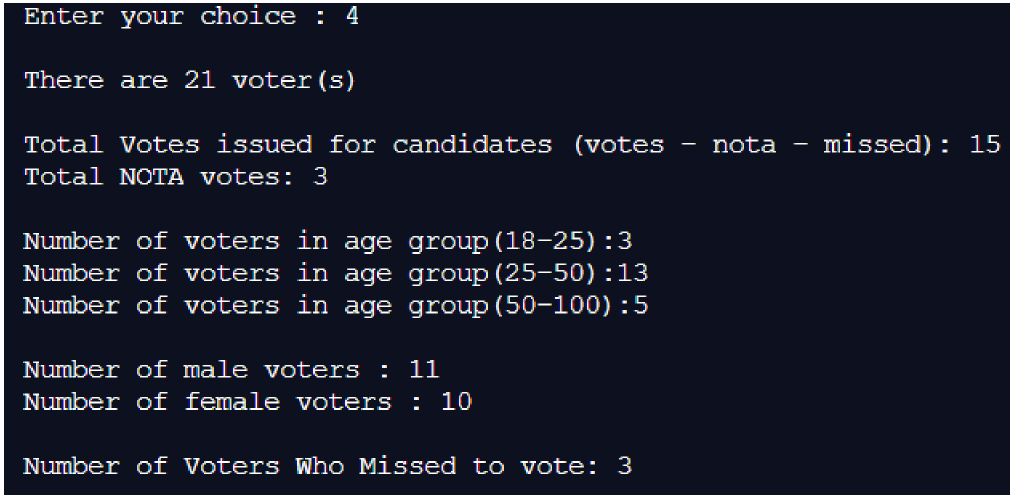
- ➢ Here 20 random voters will be generated.

# Results



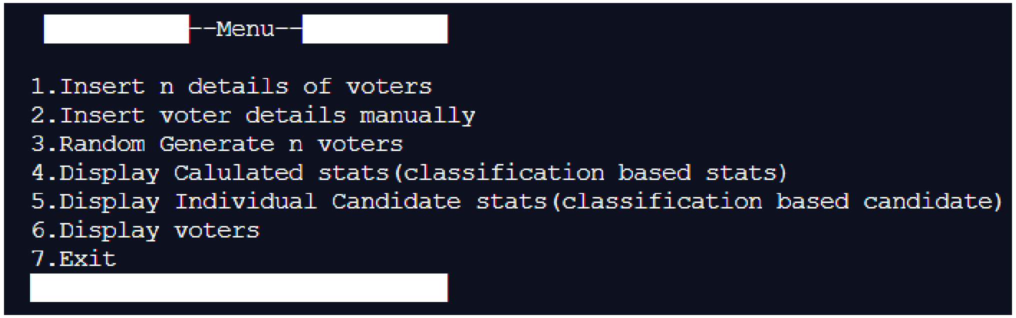
```
          --Menu--

1.Insert n details of voters
2.Insert voter details manually
3.Random Generate n voters
4.Display Calulated stats(classification based stats)
5.Display Individual Candidate stats(classification based candidate)
6.Display voters
7.Exit
```

- Output when the fourth(4.) option is selected.



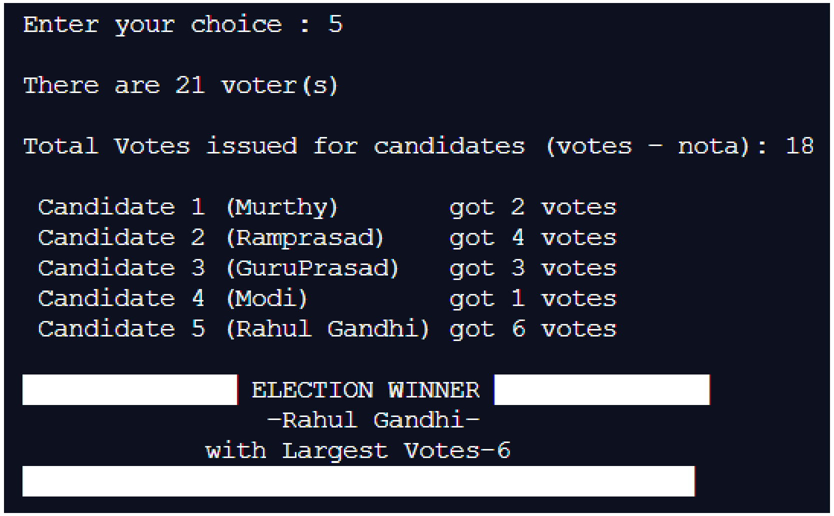
```
Enter your choice : 4

There are 21 voter(s)

Total Votes issued for candidates (votes - nota - missed): 15
Total NOTA votes: 3

Number of voters in age group(18-25):3
Number of voters in age group(25-50):13
Number of voters in age group(50-100):5

Number of male voters : 11
Number of female voters : 10

Number of Voters Who Missed to vote: 3
```

