Call Log | Simulation | Contacts -- C++

CODE:

```
#include<bits/stdc++.h>
#include<time.h>
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<string.h>
using namespace std;

struct person //contact list entry structure
{
   char ph[10];
   string pname;
};
////////////////////////////////////////////////
struct Log_list //call log entry structure
{
   string num,name,timelog;

};
////////////////////////////////////////////////
class queue //main class whic contains all calling, logging and storage facilities
{
public:
 int n = 10, front = -1, rear = -1, c=0;
person P[10];
Log_list L[10];

 //////////////////
 void Continue() //Encapsulates the "press-to-continue" procedure
 {
 int c;
 printf( "\nPress a key to continue back to menu" );
 c = getch();
 if (c == 0 || c == 224) getch();
 }
 //////////////////
 void Update_Log(person t) //Updates call log after making a call
 {
   int val, i;
     front = 0;
   if (rear == n - 1)
     {
            for (i = 0; i < n - 1; i++)
              {
                L[i] = L[i + 1];
              }
     }
   else
     {
            rear++;
     }
   L[rear].num=t.ph;
   L[rear].name=t.pname;
   time_t my_time = time(NULL);
```

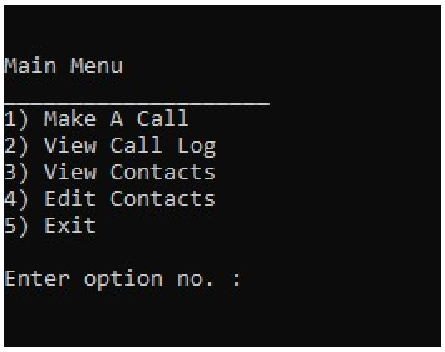
```
  L[rear].timelog=ctime(&my_time);
}
//////////////////
void call(person t) //Placeholder function for simulating placing a call
{
    int i;
    cout<<"\n\n\n";
    cout<<"Calling "<<t.pname<<endl;
    for(i=0;i<4;i++)
    {
        Sleep(2000);
        cout<<".   ";
    }
    cout<<"\nCall not recieved, please try later";
}
//////////////////
void Disp_Log() //Displays call log
{
  if (front == -1)
    {
        cout << "\nCall log is empty!";
    }
  else
    {
                cout<<"CALL LOG\n____________________\n";
                    for (int i = rear; i>=front; i--)
                            {
                            cout<<endl<<L[i].name<<" - "<<L[i].num<<endl<<L[i].timelog<<endl;
                            }
    }
}
//////////////////
void Add_Contact() //Adds contact to contact list
{
    cout<<"\n\n\n";
    person t;
    int i;
    if(c==n)
    {
        cout<<"\nContact storage limit reached (up to 10)\nPlease free up storage to continue";
    }
    else
    {
    cout<<"\nEnter Phone no. : "; cin>>P[c].ph;
    cout<<"\nEnter Name      : "; cin>>P[c].pname;
    for(i=0;i<c;i++)
    {
        if(P[i].pname>P[c].pname)
        {
            t=P[i];
            P[i]=P[c];
            P[c]=t;
        }
    }
    if(c<n)
    {
        c++;
    }
    cout<<"\nContact Added\n";
    }
}
//////////////////
void Disp_Contacts() //Displays contact list
```

```
 {
    int i;
                    cout<<"\n\n\n";
    if(c==0)
    {
       cout<<"No contacts added";
    }
    else
    {
                  cout<<"CONTACT LIST\n____________________";
       for(i=0;i<c;i++)
       {
          cout<<endl<<endl<<i+1<<".\n"<<P[i].pname<<endl<<P[i].ph;
       }
    }
 }
 //////////////////
 void Del_Contact() //Deletes specific contact from contact list
 {
    cout<<"\n\n\n";
    int ind, i;
    person T;
    if(c==0)
    {
       cout<<"No contacts added";
    }
    else
    {
       Disp_Contacts();
       cout<<"\nEnter the index no. of the contact to be deleted : ";
       cin>>ind;
       for(i=ind-1;i<c;i++)
       {
          T=P[i+1];
          P[i+1]=P[i];
          P[i]=T;
       }
    cout<<"\n\n Contact Deleted";
    c--;
    }
 }
 //////////////////
 void menu() //Main UI
{
   int a;
   cout<<"\n\nMain Menu\n____________________\n";
   cout<<"1) Make A Call\n2) View Call Log\n3) View Contacts\n4) Edit Contacts\n5) Exit\n\nEnter option no. : ";
   cin>>a;
   cout<<"\n\n\n";
   if(a==1)
   {
      if(c==0)
      {
         Disp_Contacts();
      }
      else
      {
         Disp_Contacts();
         cout<<"\nEnter index no. of contact to be called : ";
         cin>>a;
         call(P[a-1]);
         Update_Log(P[a-1]);
      }
   }
   else if(a==2)
   {
      Disp_Log();
   }
   else if(a==3)
   {
      Disp_Contacts();
   }
   else if(a==4)
   {
      cout<<"1) Add a contact\n2) Delete a contact\n\nEnter option no. : ";
      cin>>a;
      if(a==1)
      {
         Add_Contact();
      }
      else if(a==2)
      {
         Del_Contact();
      }
      else
      {
         cout<<"Invalid input";

      }
   }
   else if(a==5)
   {
               exit(0);
               }
   else
   {
      cout<<"Invalid Input";

   }
   Continue();//menu_call();
   menu();
}

} temp; //"temp" declared as placeholder call log object
////////////////////////////////////////////////
int main () //Driver Code
{
 temp.menu();
 return 0;
}
```

