Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Summer, Year:2022), B.Sc. in CSE (Day)

Course Title: Data Structure Lab
Course Code: CSE-106 Section:213-DB

Lab Project Name: CR VOTING SYSTEM

Details

Name	ID
Md Jabed Hossen	213902046

Submission Date: 09/ 10 / 2022
Course Teacher's Name: Farhana Akter Sunny

[For Teacher's use only: Don't Write Anything inside this box]

Lab Project Status

Marks: …… Signature: ..... Comments:

..... Date: .....

Table of Contents

Chapter 1

Introduction

1.1 Introduction

CR Voting System is a software solution for electing the CR. it is specifically designed for the students, and we tried to add its many features. This system helps the mentor and students elect CR

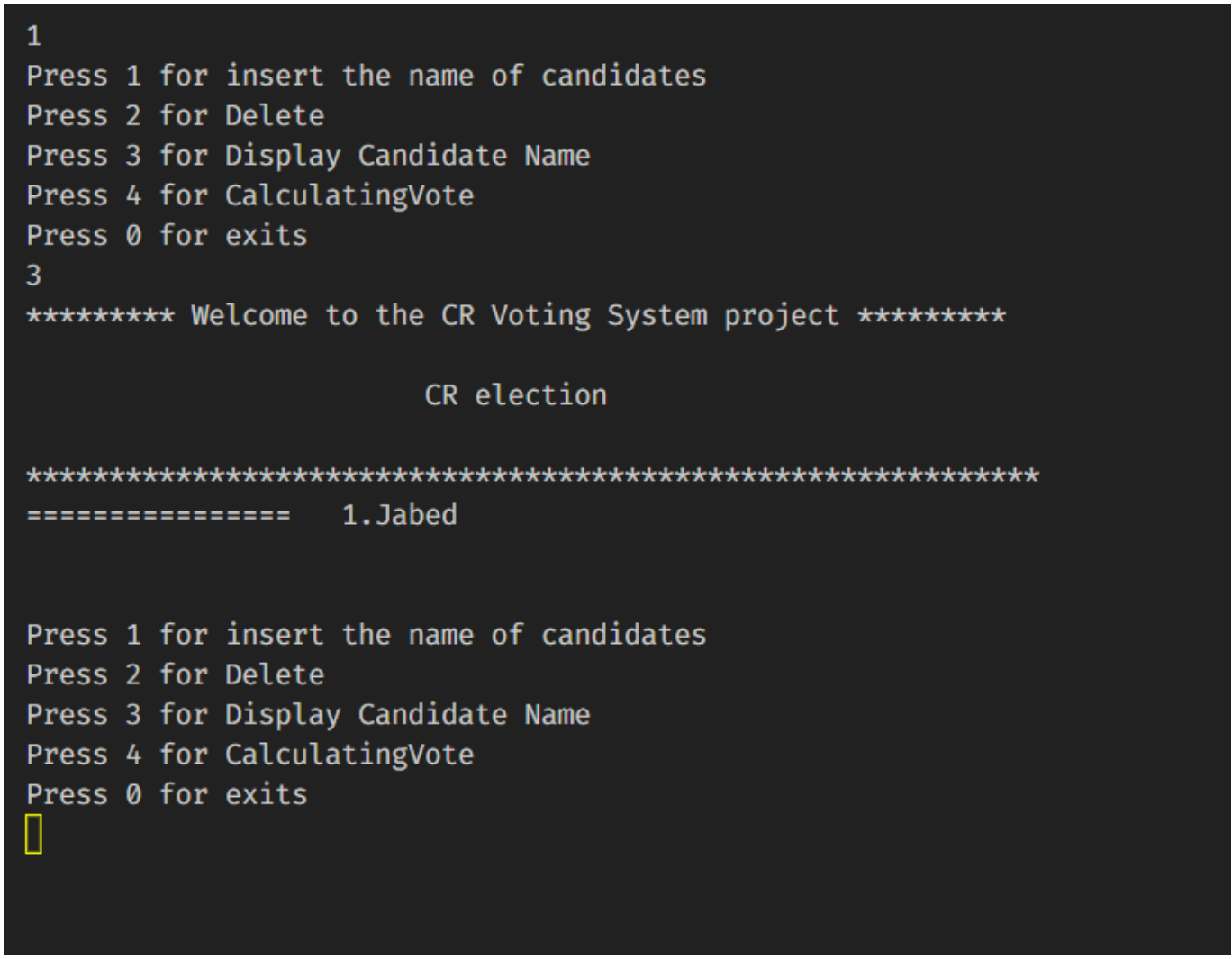
1.2 Design Goals/Objective

- ❖ The main focus of this project is to reduce time and lessen human efforts
- ❖ To provide a user-friendly environment where students can elect their favorite candidate

Chapter 2

CR Voting System I made using the C++ program using Linked List will look like this.

