```
#include<iostream.h>
#include<ctype.h>
#include<conio.h>
#include<string.h>
#include<fstream.h>
#include<time.h>
#include<math.h>
#include<stdio.h>
#include<process.h>
#include<dos.h>
const char a=176;
const char b=219;
const char ll=179;
void tp_lgo()
{
        clrscr();
        cout<<"\n\t\t\t"<<a<<a<<a<<a<<a<<"  "<<a<<"  "<<a<<"  "<<a<<a<<a<<a<<a;
        cout<<"  "<<a<<a<<"  "<<a<<"  "<<a<<"  "<<a<<"   ";
        cout<<"\n\t\t\t"<<"  "<<a<<"  "<<"  "<<a<<a<<a<<a<<"  ";
        cout<<"  "<<a<<"  "<<"  "<<a<<" "<<a<<" "<<a<<"  "<<a<<a<<a;
        cout<<"   "<<" Smart ";
        cout<<"\n\t\t\t"<<"  "<<a<<"  "<<"  "<<a<<"  "<<a<<"  "<<a<<a<<a<<a<<a;
        cout<<"  "<<a<<"  "<<a<<a<<"  "<<a<<"  "<<a<<"   ";

}
void std_err1()
{
        cout<<"\n Error : Filed Cannot Be Left Blank -> ";
}
void std_err2()
{
        cout<<"\n Error : Invalid Parameter -> ";
}
void std_err3()
{
        cout<<"\n Press ~ [SHIFT+`] to retain the old one \n";
}
int digit_sqncr(int key_var_digit)
{
        int sum=0;
        int digit=0;
        while(key_var_digit>0)
        {
                digit=key_var_digit%10;
```

```
                sum+=digit;
                key_var_digit=key_var_digit/10;
        }
        return(sum);
}
int length_calc(char stringlen[100])
{
        for(int len=0;stringlen[len]!='\0';++len);
        return(len);
}
int oddmeter(int key_var_digit2)
{
                if(key_var_digit2%2==0)
                {
                        return(0);
                }
                else if(key_var_digit2%2==1)
                {
                        return(1);
                }
}
int licence_key(char key[])
{
        int validation=0;
        if(key[9]!='\0')
        {
                return(1);

        }
        else if(key[9]=='\0')
        {
                int x;
                x=key[1];
                int temp_key1=digit_sqncr(x);
                x=key[2];
                int temp_key2=digit_sqncr(x);
                if(temp_key1==temp_key2)
                {
                        validation++;
                }
                temp_key1=key[0];
                temp_key2=key[3];
                x=temp_key1+temp_key2;
                int check=oddmeter(x);
```

```
                if(check==1)
                {
                        validation++;
                }
                x=key[5];
                temp_key1=digit_sqncr(x);
                x=key[8];
                temp_key2=digit_sqncr(x);
                if(temp_key1==temp_key2)
                {
                        validation++;
                }
                temp_key1=key[6];
                temp_key2=key[7];
                x=temp_key1+temp_key2;
                check=oddmeter(x);
                if(check==1)
                {
                        validation++;
                }
                if(validation==4)
                {
                        return(0);
                }
                else
                {
                        return(1);
                }
        }
}
class login_profile
{
        private:
        char profile_user_fname[20];
        char profile_user_lname[20];
        char profile_user_appname[20];
        char login_password[36];

        public:
        void getfname()
        {
                gets(profile_user_fname);
        }
        void getlname()
```

```
	{
		gets(profile_user_lname);
	}
	void getappname()
	{
		gets(profile_user_appname);
	}
	void getapppass()
	{
		while(1)
		{
			cin>>login_password;
			char login_password_confirm[36];
			cout<<" Confirm Your Password		: ";
			cin>>login_password_confirm;
			while(1)
			{
				if(strcmp(login_password,login_password_confirm)==0)
				{
					break;
				}
				else
				{
					cout<<"\n ERROR : Password Does not Match \n";
					cout<<" Confirm Your Password		: ";
					cin>>login_password_confirm;
					cout<<"\n\n ! Remember your password well in case you
lose it all the files associated with it will be Destroyed \n";
				}
			}
			break;
		}
	}
	void getdata()
	{
		cout<<"\n Enter Your First Name		: ";
		gets(profile_user_fname);
		cout<<" Enter Your last Name		: ";
		gets(profile_user_lname);
		cout<<" Enter Your Profile Name		: ";
		gets(profile_user_appname);
