**Source code:**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<malloc.h>

struct node{

char party\_name[20],candidate\_name[20],district\_name[20];

int vote\_count;

struct node\* next;

};

void add\_nominee(struct node\* header,char partyname[20]){

char k[20];

struct node\* n=(struct node\*)malloc(sizeof(struct node));

printf("Enter candidate name for %s district from party %s:",header->district\_name,partyname);

scanf("%s",&k);

strcpy(n->district\_name,header->district\_name);

strcpy(n->party\_name,partyname);

strcpy(n->candidate\_name,k);

n->vote\_count=0;

n->next=NULL;

if(header->next==NULL){

header->next=n;

}

else{

struct node\* temp=header;

while(temp->next!=NULL){

temp=temp->next;

}

temp->next=n;

temp=temp->next;

temp->next=NULL;

}

printf("%s from %s successfully added to the list!!!\n",n->candidate\_name,n->party\_name);

}

void del(struct node\* header){

char n;

printf("Enter nominee no. to delete=");

scanf("%d",&n);

int i=0;

struct node \*temp=header,\*temp2=NULL;

while(i!=n){

temp2=temp;

temp=temp->next;

i++;

}

if(temp==NULL){

printf("WRONG CHOICE!!!\n");

return;

}

if(temp->next==NULL)

temp2->next=NULL;

else

temp2->next=temp->next;

printf("%s from %s is removed from list!!!\n",temp->candidate\_name,temp->party\_name);

getch();

}

void display(struct node\* header){

struct node\* temp=header;

if(temp->next==NULL){

printf("Empty List!!!");

}

temp=temp->next;

int i=1;

while(temp!=NULL){

printf("%d. %s(%s)\n",i,temp->party\_name,temp->candidate\_name);

temp=temp->next;

i++;

}

}

void take\_vote(struct node\* header){

display(header);

int v;

printf("Enter your vote=");

scanf("%d",&v);

int i=0;

struct node\*temp=header;

while(i!=v){

temp=temp->next;

i++;

}

if(temp==NULL){

printf("WRONG CHOICE!!!\n");

return;

}

temp->vote\_count++;

if(header->vote\_count<temp->vote\_count){

header->vote\_count=temp->vote\_count;

strcpy(header->party\_name,temp->party\_name);

strcpy(header->candidate\_name,temp->candidate\_name);

}

}

void reset\_votes(struct node\* header){

struct node\* temp=header;

if(header==NULL)

return;

header->vote\_count=-1;

temp=temp->next;

while(temp!=NULL){

temp->vote\_count=0;

temp=temp->next;

}

}

int main() {

system("cls");

int n,p,i=0,c=0;

char city\_name[20];

printf("Enter city name:");

scanf("%s",&city\_name);

printf("Enter total number of standing parties:");

scanf("%d",&p);

char party\_names[p][20];

for(i=0;i<p;i++){

printf("Enter name of party %d=",i+1);

scanf("%s",&party\_names[i]);

}

system("cls");

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

printf(" %s Voting System\n",&city\_name);

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

printf("Enter details of districts\n");

printf("Enter total number of districs:");

scanf("%d",&n);

struct node \*header[n];

for(i=0;i<n;i++){

char k[20]="";

struct node\* temp=(struct node\*)malloc(sizeof(struct node));

temp->vote\_count=-1;

strcpy(temp->party\_name,k);

strcpy(temp->candidate\_name,k);

printf("Enter name of District %d=",i+1);

scanf("%s",&k);

strcpy(temp->district\_name,k);

temp->next=NULL;

header[i]=temp;

for(int j=0;j<p;j++){

add\_nominee(header[i],party\_names[j]);

}

}

printf("\nCandidate details entered successfully!!!");

getch();

while(i!=1){

system("cls");

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

printf(" %s Voting System\n",&city\_name);

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

printf("1.Display\n");

printf("2.Change candidate\n");

printf("3.Start voting\n");

printf("4.Reset\n");

printf("5.Show Results\n");

printf("6.Exit\n");

printf("Enter choice:");

scanf("%d",&c);

switch(c){

case 1:{

for(int j=0;j<n;j++){

printf("District:%s\n",header[j]->district\_name);

display(header[j]);

}

getch();

break;

}

case 2:{

int k1,k2;

for(int j=0;j<n;j++){

printf("%d. %s\n",j+1,header[j]->district\_name);

}

printf("Enter name of candidate's district number:");

scanf("%d",&k1);

display(header[k1]);

printf("Enter candidat number:");

scanf("%d",&k2);

struct node \*temp=header[k1];

while(k2!=0){

if(temp==NULL)

break;

temp=temp->next;

k2--;

}

if(temp==NULL)

printf("WRONG CHOICE!!!");

else{

char name[20];

printf("Enter name of new candidate:");

scanf("%s",&name);

strcpy(temp->candidate\_name,name);

printf("Candidate changed successfully!!!");

}

getch();

break;

}

case 3:{

system("cls");

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

printf(" %s Voting System\n",&city\_name);

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

printf("Choose district to start voting from\n");

int k=0;

for(int j=0;j<n;j++)

printf("%d. %s\n",j+1,header[j]->district\_name);

printf("Enter District number:");

scanf("%d",&k);

k--;

struct node\* temp=header[k];

if(temp->vote\_count!=-1){

printf("Voting for this district is complete!!!\n");

getch();

break;

}

int j=0;

system("cls");

while(j==0){

system("cls");

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

printf(" %s Voting System\n",&city\_name);

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

printf("Voting for %s district\n",temp->district\_name);

take\_vote(temp);

char t[1];

printf("Enter new vote(y/n):");

scanf("%s",&t);

if(strcmp(t,"n")==0){

break;

}

}

printf("\n!!!Voting completed for %s district!!!",temp->district\_name);

break;

}

case 4:{

for(int j=0;j<n;j++){

reset\_votes(header[j]);

}

break;

}

case 5:{

int votes[p],max=0,id=0;

for(int j=0;j<p;j++)

votes[j]=0;

for(int j=0;j<n;j++){

if(header[j]->vote\_count!=-1){

printf("%s from %s has won from %s with %d votes\n",header[j]->candidate\_name,header[j]->party\_name,header[j]->district\_name,header[j]->vote\_count);

for(int k=0;k<p;k++){

if(strcmp(header[j]->party\_name,party\_names[k])==0){

votes[k]+=header[j]->vote\_count;

if(max<votes[k]){

max=votes[k];

id=k;

}

break;

}

}

}

else

printf("Voting for district %s has not been done.\n",header[j]->district\_name);

}

printf("\n%s has won the %s elections!!!",party\_names[id],&city\_name);

getch();

break;

}

case 6:{

printf("Quiting!!!");

i=1;

break;

}

default :{

printf("Wrong Choice");

break;

}

}

}

return 0;

}