2.1 Interface



2.2 Implementation of the Project:

➢ **Start of the code:**

```
#include <bits/stdc++.h>
using namespace std;

/*
 all insertion function for candidate

*/
struct Node
{
    char name[30];
    int VoteCus=0;
    struct Node *next;
};

int totalVote=0;

struct Node *head;
```

➢ Main Function

```c
int main()
{
    int choice,target,position;
    char name[30];

    while(1)
    {
mainmenu :
        printf("Press 1 for insert the name of candidates \nPress 2 for Delete \nPress 3 for Display
Candidate Name \nPress 4 for CalculatingVote \nPress 0 for exits \n");
        scanf("%d",&choice);
        switch (choice)
        {
        case 1:
            printf("Insert candidate Name\n");
            fflush(stdin);
            fgets(name, sizeof(name), stdin);
            while(1)
            {
                printf("Press 1 for InsertFirst \nPress 2 for InsertEnd \nPress 3 for InsertAnyPosition
\n");
                scanf("%d",&target);
                switch(target)
                {
                case 1 :
                    insertFirst(name);
                    goto mainmenu;
                    break;

                case 2 :
                    insertEnd(name);
                    goto mainmenu;
                    break;

                case 3 :
                    printf("please select the position where you can insert the element\n");
                    scanf("%d",&position);
                    insertAnyPosition(name,position);
                    goto mainmenu;
                    break;
                default :
                    goto mainmenu;
                }
            }

        case 2:
            while(1)
            {
                printf("Press 1 for DeleteFirst \nPress 2 for DeleteEnd \n Press 3 for DeleteAnyPosition
\n");
                scanf("%d",&target);
                switch(target)
                {
                case 1 :
                    delete_First();
                    goto mainmenu;
                    break;

                case 2 :
                    delete_End();
                    goto mainmenu;
                    break;

                case 3 :
                    printf("Please select the position where you can delete the Elements the element\n");
                    scanf("%d",&position);
                    deleteAtAnyPosition(position);
                    goto mainmenu;
                    break;
                default :
                    goto mainmenu;
                }
            }
```

```
case 3 :
            display();
            break;
        case 4 :
            while(1)
            {
secondMenu:
                printf("Press 1 for vote \nPress 2 for view Result \nPress 3 for individual result
\nPress 4 for winner \nPress 5 for main menu\n");
                scanf("%d",&target);
                if(target==1)
                {
                    display();
                    printf("whom do you want to vote ?\n");
                    scanf("%d",&position);
                    calculatingVote(position);
                    goto secondMenu;
                }
                else if (target==2)
                {
                    ResultDisplay();
                    goto secondMenu;
                }
                else if (target==3)
                {
                    individualResult();
                    goto secondMenu;
                }
                else if (target==4)
                {
                    winner();
                    goto secondMenu;
                }
                else
                {
                    goto mainmenu;
                    break;
                }
            }

        case 0 :
            exit(0);
        }
    }

}
```

➢ **Insert All Function ()**

This function is used to insert the candidate name

```
void insertFirst(char data[30])
{
    struct Node *newNode;
    newNode =(struct Node*) malloc(sizeof(struct Node));
    strcpy(newNode->name,data);
    if(head==NULL)
    {
        newNode->next=NULL;
        head=newNode;
    }
    else
    {
        newNode->next=head;
        head=newNode;
    }
}
//insert end
void insertEnd(char data[30])
{
    struct Node *newNode;
    newNode =(struct Node*) malloc(sizeof(struct Node));
    strcpy(newNode->name,data);
    newNode->next=NULL;
    if(head==NULL)
    {
        head=newNode;
    }
    else
    {
        struct Node *temp=head;
        while(temp->next!=NULL)
        {
            temp= temp->next;
        }
        temp->next=newNode;
    }
}
```

```c
//insert any position
void insertAnyPosition(char data[30],int position)
{
    struct Node *newNode;
    newNode=(struct Node*) malloc(sizeof(struct Node));
    strcpy(newNode->name,data);
    if(position==1)
    {
        newNode->next=head;
        head=newNode;
        return ;
    }
    else
    {
        struct Node *temp=head;
        for(int i=0; i<position-2; i++)
        {
            temp=temp->next;
        }
        newNode->next=temp->next;
        temp->next=newNode;
    }
}
```

➢Delete all Function( )

This function is used to delete candidate

```
//delete first
void delete_First ()
{
    head = head->next;
}
//delete end
void delete_End ()
{
    struct Node* temp = head;
    while(temp->next->next!=NULL)
    {
        temp = temp->next;
    }
    temp->next = NULL;
}
```

```c
//delete at any postion
void deleteAtAnyPosition(int position)
{
    struct Node* temp = head;
    if(position==1)
    {
        head=temp->next->next;
        return ;
    }
    else
    {
        for(int i=2; i< position; i++)
        {
            if(temp->next!=NULL)
            {
                temp = temp->next;
            }
        }
        temp->next = temp->next->next;
    }
}
```

