PHONEBOOK DIRECTORY

Member 1 Detail:-

Name:- Abhishek Anand

Section & Roll:- A-57

Enrollment No.- 12019002004061

Email ID:-abhianand2308@gmail.com

Member 2 Detail:-

Name:- Jay Prakash Panday

Section & Roll:- A-44

Enrollment No.- 12019002004044

Email ID:- jayprakashpandeyallen@gmail.com

Member 3 Detail:-

Name:- Saurav Kumar

Section & Roll:- A-48

Enrollment No.- 12019002004051

Email ID:- sauravkiul811310@gmail.com

Member 4 Detail:-

Name:- Saurav Anand

Section & Roll:- A-59

Enrollment No.- 12019002004064

Email ID:- sauravaanand700@gmail.com

PHONEBOOK DIRECTORY

Objective:-

We are determined to design a system which is intended in saving name and number of desired persons. The main objective of telephone directory is to add, search, edit and delete various contacts.

Introduction:-

A phonebook directory (also called a telephone book) is a listing of telephone subscribers in a geographical area or subscribers to services provided by the organization that publishes the directory. It consists of the name as well as the contact number of people added as contact in the directory. Name and contact number are displayed in alphabetical order.

Description:-

It is a friendly easy to use interface developed with data structure as the backend to store the details. This application stores all the details like name, telephone number and email id in a database. This system is developed to reduce the errors that creep up in manual systems. This software also allows to edit, update and search various contact details. It is secure, easy to use and reliable software system. It also provides a good level of security as there is an admin who can only edit and update details.

- Admin module: This is the main module where the administrator logins into the database. All the admin details are stored with user id.
- Create Contact module: This module allows the user to enter all the details like name, phone number and email id.
- Insert module: In this module, all the details regarding contacts are inserted.
- Delete module: In this module, the records with a specific id is deleted.
- Display module: This module allows the user to view all contact details.
- Search module: This module allows to search for the contact details by just entering its name.