		cout<<" Enter You Password		: ";
		getapppass();
	}
```

```
void putdata()
{
        cout<<"\n Your First Name          : ";
        cout<<profile_user_fname;
        cout<<"\n Your last Name           : ";
        cout<<profile_user_lname;
        cout<<"\n Your Profile Name        : ";
        cout<<profile_user_appname;
        cout<<"\n You Password             : ";
        cout<<login_password;
}
char* return_profile_user_fname()
{
        return(profile_user_fname);
}
char* return_profile_user_lname()
{
        return(profile_user_lname);
}
char* return_profile_user_appname()
{
        return(profile_user_appname);
}
char* return_login_password()
{
        return(login_password);
}
void modify()
{
        char mprofile_user_fname[20];
        char mprofile_user_lname[20];
        char mprofile_user_appname[20];
        char mlogin_password[36];
        std_err3();
        cout<<"\n Enter Your First Name          : ";
        gets(mprofile_user_fname);
        std_err3();
        cout<<" Enter Your last Name           : ";
        gets(mprofile_user_lname);
        std_err3();
        cout<<" Enter Your Profile Name        : ";
        gets(mprofile_user_appname);
        std_err3();
        cout<<" Enter You Password             : ";
```

```
                cin>>mlogin_password;
                if(strcmp(mprofile_user_fname,"~")!=0)
                {
                        strcpy(profile_user_fname,mprofile_user_fname);
                }
                if(strcmp(mprofile_user_lname,"~")!=0)
                {
                        strcpy(profile_user_lname,mprofile_user_lname);
                }
                if(strcmp(mprofile_user_appname,"~")!=0)
                {
                        strcpy(profile_user_appname,mprofile_user_appname);
                }
                if(strcmp(mlogin_password,"~")!=0)
                {
                        char mlogin_password_confirm[36];
                        cout<<" Confirm Your Password           : ";
                        cin>>mlogin_password_confirm;
                        while(1)
                        {
                                if(strcmp(mlogin_password,mlogin_password_confirm)==0)
                                {
                                        break;
                                }
                                else
                                {
                                        cout<<"\n ERROR : Password Does not Match \n";
                                        cout<<" Confirm Your Password           : ";
                                        cin>>mlogin_password_confirm;
                                        cout<<"\n\n ! Remember your password well in case you
lose it all the files associated with it will be Destroyed \n";
                                }
                        }
                        strcpy(login_password,mlogin_password);
                }

        }
};
class server_security
{
        private:
        char password[40];
        char security_question[50];
        char security_ans[20];
```

```
public:
void getdata()
{
        cout<<"\n Enter the Password : ";
        cin>>password;
        cout<<" Enter The Security Question : ";
        gets(security_question);
        cout<<" Enter the answer of Security Question : ";
        gets(security_ans);
}
void putdata()
{       cout<<"\n The Password : ";
        cout<<password;
        cout<<"\n The Security Question : ";
        cout<<security_question;
        cout<<"\n Enter the answer of Security Question : ";
        cout<<security_ans;
}
void modify()
{
        char mpassword[40]=" ";
        char msecurity_question[50]=" ";
        char msecurity_ans[20]=" ";
        std_err3();
        cout<<" Enter the Password : ";
        cin>>mpassword;
        std_err3();
        cout<<" Enter The Security Question : ";
        gets(msecurity_question);
        std_err3();
        cout<<" Enter the answer of Security Question : ";
        gets(msecurity_ans);
        if(strcmp(mpassword,"~")!=0)
        {
                strcpy(password,mpassword);
        }
        if(strcmp(msecurity_question,"~")!=0)
        {
                strcpy(security_question,msecurity_question);