➢Calculate Function( )

This function is used to calculate the vote

```
void calculatingVote(int position)
{
    struct Node *temp=head;
    for(int i=1; i< position; i++)
    {
        temp = temp->next;
    }
    totalVote++;
    temp->VoteCus++;
}
```

➢Result Display Function( )

This function is used to view all the votes for every candidate

```
void ResultDisplay()
{
    struct Node *i,*j;
    int tempData;
    char name[100];

    for(i=head; i->next!=NULL; i=i->next)
    {
        for(j=i->next; j!=NULL; j=j->next)
        {
            if(i->VoteCus<j->VoteCus)
            {
                tempData=i->VoteCus;
                i->VoteCus=j->VoteCus;
                j->VoteCus=tempData;
                strcpy(name,i->name);
                strcpy(i->name,j->name);
                strcpy(j->name,name);
            }
        }
    }
    if(head==NULL)
    {
        printf("empty");
    }
    else
    {
        int count=1;
        printf("*********         Cr Voting Result    *********\n\n");
        printf("                     CR election                        \n\n");
        printf("************************************************************\n");

        struct Node *temp=head;
        while (temp->next!=NULL)
        {
            printf(" %d.%s  == Total vote ================ %d\n",count,temp->name,temp->VoteCus);
            temp=temp->next;
            count++;
        }
        printf(" %d.%s  == Total vote ================ %d\n",count,temp->name,temp->VoteCus);

    }
}
```

➢Result Individual Function( )

This function is used to view individual Vote

```
void individualResult()
{
    printf("Welcome to Individual Result\n");
    char data[30];
    printf("search the name\n");
    fflush(stdin);
    fgets(data,sizeof(data),stdin);
    struct Node *temp=head;
    while(temp->next!=NULL)
    {
        if(strcmp(temp->name, data) == 0)
        {
            printf("Candidate is found\n");
            printf("Candidate Name is    %s\n",temp->name);
            printf("Candidate Total Vote %d\n",temp->VoteCus);
            return;
        }
        temp=temp->next;
    }
    //last node check
    if(strcmp(temp->name, data) == 0)
    {
        printf("Candidate is found\n");
        printf("Candidate Name is    %s\n",temp->name);
        printf("Candidate Total Vote %d\n",temp->VoteCus);
    }
    else
    {
        printf("Candidate is not found");
    }
}
```

➢Winner Function( )

This function is used to show who is the winner

```
void winner()
{

    struct Node * temp=head;

    int max=-1;

    while(temp!=NULL)
    {
        if(max<temp->VoteCus)
        {
            max=temp->VoteCus;
        }
        temp=temp->next;
    }

    temp=head;
    while(temp!=NULL)
    {
        if(temp->VoteCus==max)
        {
            int per=(temp->VoteCus*100)/totalVote;
            printf("==========Congratulations  %s============\n",temp->name);
            printf("The CR Winner name is %s\n",temp->name);
            printf("The Winner Total Vote %d\n",temp->VoteCus);
            printf("The Winner percentage vote %d%c \n",per,37);
            printf("Thank you all for the participating ");
            printf("=======================================");
            exit(0);
        }
        temp=temp->next;
    }
}
```

