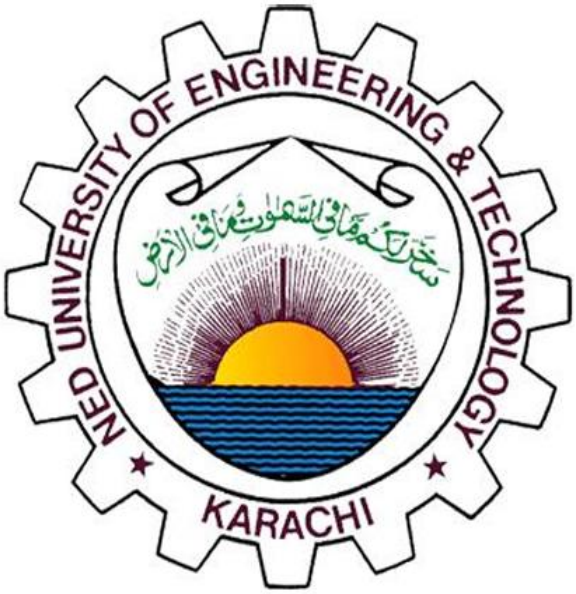
Voting System Project Report

Programming Fundamentals - CCP

Project Title: Voting System

Language Used: C



Submitted By:

1. Mariya Vayani (CT-061)
2. Rubaisha Arif (CT-067)
3. Ujala Usman (CT-073)

Submitted To:

Instructor: Mr. Abdullah

Submission Date: 10th November 2025

Index

1. Problem Solving Background

This project is designed to simulate a **voting system** using the C programming language. The objective is to design a program that allows users to:

- Vote for one of three candidates.
- Display live voting results.
- Identify ties or determine the winner automatically.
- Exit gracefully from the system.

This task was selected to practice and demonstrate key programming concepts such as:

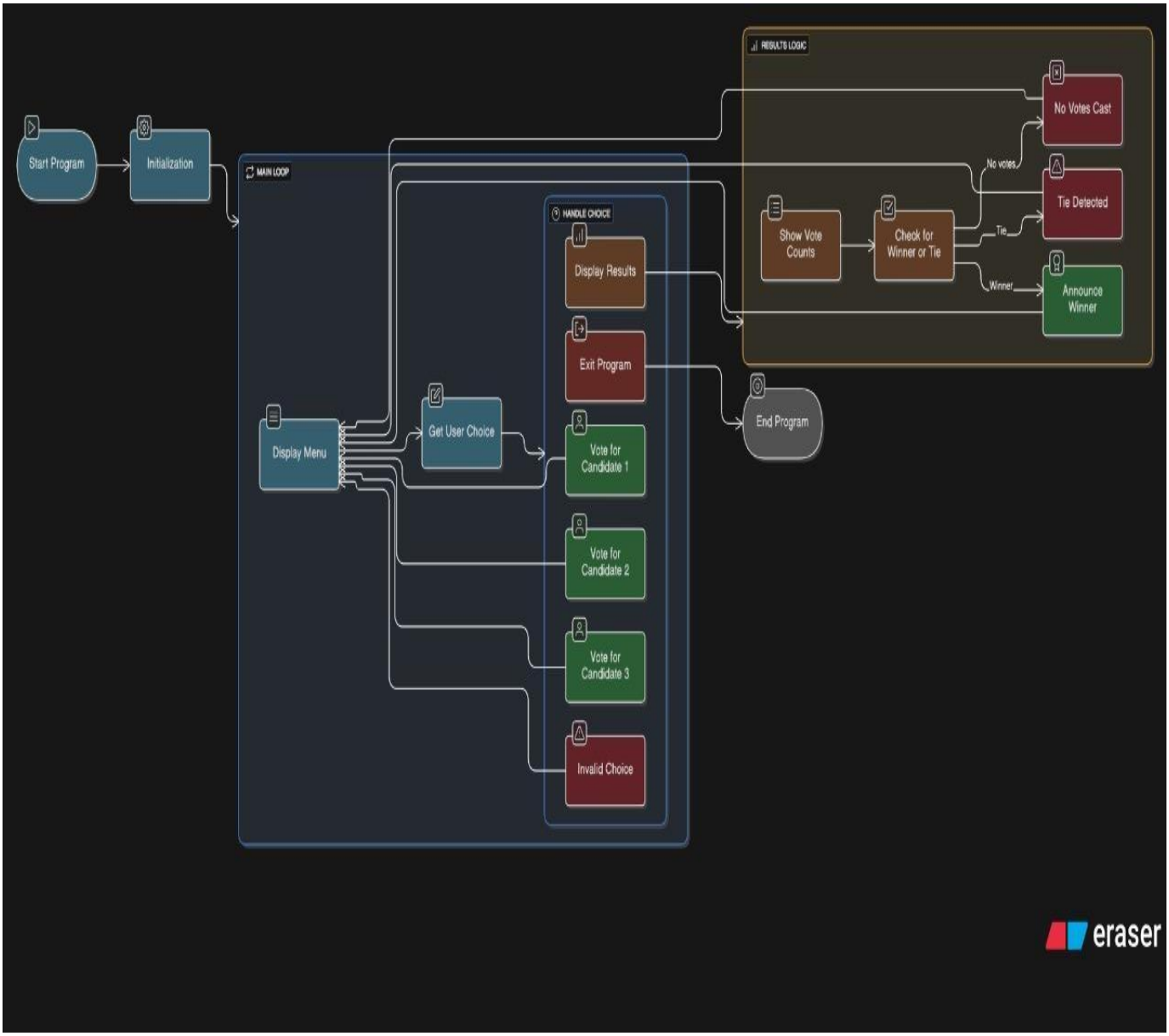
- Use of **loops** for repetitive tasks.
- **Arrays** for structured data storage.
- **Decision-making statements** for control flow.
- **Functions** for modular design.