        }
        if(strcmp(msecurity_ans,"~")!=0)
        {
                strcpy(security_ans,msecurity_ans);
```

```
                }
        }
};
class server_login : public server_security
{
        private:
        char protocol[10];
        char site_name[15];
        char id[36];
        char domain[10];

        public:

        void getdata()
        {
                cout<<"\n Enter the Transfer Protocol : ";
                cin>>protocol;
                cout<<" Enter the Site Name : ";
                gets(site_name);
                cout<<" Enter the Domain  : ";
                cin>>domain;
                cout<<" Enter the Login ID ( complete ): ";
                gets(id);
                server_security :: getdata();
        }
        void putdata()
        {
                cout<<"\n The Transfer Protocol : ";
                cout<<protocol;
                cout<<"\n The Site Name : ";
                cout<<site_name;
                cout<<"\n The Domain  : ";
                cout<<domain;
                cout<<"\n Login ID : ";
                cout<<id;
                server_security :: putdata();
        }
        char* return_id()
        {
                return(id);
        }
        char* return_sitename()
        {
                return(site_name);
```

```
	}
	void modify()
	{
		char mprotocol[10]=" ";
		char msite_name[15]=" ";
		char mid[36]=" ";
		char mdomain[10]=" ";
		std_err3();
		cout<<" Enter the new Transfer Protocol : ";
		cin>>mprotocol;
		std_err3();
		cout<<" Enter the new Site Name : ";
		gets(msite_name);
		std_err3();
		cout<<" Enter the new Domain  : ";
		cin>>mdomain;
		std_err3();
		cout<<" Enter the new  Login ID ( complete ): ";
		gets(mid);
		if(strcmp(mprotocol,"~")!=0)
		{
			strcpy(protocol,mprotocol);
		}
		if(strcmp(msite_name,"~")!=0)
		{
			strcpy(site_name,msite_name);
		}
		if(strcmp(mdomain,"~")!=0)
		{
			strcpy(domain,mdomain);
		}
		if(strcmp(mid,"~")!=0)
		{
			strcpy(id,mid);
		}
		server_security :: modify();
	}
};
class user : public server_login
{
	private:

	char f_name[20];
	char l_name[20];
```

```
        public:

        void getdata()
        {
                cout<<"\n Enter The First Name of User : ";
                gets(f_name);
                cout<<" Enter The Last Name of the User : ";
                gets(l_name);
                server_login :: getdata();
        }
        void putdata()
        {
                cout<<"\n The First Name of the User is : ";
                cout<<f_name;
                cout<<"\n The Last Name of the User is : ";
                cout<<l_name;
                server_login :: putdata();
        }
        void modify()
        {
                char mf_name[20]=" ";
                char ml_name[20]=" ";
                std_err3();
                cout<<" Enter the  New First Name of User  : ";
                gets(mf_name);
                std_err3();
                cout<<" Enter the New Last Name of the User : ";
                gets(ml_name);
                if(strcmp(mf_name,"~")!=0)
                {
                        strcpy(f_name,mf_name);
                }
                if(strcmp(ml_name,"~")!=0)
                {
                        strcpy(l_name,ml_name);
                }
                server_login :: modify();
        }
};
int logix(int logix_low,unsigned char logix_in,int logix_high)
{
        unsigned int logix_in_conver=logix_in;
        if((logix_in_conver>=logix_low)&&(logix_in_conver<=logix_high))
```

```
	{
		return(0);
	}
	else
	{
		return(1);
	}
}
void encom_lgo()
{
	cout<<"\n\n\t\t\t"<<b<<b<<b<<b<<b<<b<<"  ";
	cout<<"\n\t\t\t"<<b<<"            ";
	cout<<"\n\t\t\t"<<b<<b<<b<<b<<"   "<<b<<b<<"  "<<b<<"  ";
	cout<<b<<b<<b<<b<<"  "<<b<<b<<b<<b<<b<<"   "<<b<<b<<" "<<b<<b;
	cout<<"\n\t\t\t"<<b<<"     "<<b<<" "<<b<<" "<<b<<"  ";
	cout<<b<<"  "<<"  "<<b<<"  "<<b<<"  "<<b<<" "<<b<<" "<<b;
	cout<<"\n\t\t\t"<<b<<b<<b<<b<<b<<b<<"  "<<b<<"  "<<b<<b<<"  ";
	cout<<b<<b<<b<<b<<"  "<<b<<b<<b<<b<<b<<"  "<<b<<"  "<<b;
}
void ln_drw(int a)
{
	if(a==1)
	{
		for(int i=0;i<80;++i)
		{
			cout<<"=";
		}
	}
	else
	{	for(int i=0;i<80;++i)
		{
			cout<<"_";
		}
	}

}
void mrgn_sqncr(int spc,int activate_line)