SAMPLE OUTPUT:

Main UI/Menu

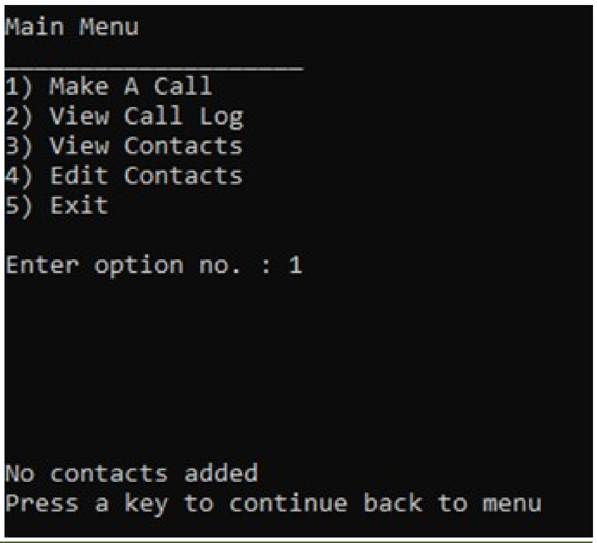
Empty contact list and empty call log at initial state



```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. :
```

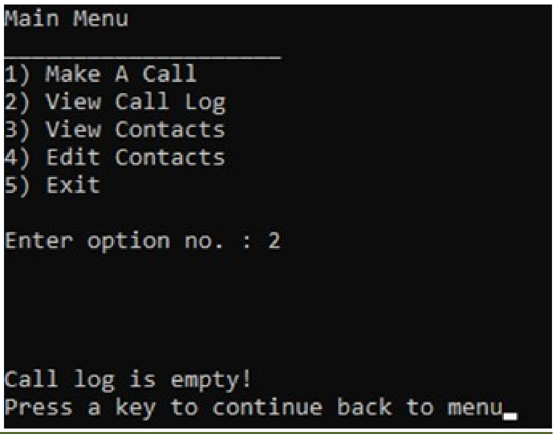
Testing all functions in initial state

(Making a call)