Code associated with this project:-

```
#include <iostream>
#include<cstring>
using namespace std;
class dnode
{
  public:
    char number[50];
    char gmail[40];
    char name[30];
    dnode *prev,*next;
    dnode(char n[],char r[],char g[])
```

```
{
            strcpy(name,n);
             strcpy(number,r);
            strcpy(gmail,g);
             next=NULL;
             prev=NULL;
}
          friend class dlist;
};
class dlist
{
   dnode *head,*temp,*ptr;

   dnode *ptr1, *ptr2, *dup;
   public:
   dnode *prevn;

   void insert();
   void sort();
   void deletecontact(char n[20]);
   void deletesamenumber();
   void deletesamename();
   void deletesamegmail();
   void searchbyname(char p[20]);
   void searchbynumber(char no[30]);
   void searchbygmail(char g[20]);

      void accept();
      void display();
```

```
        void update(char ch[10]);
        dlist()
                {
                        head=NULL;
                        temp=NULL;
                        ptr=NULL;
                        ptr1=NULL;
                        ptr2=NULL;
                        dup=NULL;
                }
};
void dlist::accept()
{
        char number[50];
        char gmail[40];
        char name[30];
    char ans;
      do
        {
    cout<<"ENTER NAME      :";
    cin>>name;
    cout<<"ENTER NUMBER    :";
    cin>>number;
    while(strlen(number)!=10)
    {
    cout<<"ENTER VALID NUMBER  :";
    cin>>number;
    }
    cout<<"ENTER G-MAIL    :";
```

```
  cin>>gmail;
  temp=new dnode(name,number,gmail);
  if(head==NULL)
  {
    head=temp;
  }
  else
  {
   ptr=head;
             while(ptr->next!=NULL)
             {
ptr=ptr->next;
             }
             ptr->next=temp;
             temp->prev=ptr;
  }
        cout<<"DO YOU WANT TO CONTINUE?????????";
  cin>>ans;
}while(ans=='y');

}
void dlist::display()
{
  ptr=head;
      while(ptr!=NULL)
      {
                   cout<<"\n\nNAME  ::\t"<<ptr->name;
                   cout<<"\nNUMBER::\t+91-"<<ptr->number;
                   cout<<"\nG-MAIL::\t"<<ptr->gmail;
```

```
                    ptr=ptr->next;
       }
}

  void dlist::insert()
  {
    accept();
  }
  void dlist::sort()
  {
    dnode *i,*j;
    int temp;
    char n[10];
    for(i=head;i->next!=NULL;i=i->next)
    {
      for(j=i->next;j!=NULL;j=j->next)
      {
        temp=strcmp(i->name,j->name);
        if(temp>0)
        {
          strcpy(n,i->name);
          strcpy(i->name,j->name);
          strcpy(j->name,n);
        }
      }
    }

  }
  void dlist::deletecontact(char s[20])
```

```
{
    int c=0;
    ptr=head;
    while(ptr!=NULL)
    {
        if(strcmp(s,ptr->name)==0)
        {
            c=1;
            break;
        }
        else
        {
            c=2;
        }
        ptr=ptr->next;
    }
    if(c==1 && ptr!=head && ptr->next!=NULL)
    {
        ptr->prev->next=ptr->next;
       ptr->next->prev=ptr->prev;
       delete(ptr);
       cout<<"YOUR CONTACT IS SUCCESSFULLY DELETED\n\n";
    }
    if(ptr==head)
    {
        head=head->next;
        head->prev=NULL;
        delete(ptr);
       cout<<"YOUR CONTACT IS SUCCESSFULLY DELETED\n\n";