{
	for(int i=0;i<spc;++i)
	{
		cout<<"\t";
	}
	if(activate_line==0)
	{
```

```
                cout<<ll;
        }

}
void hedr_tle1()
{
        cout<<"\t\t\t    Your Smart Password Manager \n";
}
void hedr_tle2()
{
        cout<<"\n\t\t\t         Terminal Window\n";

}
void devlprs()
{
        cout<<"\n";
        cout<<(char)1;
        cout<<" Encom Vers 0.01 \n  Devloped by Harshit Yadav\n\n";
}
void fwrd_msg()
{
        cout<<"\n  PRESS ANY KEY TO CONTINUE :\>";
}
int dsp_gui1()
{
        unsigned char mnu_1;
        int mnu_flag=0;
        cout<<"\n";
        mrgn_sqncr(6,0);
        cout<<" Select An Choice to continue :";
        mrgn_sqncr(6,0);
        cout<<"  1. User Profile\n ";
        mrgn_sqncr(6,0);
        cout<<"  2. Enter Data\n";
        mrgn_sqncr(6,0);
        cout<<"  3. Saved Data\n";
        mrgn_sqncr(6,0);
        cout<<"  4. Time Pass\n ";
        mrgn_sqncr(6,0);
        cout<<"  5. Control Panel\n ";
        mrgn_sqncr(6,0);
        cout<<"  6. About us\n ";
        mrgn_sqncr(6,0);
        cout<<"  0. Exit\n\n ";
```

```
        mrgn_sqncr(6,1);
        cout<<"  --> ";
        cin>>mnu_1;
        mnu_flag=logix(48,mnu_1,54);
        while(mnu_flag)
        {
                mrgn_sqncr(5,1);
                std_err2();
                cout<<"     --> ";
                cin>>mnu_1;
                mnu_flag=logix(48,mnu_1,54);

        }
        int rt_disp_guide=mnu_1;
        return(rt_disp_guide);

}
void login_time()
{
        time_t rawtime;
        struct tm * timeinfo;
        time (&rawtime);
        timeinfo = localtime (&rawtime);
        cout<<asctime(timeinfo);

}
void profile_creator()
{
        clrscr();
        ln_drw(1);
        mrgn_sqncr(2,2);
        cout<<"SIGN UP TERMINAL \n";
        ln_drw(1);
        char case_49_loop;
        int case_49_loop_var1=0;
        login_profile gamma;
        cout<<"\n Enter Your First Name          : ";
        gamma.getfname();
        cout<<" Enter Your last Name           : ";
        gamma.getlname();
        cout<<" Enter Your Profile Name        : ";
        gamma.getappname();
        cout<<" Enter You Password             : ";
        gamma.getapppass();
```

```
        cout<<"\n \n Select From Below : ";
        cout<<"\n 7. Submit ";
        cout<<"\n 8. Retry \n-->  ";
        cin>>case_49_loop;
        case_49_loop_var1=logix(55,case_49_loop,56);
        while(case_49_loop_var1)
        {
                std_err2();
                cin>>case_49_loop;
                case_49_loop_var1=logix(55,case_49_loop,56);
        }
        switch(case_49_loop)
        {
                case 55:
                {
                        ofstream pif;
                        pif.open("EncomLog.exe",ios::out);
                        pif.write((char*)&gamma,sizeof(login_profile));
                        pif.close();
                        cout<<"\n Message : File Sucessfully Written ";
                        break;
                }
                case 56:
                {
                        profile_creator();
                }
        }
}
void tron();
void password(char lock[])
{
        char password[40];
        int value;
        gotoxy(26,12);
        cout<<"Enter your password : ";
        cin>>password;
        value=strcmp(lock,password);
        clrscr();
        if(value!=0)
        {
        gotoxy(33,12);
        cout<<"INVALID PASSWORD ";
        delay(700);
        gotoxy(26,14);
```

```
        cout<<" :> Restarting Application <: ";
        delay(1000);
        tron();
        }

}
void tron()
{
        int tron_1=0;
        int tron_2=0;
        int tron_3=0;
        clrscr();
        cout<<"\n MESSAGE : Activating Tron (Runtime Application Gaurdian)";
        cout<<"\n\n\n\n\n";
        delay(1000);
        cout<<"\nMESSAGE : TRON STATUS REPORT::\n\n";
        cout<<"\n Diagnostics Complete at : \n\n\n\t\t\t";
        login_time();
        cout<<"\n";
        cout<<"\n  Request :   System File                                    [";
        ofstream file_check1;
        file_check1.open("EncomQ.exe",ios::nocreate);
        if(!file_check1)
        {
                cout<<" >__< ";
                tron_1=1;
        }