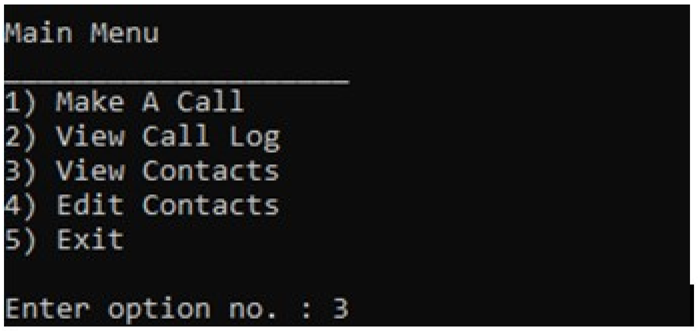
```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 1






No contacts added
Press a key to continue back to menu
```

(Viewing empty call log)

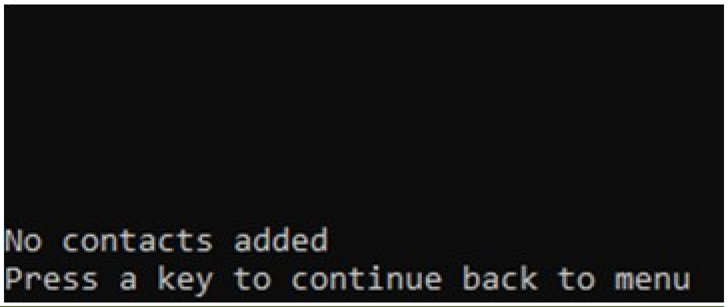


```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 2




Call log is empty!
Press a key to continue back to menu
```

(Viewing empty contact list)

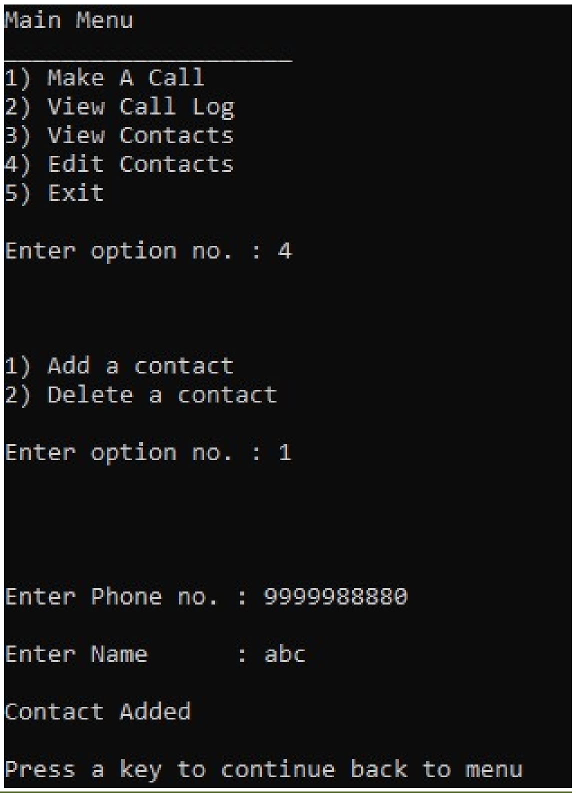


```
Main Menu
_______________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 3
```



```
No contacts added
Press a key to continue back to menu
```