    }
    if(ptr->next==NULL)
    {
        ptr->prev->next=NULL;
        ptr->prev=NULL;
        delete(ptr);
       cout<<"YOUR CONTACT IS SUCCESSFULLY DELETED\n\n";
    }
    if(c==2)
    {
        cout<<"YOUR ENTERED NAME IS NOT IN THE LIST...";
    }
}
void dlist::deletesamename()
{
    ptr1=head;
    while (ptr1 != NULL && ptr1->next != NULL)
{
    ptr2 = ptr1;
    while (ptr2->next != NULL)
    {
        if (strcmp(ptr1->name,ptr2->next->name)==0)
        {
            dup = ptr2->next;
            ptr2->next = ptr2->next->next;
            delete(dup);
        }
        else
        {
```

```
      ptr2 = ptr2->next;
  }
  }
  ptr1 = ptr1->next;
}
}
void dlist::deletesamegmail()
{
  ptr1=head;
  while (ptr1 != NULL && ptr1->next != NULL)
{
  ptr2 = ptr1;
  while (ptr2->next != NULL)
  {
    if (strcmp(ptr1->gmail,ptr2->next->gmail)==0)
    {
      dup = ptr2->next;
      ptr2->next = ptr2->next->next;
      delete(dup);
    }
    else
    {
      ptr2 = ptr2->next;
  }
  }
  ptr1 = ptr1->next;
}
}
void dlist::deletesamenumber()
```

```
{
 ptr1=head;
    while (ptr1 != NULL && ptr1->next != NULL)
{
    ptr2 = ptr1;
    while (ptr2->next != NULL)
    {
        if (strcmp(ptr1->number,ptr2->number)==0)
        {
            dup = ptr2->next;
            ptr2->next = ptr2->next->next;
            delete(dup);
        }
        else
        {
            ptr2 = ptr2->next;
    }
    }
    ptr1 = ptr1->next;
}
}
void dlist::searchbyname(char na[10])
{
    ptr=head;
    while(ptr!=NULL)
    {
        if(strcmp(na,ptr->name)==0)
        {
            cout<<"NAME FOUND"<<endl;
            cout<<"CONTACT DETAILS ARE BELOW:\n"<<endl;
         cout<<"\n\nNAME  ::\t"<<ptr->name;
                                    cout<<"\nNUMBER::\t+91-"<<ptr->number;
                                    cout<<"\nG-MAIL::\t"<<ptr->gmail;


        }
        ptr=ptr->next;
    }
}
void dlist::searchbynumber(char num[20])
{
 ptr=head;
    while(ptr!=NULL)
    {
        if(strcmp(num,ptr->number)==0)
        {
            cout<<"NUMBER FOUND\n"<<endl;
            cout<<"CONTACT DETAILS ARE BELOW:\n"<<endl;
         cout<<"\n\nNAME  ::\t"<<ptr->name;
                                    cout<<"\nNUMBER::\t+91-"<<ptr->number;
                                    cout<<"\nG-MAIL::\t"<<ptr->gmail;


        }
        ptr=ptr->next;
    }
}
void dlist::searchbygmail(char gm[20])
{
    ptr=head;
    while(ptr!=NULL)
    {
        if(strcmp(gm,ptr->gmail)==0)
        {
            cout<<"G-MAIL FOUND\n"<<endl;
            cout<<"CONTACT DETAILS ARE BELOW:\n"<<endl;
         cout<<"\n\nNAME  ::\t"<<ptr->name;
                                    cout<<"\nNUMBER::\t+91-"<<ptr->number;
                                    cout<<"\nG-MAIL::\t"<<ptr->gmail;


        }
        ptr=ptr->next;
    }
}
void dlist::update(char n[20])
{
    char ans;
    int c;
    ptr=head;
    while(ptr!=NULL)
    {
        if(strcmp(n,ptr->name)==0)
        {

            do
            {
                cout<<"\nWHAT DO YOU WANT TO UPDATE?\n1.NAME\n2.PHONE NUMBER\n3.G-MAIL\n";
            cin>>c;
            switch(c)
```