        else
        {
                cout<<" VALID ";
                tron_1=2;
        }
        cout<<"]\n";
        file_check1.close();
        delay(500);
        cout<<"  Request :      User File                                   [";
        ifstream file_check2;
        file_check2.open("EncomLog.exe",ios::binary|ios::in);
        if(!file_check2)
        {
                cout<<" NOT FOUND ";
                tron_2=2;
        }
        else
```

```
{
        cout<<" FOUND ";
        tron_2=3;

}
file_check2.close();
cout<<"]\n";
delay(500);
cout<<"  Request :      Vault File                                    [";
ofstream file_check3;
file_check3.open("vault.exe",ios::noreplace);
if(!file_check3)
{
        cout<<" FOUND ";
}
else
{
        cout<<" NOT FOUND ";
        tron_3=3;
}
cout<<"]\n";
file_check3.close();
fwrd_msg();
getch();
if(tron_1==1)
{
        clrscr();
        gotoxy(17,12);
        cout<<"ENCOM : System File Invaild Appliction Will Terminate ";
        gotoxy(28,17);
        fwrd_msg();
        getch();
        exit(1);
}
if(tron_2==3)
{
        ifstream file_check2;
        file_check2.open("EncomLog.exe",ios::binary|ios::in);
        login_profile logalpha;
        file_check2.read((char*)&logalpha,sizeof(login_profile));
        clrscr();
        gotoxy(26,10);
        cout<<"User Profile Name   : "<<logalpha.return_profile_user_appname();
        password(logalpha.return_login_password());
```

```
                file_check2.close();
        }
        if(tron_2==2)
        {
                clrscr();
                gotoxy(1,12);
                cout<<"USER FILE NOT FOUND :CREATE NEW PROFILE ( Previous Record File will
be deleted )";
                gotoxy(22,15);
                cout<<"Press 1 To Confirm Else Press Any Key To Exit ";
                int rom;
                gotoxy(40,16);
                cin>>rom;
                if(rom==1)
                {
                        gotoxy(28,18);
                        fwrd_msg();
                        getch();
                   remove("vault.exe");
                        profile_creator();
                }
                else
                {
                        exit(0);
                }
        }
        if(tron_3==3)
        {
                clrscr();
                gotoxy(13,12);
                cout<<" APPLICATION FILE MISSING: Previous Record File Not Found  ";
                gotoxy(28,17);
                fwrd_msg();
                getch();
        }
}
int intelligence(char intellarr[])
{
        int intelsize=length_calc(intellarr);
        long int intlsum=0;
        if(stricmp(intellarr,"exit")==0)
        {
                return(0);
        }
```

```
        else
        {
                for(int intllloop=0;intllloop<intelsize;++intllloop)
                {
                        intlsum+=intellarr[intllloop];
                }
                int intl_para;
                intl_para=oddmeter(intlsum);
                switch(intl_para)
                {
                        case 0:
                        {
                                cout<<"\n Encom  : Yes ";
                                break;
                        }
                        case 1:
                        {
                                cout<<"\n Encom  : No ";
                                break;
                        }
                }
                return(1);
        }
}
void main()
{
        clrscr();
        encom_lgo();
        cout<<"\n\n";
        ln_drw(2);
        hedr_tle1();
        devlprs();
        fwrd_msg();
        getch();
        clrscr();
        tron();
        char start='n';
        do
        {
                clrscr();
                tp_lgo();
                cout<<"\n";
                ln_drw(2);
                int mnu_switch_guide=dsp_gui1();
```

```
switch(mnu_switch_guide)
{
	case 49:
	{
		clrscr();
		ln_drw(1);
		mrgn_sqncr(4,2);
		cout<<"USER PROFILE \n";
		ln_drw(1);
		fstream file_check2;

file_check2.open("EncomLog.exe",ios::nocreate|ios::binary|ios::in|ios::out);
		login_profile logalpha;
		long upos=0;
		file_check2.read((char*)&logalpha,sizeof(login_profile));
		logalpha.putdata();