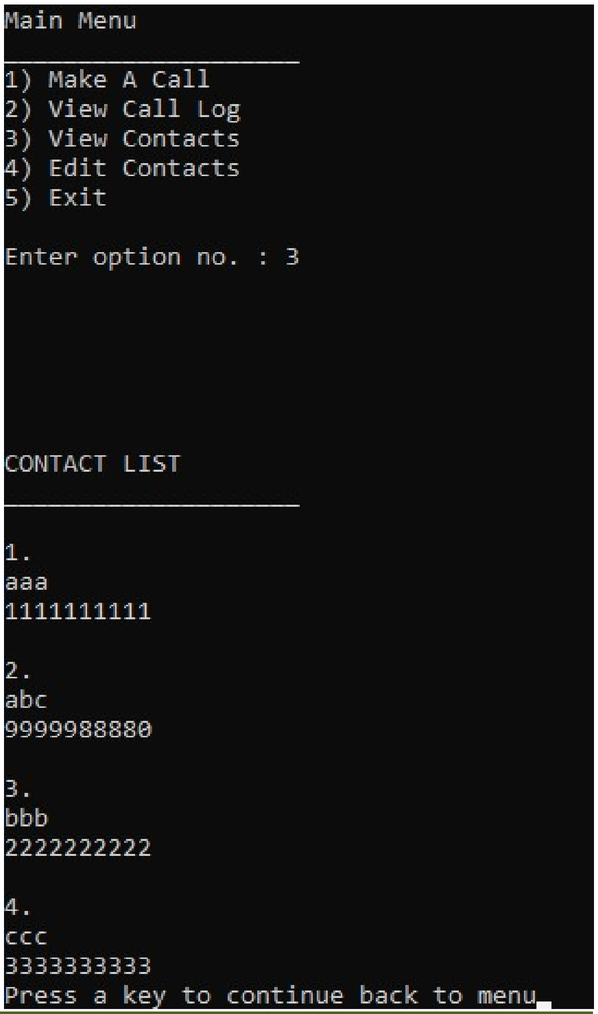
Adding new contacts



```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 4



1) Add a contact
2) Delete a contact

Enter option no. : 1




Enter Phone no. : 9999988880

Enter Name      : abc

Contact Added

Press a key to continue back to menu
```

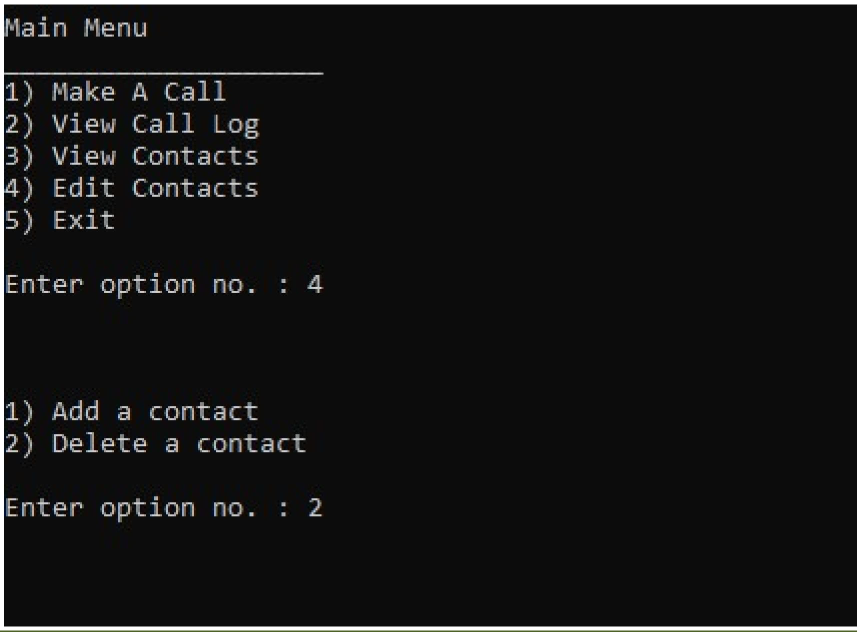
Viewing contact list after filling in entries

(Contact list is automatically sorted alphabetically)