By completing this project, students enhance their ability to structure logical code and understand program flow in C.

---

2. Flowchart

The following diagram illustrates the complete program logic, starting from initialization to result display.



Flow Summary:

1. The program starts and initializes candidate names and votes.
2. Displays the menu and prompts the user for input.
3. If the user selects 1–3, the vote is cast for the respective candidate.
4. If the user selects 4, the results are displayed.
5. If the user selects 5, the program exits.
6. Any invalid input results in an error message and re-display of the menu.
7. The loop continues until the user exits.

---

3. Explanation of Code

Global Declarations

```
#define num_candidate 3
#define candidate_name 40
char candidate[num_candidate][candidate_name] = {"Candidate 1", "Candidate 2", "Candidate 3"};
int votes[num_candidate] = {0, 0, 0};
```

Stores candidate names and initializes votes to zero.

Main Function

```
int main()
   { int
   choice; int
   r=1;
   while(r==1)
      { Menu();
      printf("enter choice : ");scanf("%d",&choice);
```

A while loop ensures that the program keeps running until the user exits.

Voting Logic

```
if(choice>=1 && choice<=3)
   { votes[choice-1]++;
   printf("%s : Vote has been
cast\n\n\n\n",candidate[choice-1]);
}
```

Handles valid choices (1–3) and increments the corresponding candidate’s vote count.

Display Results

```
else if(choice==4)
   { Results();
}
```

Calls the results function when the user chooses option 4.

Exit Option

```
else if(choice==5)
   { r=0;
   printf("\t\tEXIT\t\t\n");
}
```

Terminates the program and exits the voting loop.

Invalid Input

```
else {
   printf("Invalid choice!\n\n\n\n");
}
```

Displays an error message for out-of-range or invalid inputs.

Menu Function

```
void Menu() {
   printf("\t\tVOTING SYSTEM\t\t\n");
```

```
    printf("\t\tMENU\t\t\n");
    for(i=0;i<num_candidate;i++) {
        printf("[%d]: Vote for %s\n",i+1,candidate[i]);
    }
    printf("[%d]: Display Result\n",num_candidate + 1);
    printf("[%d]: Exit\n",num_candidate + 2);
}
```

Displays menu options and guides user interaction.