		cout<<"\n \n Select From Below : ";
		cout<<"\n 7. Edit ";
		cout<<"\n 8. Main Menu \n--> ";
		char ul;
		cin>>ul;
		int ulc=logix(55,ul,56);
		while(ulc)
		{
			std_err2();
			cin>>ul;
			ulc=logix(55,ul,56);
		}
		switch(ul)
		{
			case 55:
			{
				clrscr();
				ln_drw(1);
				mrgn_sqncr(4,2);
				cout<<"EDIT OPTION \n";
				ln_drw(1);
				logalpha.modify();
				cout<<"\n File Modified ";
				file_check2.seekg(upos);

file_check2.write((char*)&logalpha,sizeof(login_profile));
				cout<<"\n File Written ";
				break;
```

```
                }
                case 56:
                {
                        break;
                }
        }
        file_check2.close();
        break;
}
case 50:
{
        clrscr();
        user alpha;
        ln_drw(1);
        mrgn_sqncr(4,2);
        cout<<"ENTER YOUR DETAILS \n";
        ln_drw(1);
        cout<<"\n Object Created of size : ";
        cout<<sizeof(alpha);
        cout<<" Bytes ";
        alpha.getdata();
        getch();
        cout<<" \n\n ";
        cout<<"\n Writing into the memory ";
        ofstream fout;
        fout.open("vault.exe",ios::out|ios::app|ios::binary);
        fout.write((char*)&alpha,sizeof(alpha));
        fout.close();
        cout<<"\n Record Sucessfully Writen into File ";
        getch();
        break;

}
case 51:
{
        clrscr();
        ln_drw(1);
        mrgn_sqncr(4,2);
        cout<<"RETRIVING DATA \n";
        ln_drw(1);
        cout<<"\n\nRecords : ";
        user retrive[6];
        int ocr=0;
```

```
                        ifstream retro;
                        retro.open("vault.exe",ios::binary|ios::in|ios::nocreate);
                        if(!retro)
                        {
                                cout<<"\n No data in File to Display ! ";
                                getch();
                                break;

                        }
                        for(int r=1;!retro.eof();++r)
                        {

                                retro.read((char*)&retrive[r],sizeof(user));
                                if(!retro.eof()==0)
                                {
                                        break;
                                }
                                ocr=1;
                                cout<<"\n"<<" "<<r<<". "<<retrive[r].return_id()<<"    (
"<<retrive[r].return_sitename()<<" )";
                                if(r==5)
                                {
                                        cout<<"\n Want to Retrive more Data (y/n) ";
                                        char sdk;
                                        cin>>sdk;
                                        if(sdk=='y')
                                        {
                                                r=1;
                                                clrscr();
                                        }
                                        if(sdk=='n')
                                        {
                                                break;
                                        }
                                }

                        }
                        if(ocr==1)
                        {
                                retro.close();
                                cout<<"\n\n Choose The Site Number whose data you
want retrive : ";
                                r-=1;
                                int sdk_no;
```

```
cin>>sdk_no;
while(1)
{
	if(sdk_no<=r)
	{
		break;
	}
	else
	{
		std_err2();
		cin>>sdk_no;
	}
}
clrscr();
ln_drw(1);
mrgn_sqncr(4,2);
cout<<retrive[sdk_no].return_sitename()<<"\n";
ln_drw(1);
retrive[sdk_no].putdata();
cout<<"\n \n Select From Below : ";
cout<<"\n 7. Edit ";
cout<<"\n 8. Delete  ";
cout<<"\n 9. Main Menu \n--> ";
char retroo;
cin>>retroo;
int retrool=logix(55,retroo,57);
while(retrool)
{
	std_err2();
	cin>>retroo;
	retrool=logix(55,retroo,57);
}
switch(retroo)
{
	case 55:
	{
		clrscr();
		ln_drw(1);
		mrgn_sqncr(4,2);
		cout<<"EDIT OPTION \n";
		ln_drw(1);
		user retrive_e;
		fstream
retr("vault.exe",ios::in|ios::out|ios::binary);
```

```
                                        long rpos=0;
                                        while(!retr.eof())