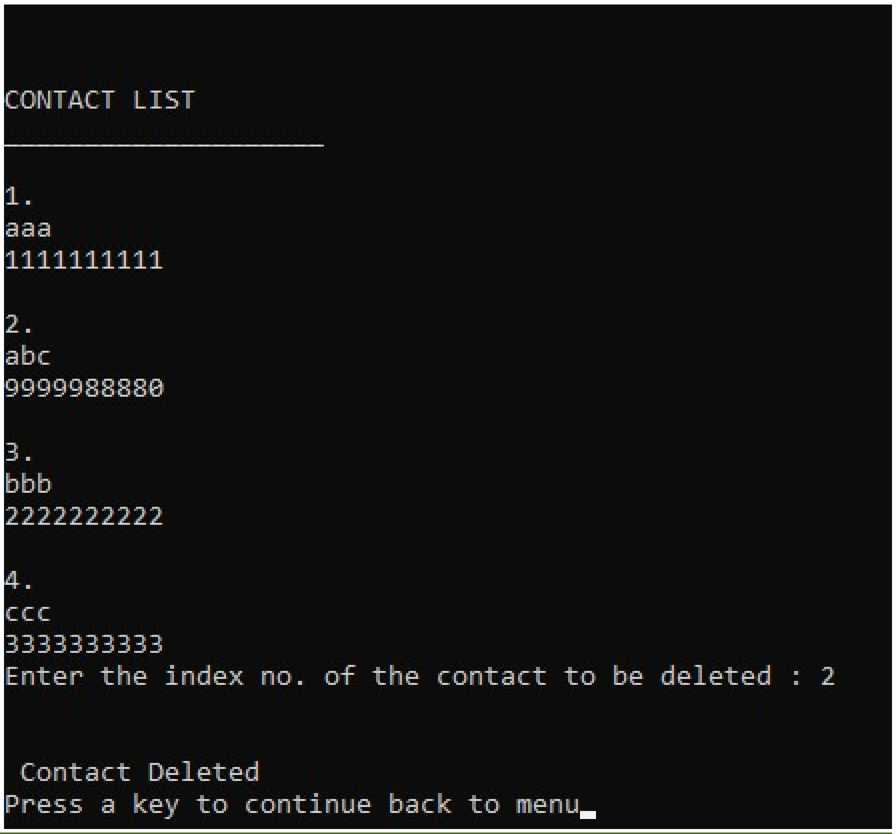


```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 3






CONTACT LIST
____________________

1.
aaa
1111111111

2.
abc
9999988880

3.
bbb
2222222222

4.
ccc
3333333333
Press a key to continue back to menu
```

Deleting contact

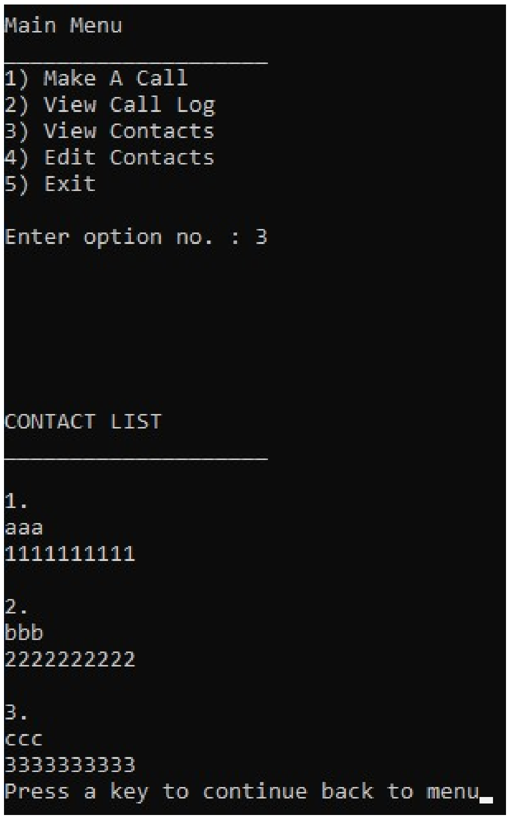


```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 4



1) Add a contact
2) Delete a contact

Enter option no. : 2
```



```
CONTACT LIST
______________________

1.
aaa
1111111111

2.
abc
9999988880

3.
bbb
2222222222

4.
ccc
3333333333
Enter the index no. of the contact to be deleted : 2


 Contact Deleted
Press a key to continue back to menu
```