Results Function

```
void Results()
    { int max=-1;
    int win=-1;
    int tie=0;
    printf("\t\tVOTING RESULTS\t\t\n");
    for(i=0;i<num_candidate;i++) {
        printf("%s--Votes: %d\n",candidate[i],votes[i]);
```

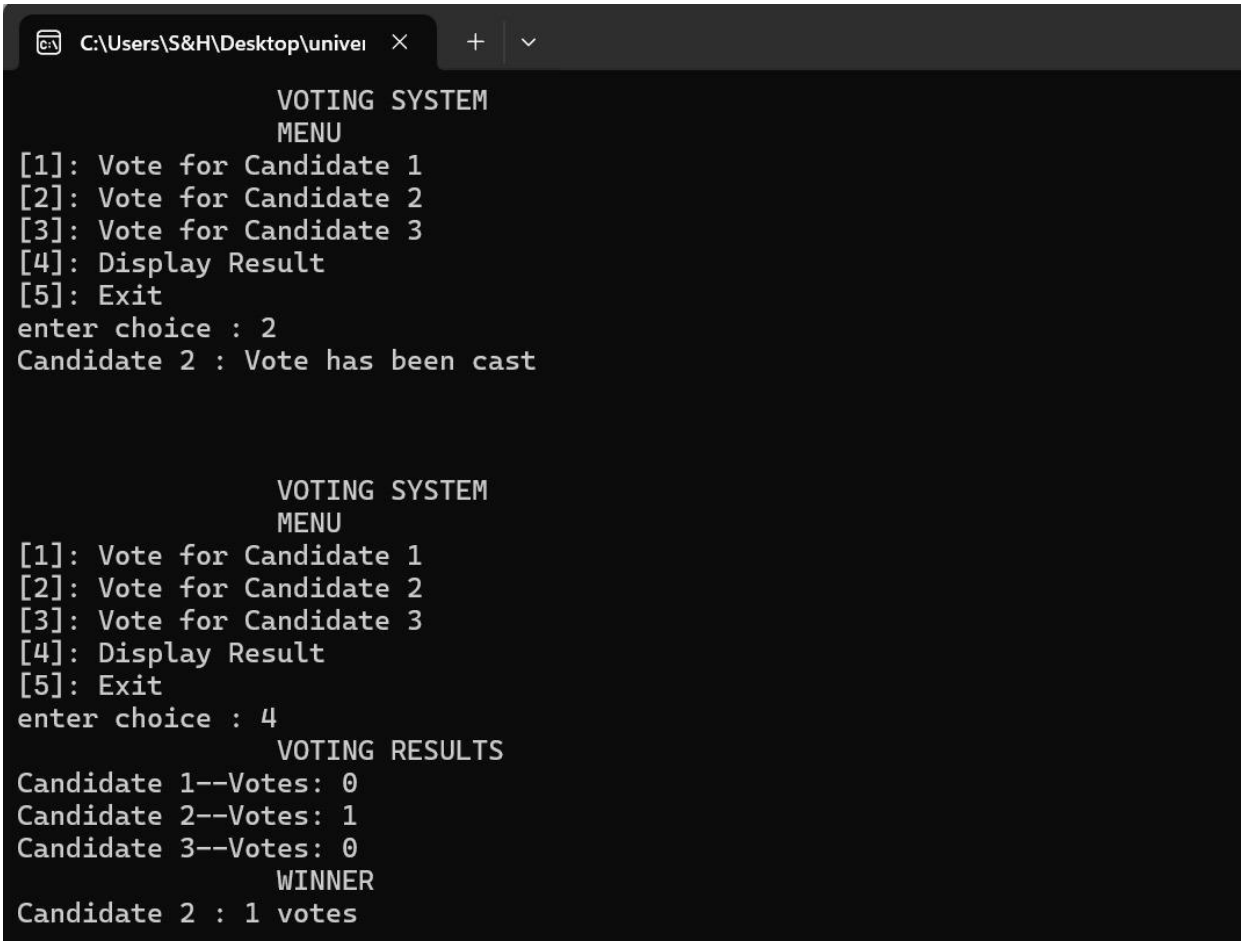
```
    if(votes[i]>max) {
        max=votes[i];
        win=i;
        tie=0;
    }
    else if(votes[i]==max && max>0)
        { tie=1;
    }
}
if(max==0) {
    printf("0 votes cast\n\n\n\n\n");
}
else if(tie==1) {
    printf("Its a TIE!\n\n\n\n\n");
}
else {
    printf("\t\tWINNER\t\t\n");
```

```
        printf("%s : %d votes\n\n\n\n",candidate[win],max);
    }
}
```