# Results



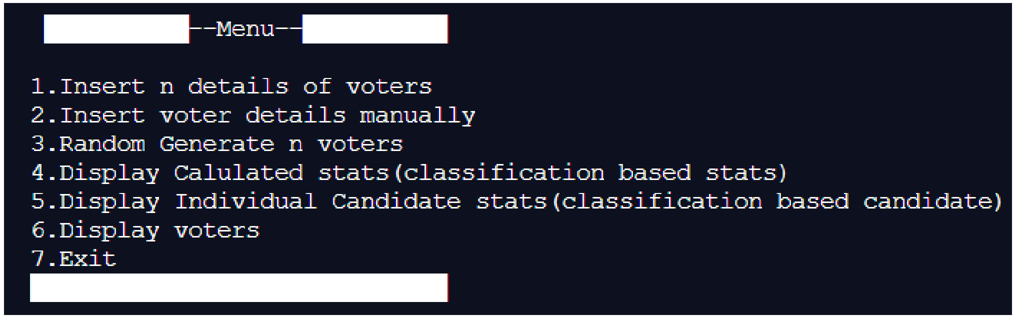
```
            --Menu--

1.Insert n details of voters
2.Insert voter details manually
3.Random Generate n voters
4.Display Calulated stats(classification based stats)
5.Display Individual Candidate stats(classification based candidate)
6.Display voters
7.Exit
```

- ➢ Output when the fifth(5.) option is selected.
- ➢ The stats and the winning candidate will be displayed.



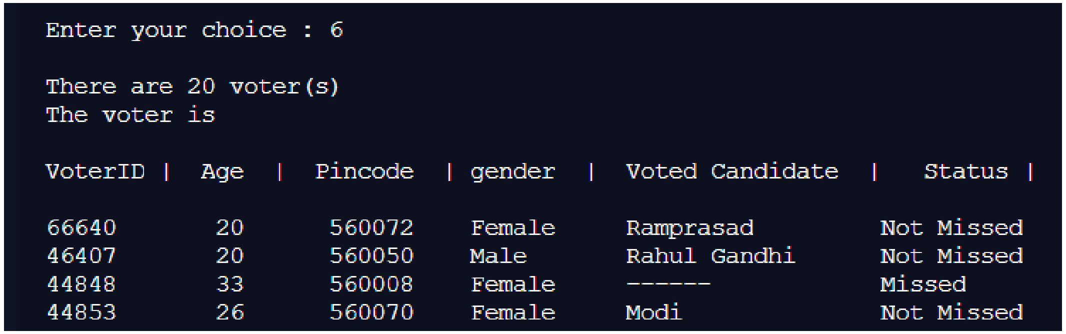
```
Enter your choice : 5

There are 21 voter(s)

Total Votes issued for candidates (votes - nota): 18

 Candidate 1 (Murthy)       got 2 votes
 Candidate 2 (Ramprasad)    got 4 votes
 Candidate 3 (GuruPrasad)   got 3 votes
 Candidate 4 (Modi)         got 1 votes
 Candidate 5 (Rahul Gandhi) got 6 votes

               ELECTION WINNER
                -Rahul Gandhi-
            with Largest Votes-6
```

# Results



```
                --Menu--

1.Insert n details of voters
2.Insert voter details manually
3.Random Generate n voters
4.Display Calulated stats(classification based stats)
5.Display Individual Candidate stats(classification based candidate)
6.Display voters
7.Exit
```

- Output when the sixth(6.) option is selected.
- The details of the randomly generated voters will be displayed(Here 20 voters were generated).



```
Enter your choice : 6

There are 20 voter(s)
The voter is

VoterID |  Age   |  Pincode  | gender   |  Voted Candidate   |   Status |

66640       20      560072     Female     Ramprasad            Not Missed
46407       20      560050     Male       Rahul Gandhi         Not Missed
44848       33      560008     Female     ------               Missed
44853       26      560070     Female     Modi                 Not Missed
```

# Conclusion & Future Enhancements

- In our upcoming days, as we learn more about data structures, we plan to implement them and enhance this project further more.
- We are also open to any type of suggestions/advises .
- Permanent data storage and also add more limits to data inputs.
- Adding a web interface for ease of use
- We have already done some implementations the teachers have suggested us

# Conclusion & Future Enhancements

- The project also now has feature when two candidates have same number of votes and displays which two candidates have the collision.

# References

- This project was built from scratch by us.
- We referred Narasimha Karumanchi's Data Structures and algorithms made easy textbook for extra knowledge on SLL.
- Visit our project in the given link:

https://repl.it/@NidPlays/DSA-Project-Work#main.c

# THANK YOU