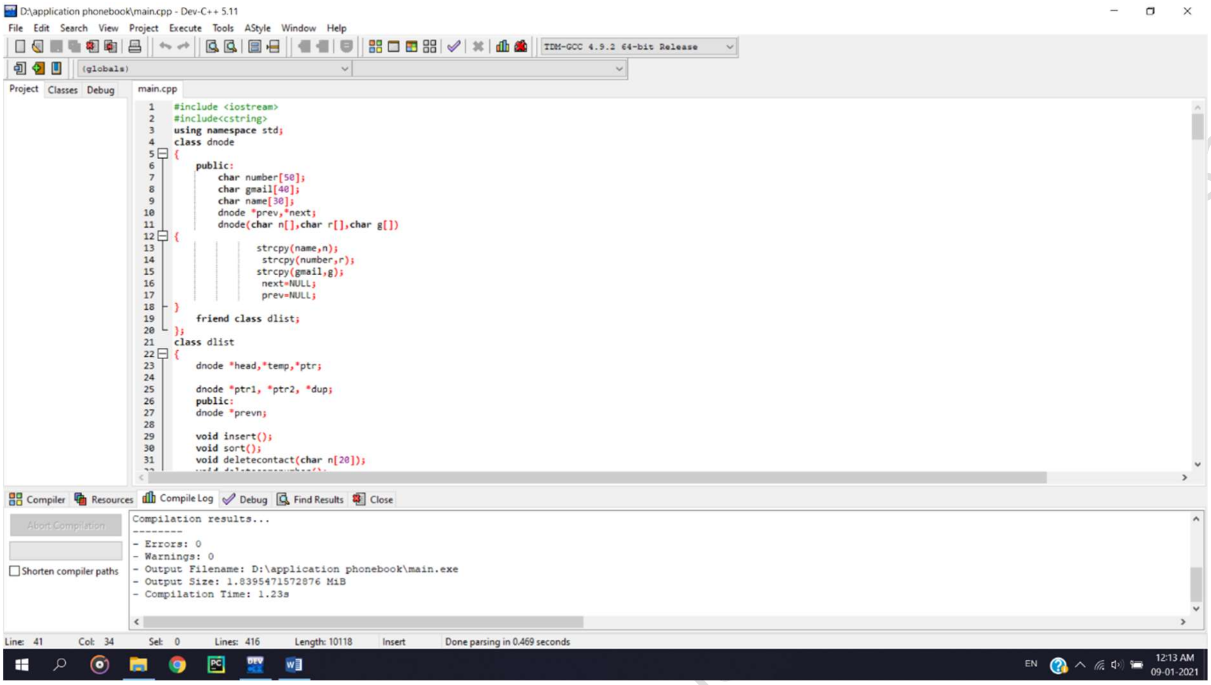
```
              {
                case 1:
                cout<<"ENTER NEW-NAME=";
                cin>>ptr->name;
                break;
                case 2:
                cout<<"ENTER NEW PHONE-NUMBER?";
                cin>>ptr->number;
                while(strlen(ptr->number)!=10)
    {
    cout<<"ENTER VALID NUMBER  :";
    cin>>ptr->number;
    }
                break;
                case 3:
                cout<<"ENTER NEW G-MAIL";
                cin>>ptr->gmail;
                break;
              }
              cout<<"DO YOU WANT TO CONTINUE UPDATING?";
              cin>>ans;
              }while(ans=='y');
            }
            ptr=ptr->next;
      }
    }
  int main()
  {
    char n[20];
```

```
   char nam[20];

   char name[10];

   char number[10];

   char gmail[20];

   dlist d1;

   char ans;

   int ch,a;

   cout<<"***************           PHONE BOOK powered by ABHISHEK ANAND,
JAYPRAKASH,  SAURAV KUMAR AND SAURAV ANAND          *********************";

   cout<<"\n\nWHAT IS YOUR NAME?\n";

   cin.getline(name,20);

   cout<<"\n\n<<<<<<<<<<<<<<<<<<   WELCOME "<<name<<"   >>>>>>>>>>>>>>>>>>>>>>";

   cout<<"\n\n\nLET'S CREATE OUR PHONEBOOK "<<name<<"  \n\n";

   d1.accept();

   d1.sort();

   do

   {

   cout<<"\n\n\n\n1) DISPLAY YOUR PHONE BOOK\n2) INSERT NEW CONTACT\n3) UPDATE
DETAILS ON EXISTING CONTACT\n4) DELETE CONTACT\n5) DELETE SAME NAME IN
PHONEBOOK\n6) DELETE SAME NUMBERS IN PHONEBOOK\n7) SEARCH\n";

   cin>>ch;

   switch(ch)

   {

   case 2:

   d1.insert();

   d1.sort();

   break;


   case 1:

   d1.display();
```

```
  break;

  case 3:


      cout<<"\n\nENTER THE NAME OF PERSON WHOSE DETAILS YOU WANT TO
UPDATE...\n";

      cin>>n;

  d1.update(n);

  d1.sort();

  break;

  case 4:

  cout<<"\nENTER THE NAME YOU WANT TO DELETE FROM PHONEBOOK\n";

  cin>>name;

  d1.deletecontact(name);

  break;

   case 5:

  d1.deletesamename();

  d1.display();

  break;

  case 6:

  d1.deletesamenumber();

  d1.display();

  break;

  case 7:

  do

  {

  cout<<"1.SEARCH BY NAME\n2.SEARCH BY NUMBER\n3.SEARCH BY GMAIL";

  cin>>a;

  switch(a)

  {

      case 1:
      cout<<"ENTER THE NAME TO BE SEARCHED\n";

      cin>>name;

      d1.searchbyname(name);

      break;

      case 2:

      cout<<"ENTER THE NAME TO BE SEARCHED\n";

      cin>>number;

      d1.searchbynumber(number);

      break;

      case 3:

      cout<<"ENTER THE NAME TO BE SEARCHED\n";

      cin>>gmail;

      d1.searchbygmail(gmail);

      break;

      default:cout<<"\nNO PROPER INPUT GIVEN.....\n";

  }

  cout<<"DO YOU WANT TO CONTINUE SEARCHING?????????";

  cin>>ans;

}while(ans=='y');


  break;

  case 8:d1.deletesamegmail();

  d1.display();

  break;

  default:cout<<"\nNO PROPER INPUT GIVEN..\n";

  }

  cout<<"\n\nDO YOU WANT TO CONTINUE OPERATIONS?????????";

  cin>>ans;

}while(ans=='y');
```