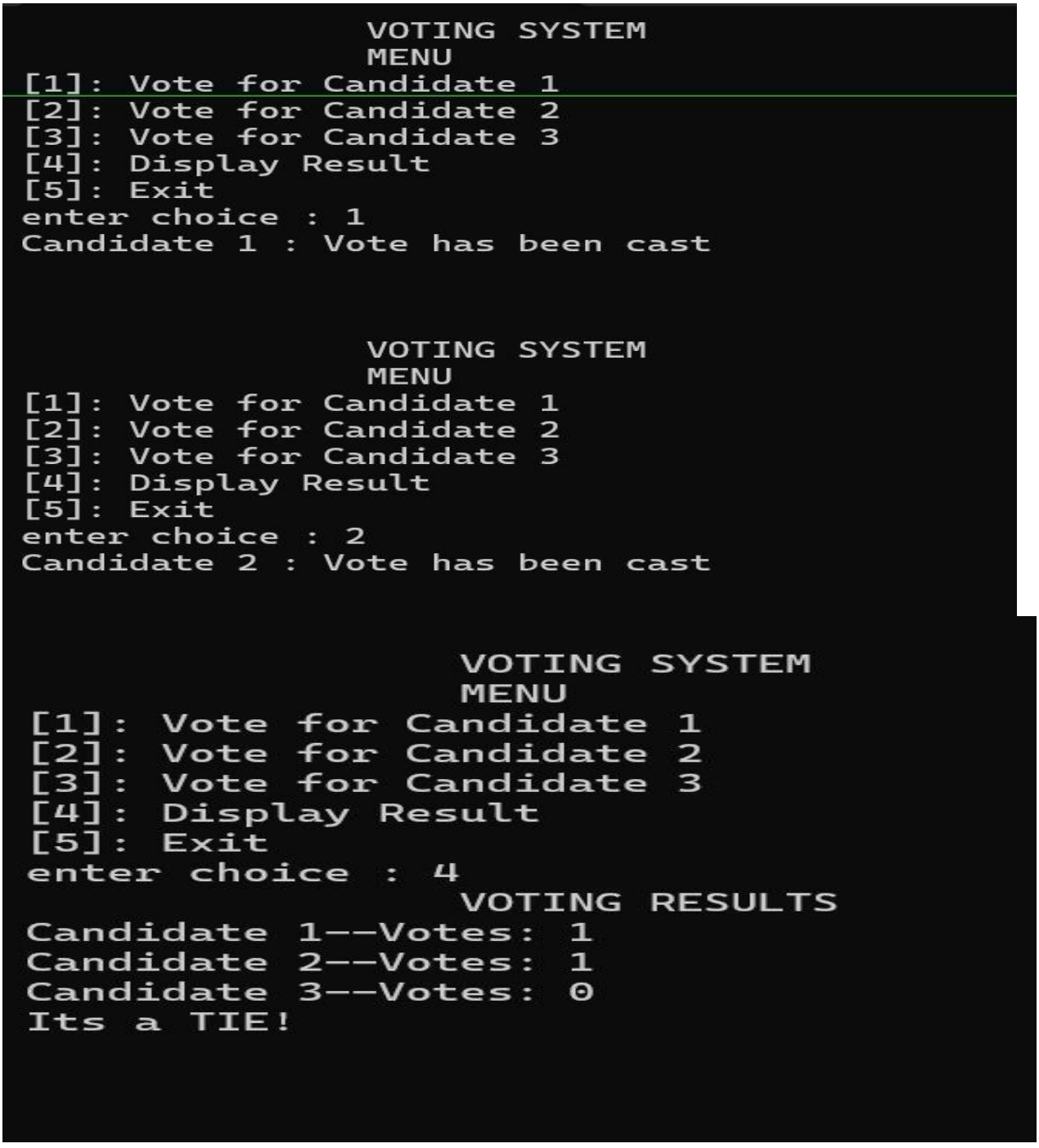
This function calculates total votes, detects ties, and displays the winner.

---

4. Sample Output



TIE SITUATION:



5. Conclusion

This **Voting System in C** project successfully demonstrates how fundamental programming logic can be combined to build an interactive console application. Key takeaways include:

- Practical understanding of **modular programming** through function-based design.
- Implementation of **decision-making constructs** using conditional statements.
- Efficient handling of repetitive tasks with loops.
- Real-time data management using arrays.

The project fulfills its purpose and sets a foundation for more advanced systems, such as file-based or web-integrated voting system.