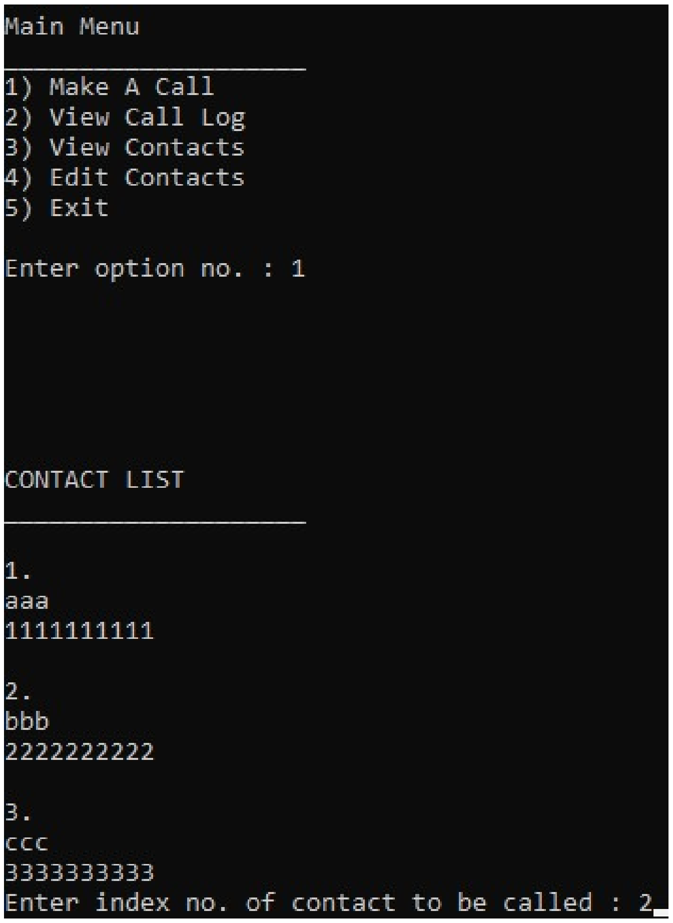
Viewing list again after deletion



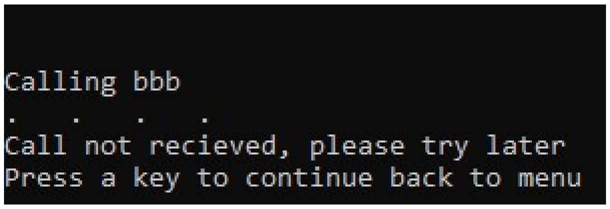
```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 3





CONTACT LIST
____________________

1.
aaa
1111111111

2.
bbb
2222222222

3.
ccc
3333333333
Press a key to continue back to menu
```

Placing a cal

(Pseudo-simulation which lasts for 8 seconds)

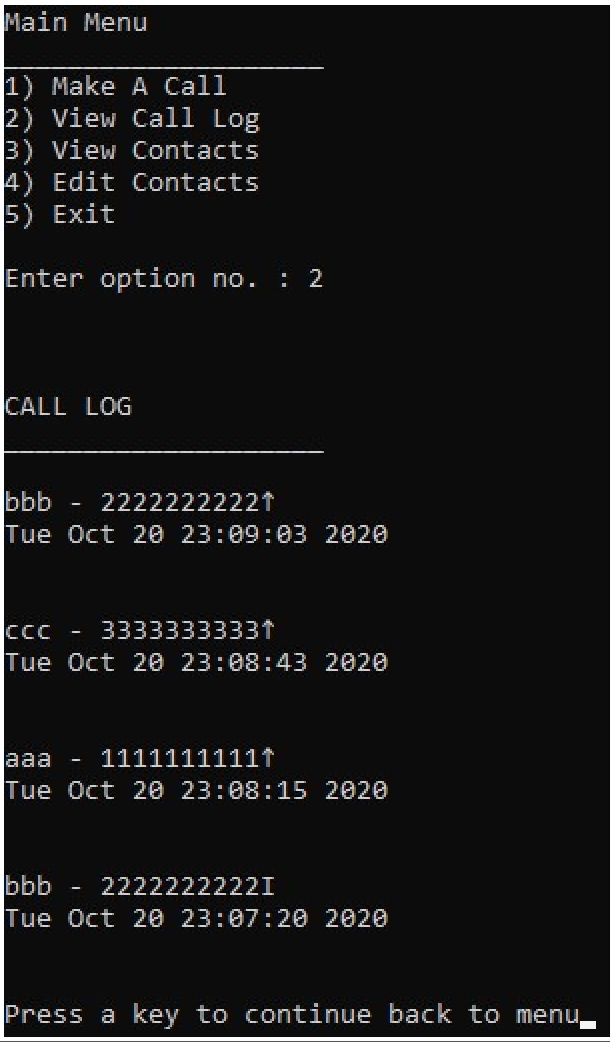


```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 1





CONTACT LIST
____________________

1.
aaa
1111111111

2.
bbb
2222222222

3.
ccc
3333333333
Enter index no. of contact to be called : 2_
```