                                        {
                                                rpos=retr.tellg();

        retr.read((char*)&retrive_e,sizeof(user));
                                                {

        if((strcmp(retrive_e.return_id(),retrive[sdk_no].return_id())==0)&&(strcmp(retrive_e.ret
urn_sitename(),retrive[sdk_no].return_sitename())==0));
                                                        {

                                                                retrive_e.modify();
                                                                cout<<"\n File
Modified ";
                                                                retr.seekg(rpos);

        retr.write((char*)&retrive_e,sizeof(user));
                                                                cout<<"\n File
Written ";
                                                                break;
                                                        }
                                                }

                                        }
                                        retr.close();
                                        cout<<"\n Message : File Sucessfully
Written ";
                                        break;
                                }
                                case 56:
                                {
                                        user retrive_e;
                                        clrscr();
                                        ln_drw(1);
                                        mrgn_sqncr(4,2);
                                        cout<<"DELETE OPTION \n";
                                        ln_drw(1);
                                        char del='n';
                                        fstream
finc6("vault.exe",ios::in|ios::out|ios::binary);
                                        fstream
finc7("temp.exe",ios::in|ios::out|ios::binary);
                                        while(!finc6.eof())
```

```
                                        {

        finc6.read((char*)&retrive_e,sizeof(user));
                                                if(!finc6.eof()==0)
                                                {
                                                        break;
                                                }

        if(strcmp(retrive_e.return_id(),retrive[sdk_no].return_id())==0)
                                                {
                                                        retrive_e.putdata();
                                                        cout<<"\n Are You Sure You
Want to Delete Data (y/n) : ";
                                                        cin>>del;
                                                        if(del=='n')
                                                        {

        finc7.write((char*)&retrive_e,sizeof(user));
                                                        }
                                                        if(del=='y')
                                                        {
                                                                cout<<"\n Record
Sucessfully Deleted : ";
                                                        }
                                                }

                                                else
                                                {

        finc7.write((char*)&retrive_e,sizeof(user));
                                                        }

                                                }
                                                finc6.close();
                                                finc7.close();
                                                remove("vault.exe");
                                                rename("temp.exe","vault.exe");
                                                break;
                                        }
                                        case 57:
                                        {
                                                break;
                                        }
                                }
```

```
                        cout<<"\n Press any key to continue ";