}

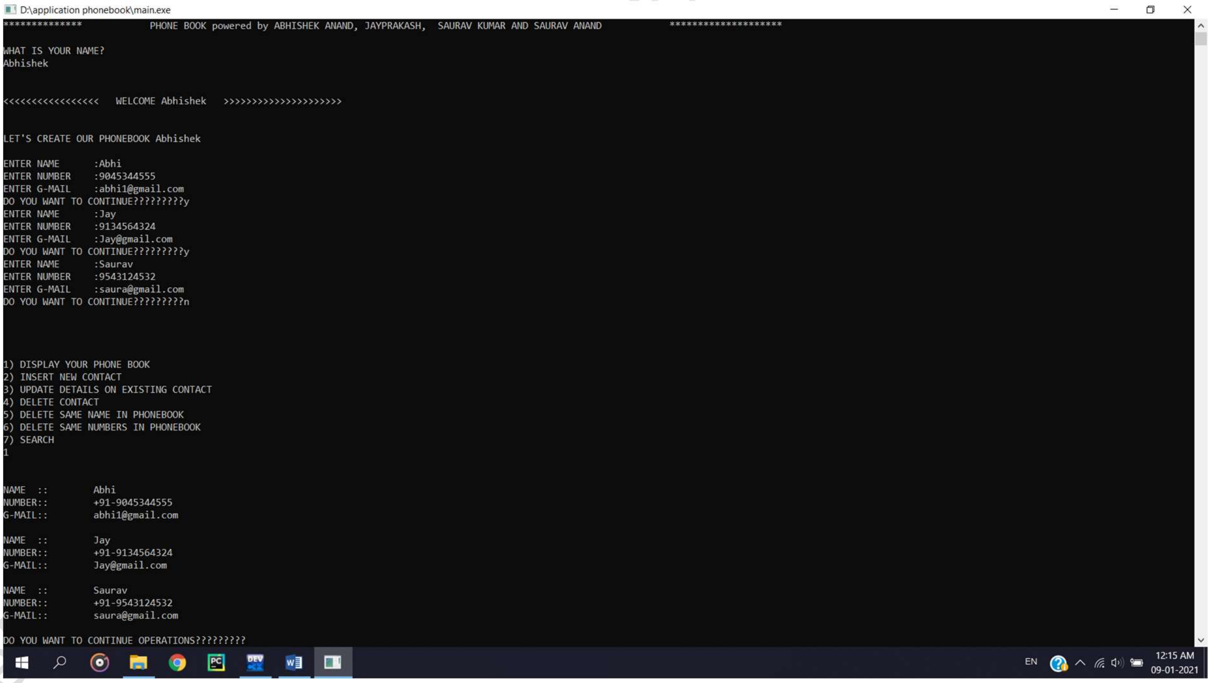
Input Output Screenshot:-


D:\application phonebook\main.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
main.cpp
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
□ Shorten compiler paths
EN
12:13 AM
09-01-2021

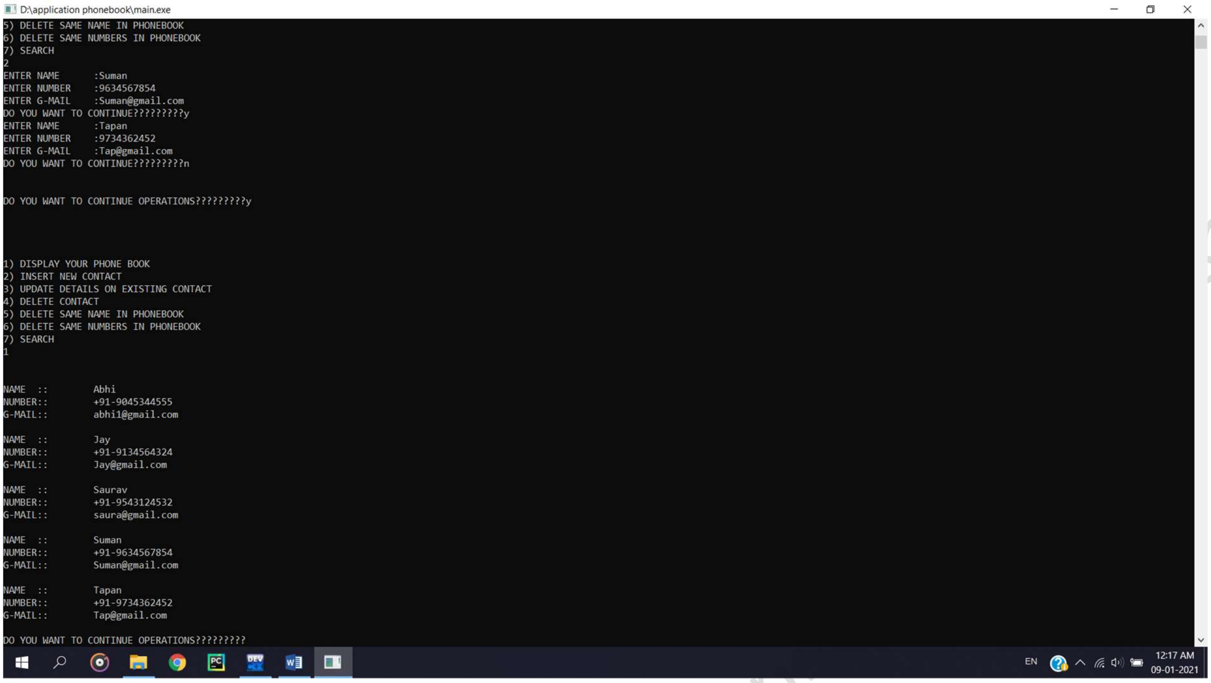
```cpp
#include <iostream>
#include<cstring>
using namespace std;
class dnode
{
	public:
		char number[50];
		char gmail[40];
		char name[30];
		dnode *prev,*next;
		dnode(char n[],char r[],char g[])
{
				strcpy(name,n);
				 strcpy(number,r);
				strcpy(gmail,g);
				 next=NULL;
				 prev=NULL;
}
	friend class dlist;
};
class dlist
{
	dnode *head,*temp,*ptr;

	dnode *ptr1, *ptr2, *dup;
	public:
	dnode *prevn;

	void insert();
	void sort();
	void deletecontact(char n[20]);
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: D:\application phonebook\main.exe
- Output Size: 1.8395471572876 MiB
- Compilation Time: 1.23s
```