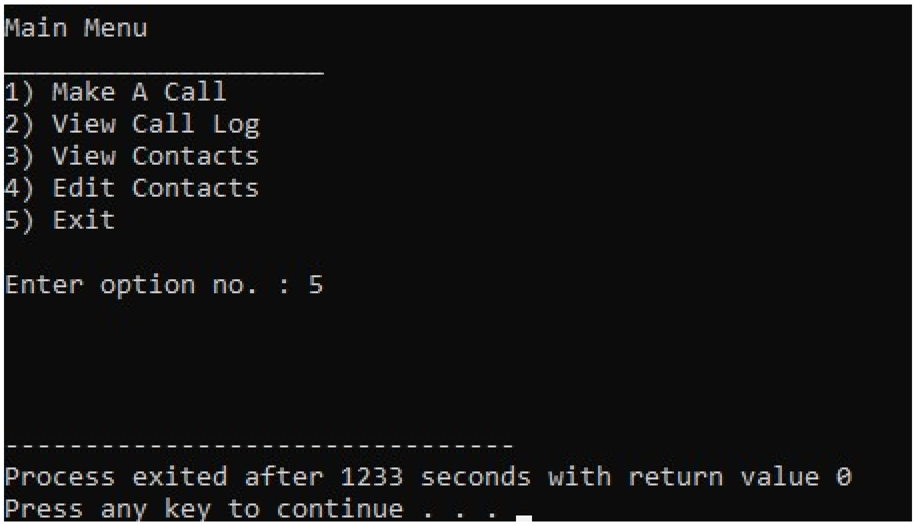
```


Calling bbb
.   .   .   .
Call not recieved, please try later
Press a key to continue back to menu
```

Viewing Call Log after making a few calls



```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 2



CALL LOG
____________________

bbb - 2222222222↑
Tue Oct 20 23:09:03 2020


ccc - 3333333333↑
Tue Oct 20 23:08:43 2020


aaa - 1111111111↑
Tue Oct 20 23:08:15 2020


bbb - 2222222222I
Tue Oct 20 23:07:20 2020


Press a key to continue back to menu
```

Exiting program



```
Main Menu
____________________
1) Make A Call
2) View Call Log
3) View Contacts
4) Edit Contacts
5) Exit

Enter option no. : 5




--------------------------------
Process exited after 1233 seconds with return value 0
Press any key to continue . . .
```

Personal side-notes:

- CONTACTS list has a placeholder capacity of 10 (assigned to variable **n**). Filling up contacts beyond that will result in a message requesting the user to free up storage by deleting some contacts first.
- CALL LOG has a specified limit of 10 (also assigned to **n**) which it temporarily shares with CONTACTS as a placeholder value for the sake of convenience of this test program. CALL LOG will automatically delete the least recent entry whenever another entry is made into it beyond that specified limit in order to make space for it.
- Time of calling in CALL LOG is in a primitive placeholder format for the sake of convenience for this program.

Possible notes for adopting a practical implementation:

- GUI could be improved and instructions can be made more point-and-tap based rather than text-input based.
- Rather than using the contact index no. as input for making calls or deleting contacts, a point-and-click based system (possibly using the aforementioned GUI) coupled with a contact-search system to quickly shortlist contacts based on the user-searched keywords matching their name components.
- Time display can be slightly modified according to required taste in design.
- Proper call functionalities to be substituted instead of its placeholder function.