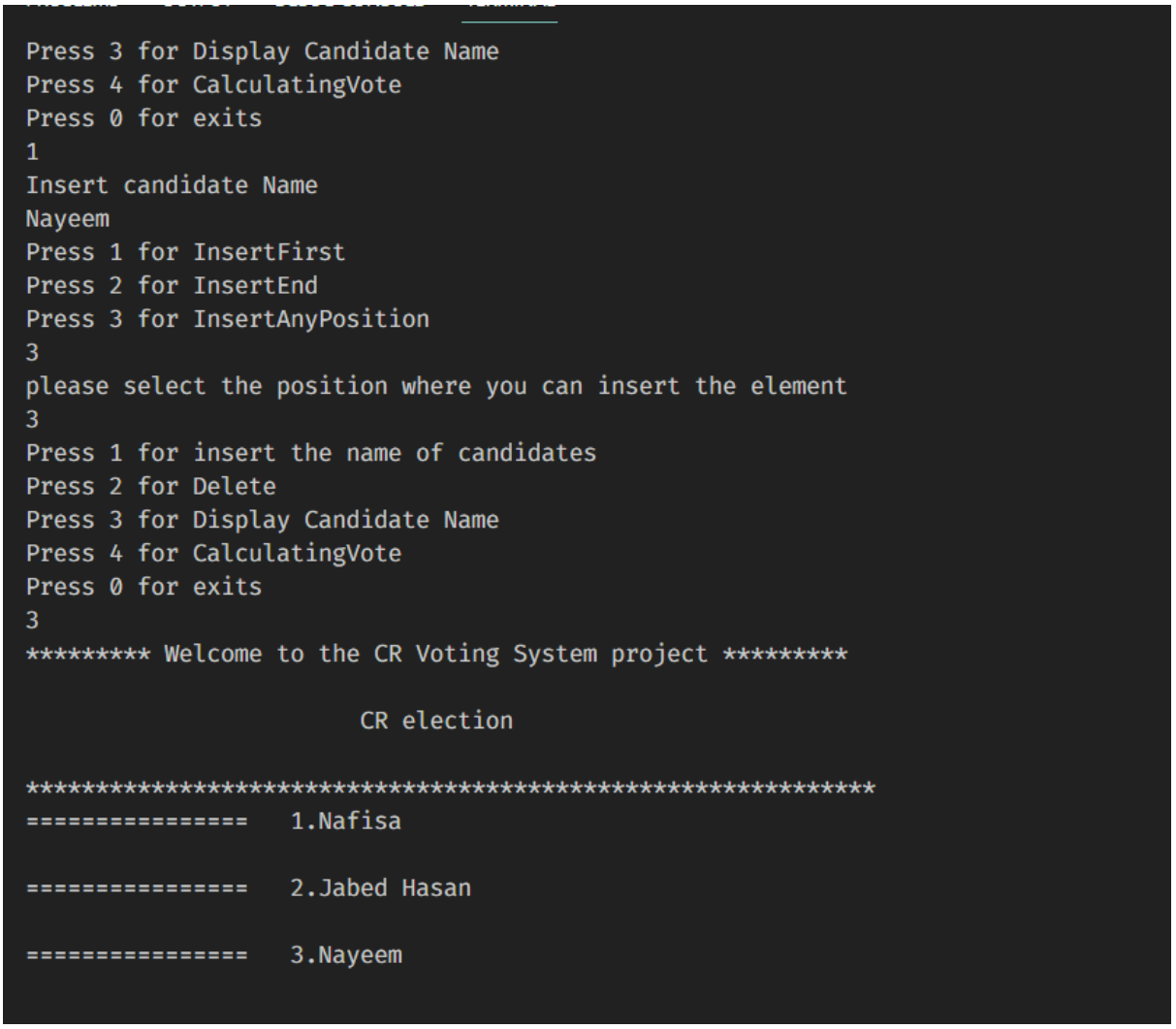
Chapter 3

Performance Evaluation

3.1 Results and Discussions

3.1.1 Output

❖ This is the main panel of the system. From there can do any of the following options.



❖ Any user enters a vote for their valuable candidate

```
********* Welcome to the CR Voting System project *********

                        CR election

************************************************************
================   1.Nafisa

================   2.Jabed Hasan

================   3.Nayeem
```

13 ❖A user press the number that he/she wants

```
1
Press 1 for vote
Press 2 for view Result
Press 3 for individual result
Press 4 for winner
Press 5 for main menu
1
```

❖If a user press 2 then the user can view all the candidate's results

```
whom do you want to vote ?
3
Press 1 for vote
Press 2 for view Result
Press 3 for individual result
Press 4 for winner
Press 5 for main menu
2
*********         Cr Voting Result    *********

                    CR election

************************************************************
 1.Nayeem
  == Total vote ================ 2
 2.Nafisa
  == Total vote ================ 1
 3.Jabed Hasan
  == Total vote ================ 0
Press 1 for vote
```

❖If a user press 4 then the user can view who is the winner

```
Press 1 for vote
Press 2 for view Result
Press 3 for individual result
Press 4 for winner
Press 5 for main menu
4
==========Congratulations  Nayeem
============
The CR Winner name is Nayeem
```

3.1.2 Analysis and Outcome

The project is built using a C++ programming language using linked list. We do the coding on Visual Studio Code using GCC compiler. This project is mainly built to reduce the pressure and do the work efficiently. We will update this project and add more features. It will be helpful for all the students and the teachers. So far we do the project using the course knowledge of Linked list insertion and delete insertion sorting, and Linear search data structure

Chapter 4

Conclusion

4.1 Introduction

The CR Voting System needs to be computerized to reduce human errors and increase efficiency. By computerizing the system we can do the work lesser errors. This project is built for electing CR.

4.1 Practical Implications

The CR Voting system helps the user to easily know who is the winner

4.2 Scope of Future Work

In the future, this can be the most useful product in school, college, and university. In the future, we can add more features to this. Like, such as adding student ID, Sections, etc. Without student's ID, it will not be possible to vote for the candidates. We will do it dynamically . This system can reduce mistakes and work more efficiently. In this way, it can be helpful for our work.