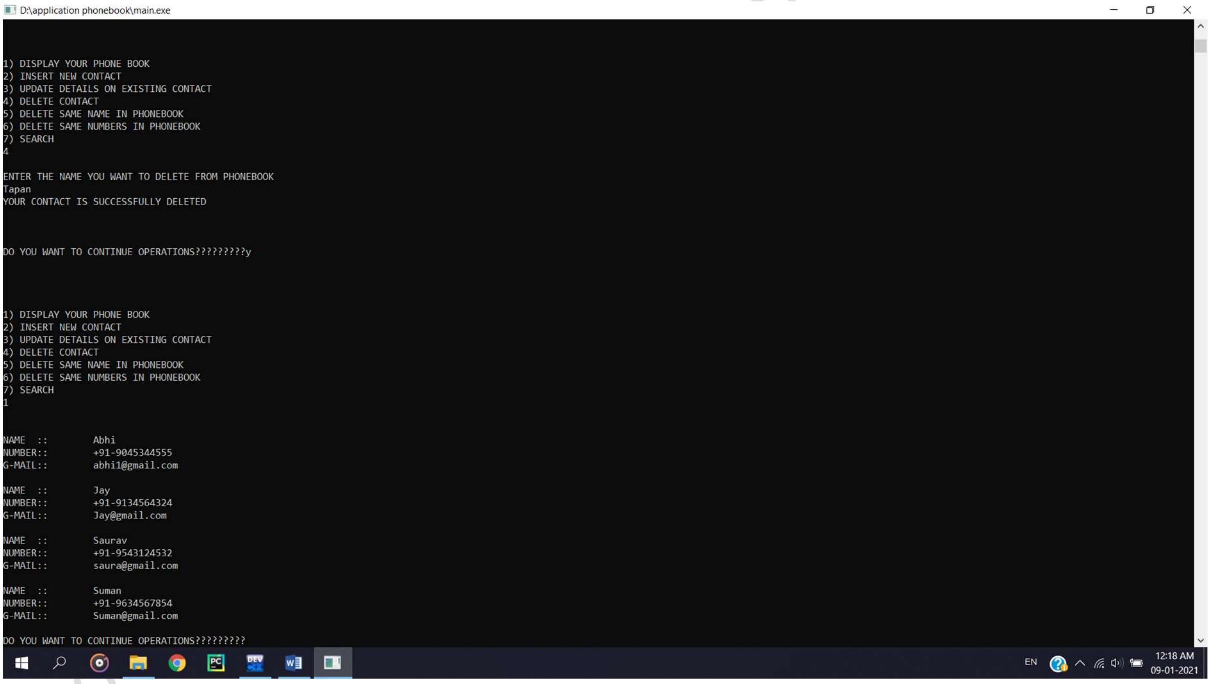
Line: 41 Col: 34 Sel: 0 Lines: 416 Length: 10118 Insert Done parsing in 0.469 seconds



```
**************              PHONE BOOK powered by ABHISHEK ANAND, JAYPRAKASH,  SAURAV KUMAR AND SAURAV ANAND          ********************

WHAT IS YOUR NAME?
Abhishek


<<<<<<<<<<<<<<<<<   WELCOME Abhishek   >>>>>>>>>>>>>>>>>>>>>


LET'S CREATE OUR PHONEBOOK Abhishek

ENTER NAME      :Abhi
ENTER NUMBER    :9045344555
ENTER G-MAIL    :abhi1@gmail.com
DO YOU WANT TO CONTINUE?????????y
ENTER NAME      :Jay
ENTER NUMBER    :9134564324
ENTER G-MAIL    :Jay@gmail.com
DO YOU WANT TO CONTINUE?????????y
ENTER NAME      :Saurav
ENTER NUMBER    :9543124532
ENTER G-MAIL    :saura@gmail.com
DO YOU WANT TO CONTINUE?????????n




1) DISPLAY YOUR PHONE BOOK
2) INSERT NEW CONTACT
3) UPDATE DETAILS ON EXISTING CONTACT
4) DELETE CONTACT
5) DELETE SAME NAME IN PHONEBOOK
6) DELETE SAME NUMBERS IN PHONEBOOK
7) SEARCH
1


NAME  ::        Abhi
NUMBER::        +91-9045344555
G-MAIL::        abhi1@gmail.com

NAME  ::        Jay
NUMBER::        +91-9134564324
G-MAIL::        Jay@gmail.com

NAME  ::        Saurav
NUMBER::        +91-9543124532
G-MAIL::        saura@gmail.com

DO YOU WANT TO CONTINUE OPERATIONS?????????
```



```
5) DELETE SAME NAME IN PHONEBOOK
6) DELETE SAME NUMBERS IN PHONEBOOK
7) SEARCH
2
ENTER NAME      :Suman
ENTER NUMBER    :9634567854
ENTER G-MAIL    :Suman@gmail.com
DO YOU WANT TO CONTINUE?????????y
ENTER NAME      :Tapan
ENTER NUMBER    :9734362452
ENTER G-MAIL    :Tap@gmail.com
DO YOU WANT TO CONTINUE?????????n


DO YOU WANT TO CONTINUE OPERATIONS?????????y




1) DISPLAY YOUR PHONE BOOK
2) INSERT NEW CONTACT
3) UPDATE DETAILS ON EXISTING CONTACT
4) DELETE CONTACT
5) DELETE SAME NAME IN PHONEBOOK
6) DELETE SAME NUMBERS IN PHONEBOOK
7) SEARCH
1


NAME  ::        Abhi
NUMBER::        +91-9045344555
G-MAIL::        abhi1@gmail.com

NAME  ::        Jay
NUMBER::        +91-9134564324
G-MAIL::        Jay@gmail.com

NAME  ::        Saurav
NUMBER::        +91-9543124532
G-MAIL::        saura@gmail.com

NAME  ::        Suman
NUMBER::        +91-9634567854
G-MAIL::        Suman@gmail.com

NAME  ::        Tapan
NUMBER::        +91-9734362452
G-MAIL::        Tap@gmail.com

DO YOU WANT TO CONTINUE OPERATIONS?????????
```



```


1) DISPLAY YOUR PHONE BOOK
2) INSERT NEW CONTACT
3) UPDATE DETAILS ON EXISTING CONTACT
4) DELETE CONTACT
5) DELETE SAME NAME IN PHONEBOOK
6) DELETE SAME NUMBERS IN PHONEBOOK
7) SEARCH
4

ENTER THE NAME YOU WANT TO DELETE FROM PHONEBOOK
Tapan
YOUR CONTACT IS SUCCESSFULLY DELETED


DO YOU WANT TO CONTINUE OPERATIONS?????????y




1) DISPLAY YOUR PHONE BOOK
2) INSERT NEW CONTACT
3) UPDATE DETAILS ON EXISTING CONTACT
4) DELETE CONTACT
5) DELETE SAME NAME IN PHONEBOOK
6) DELETE SAME NUMBERS IN PHONEBOOK
7) SEARCH
1


NAME  ::        Abhi
NUMBER::        +91-9045344555
G-MAIL::        abhi1@gmail.com

NAME  ::        Jay
NUMBER::        +91-9134564324
G-MAIL::        Jay@gmail.com

NAME  ::        Saurav
NUMBER::        +91-9543124532
G-MAIL::        saura@gmail.com

NAME  ::        Suman
NUMBER::        +91-9634567854
G-MAIL::        Suman@gmail.com

DO YOU WANT TO CONTINUE OPERATIONS?????????
```

Discussion:-

By implementing this program, we can execute the contact data, delete data, searching, updating, exit and display Numbers. For creating this application we used arrays, pointer, structure and Doubly linked list. We have written the separate functions for all the operations that we can do in this application. We used C++ language for compilation of this program for the purpose of time and space complexity.