                        getch();
                }
                else
                {
                        cout<<"\n\n File Empty Nothing To Display ! ";
                }
                break;
        }
        case 52:
        {
                clrscr();
                int intl_loopask=1;
                ln_drw(1);
                mrgn_sqncr(4,2);
                cout<<"TIME PASS \n";
                ln_drw(1);
                cout<<"\n MESSAGE : All the Reply By The System will be Yes Or
No :";
                cout<<"\n {Type exit/EXIT to exit this Menu}";
                while(intl_loopask==1)
                {
                        cout<<"\n Encom  : Enter the Question you want ask  ";
                        cout<<"\n User   : ";
                        char intlques[100];
                        gets(intlques);
                        intl_loopask=intelligence(intlques);
                }
                getch();
                break;

        }
        case 53:
        {
                clrscr();
                tp_lgo();
                cout<<"\n";
                ln_drw(1);
                mrgn_sqncr(4,2);
                cout<<"CONTROL PANEL \n";
                ln_drw(1);
                cout<<"\n \n Select From Below : ";
                cout<<"\n 6. Corrupt APPLICATION ";
```

```
                cout<<"\n 7. Delete User Profile ( Saved Data associated with it
will also be deleted ) ";
                cout<<"\n 8. Delete File Containing Saved File  ";
                cout<<"\n 9. Exit \n -->";
                int sos=0;
                cin>>sos;
                if(sos==9)
                {
                        break;
                }
                cout<<"\n Enter APPLICATION KEY TO CARRY OUT ANY OF THE
FUNCTION ";
                char appkey[9];
                cout<<"\n (ie XXXX-XXXX )\n KEY == ";
                cin>>appkey;
                int h=licence_key(appkey);
                if(h==0)
                {
                        getch();
                        clrscr();
                        switch(sos)
                        {
                                case 6:
                                {
                                        remove("EncomQ.exe");
                                        gotoxy(26,14);
                                        cout<<" :> Event Sucessfully Completed <: ";
                                        gotoxy(26,16);
                                        cout<<" :> Shutting Down Application <: ";
                                        delay(1000);
                                        exit(1);
                                }
                                case 7:
                                {
                                        remove("EncomLog.exe");
                                        gotoxy(26,14);
                                        cout<<" :> Event Sucessfully Completed <: ";
                                        gotoxy(26,16);
                                        cout<<" :> Shutting Down Application <: ";
                                        delay(1000);
                                        exit(1);

                                }
                                case 8:
```

```
                                        {
                                                remove("vault.exe");
                                                gotoxy(26,14);
                                                cout<<" :> Event Sucessfully Completed <: ";
                                                gotoxy(26,16);
                                                cout<<" :> Shutting Down Application <: ";
                                                delay(1000);
                                                exit(1);
                                        }
                                        default:
                                        {
                                                gotoxy(26,14);
                                                cout<<" :> Event Failed (Invalid Parameter)
<: ";
                                                gotoxy(26,16);
                                                cout<<" :> Shutting Down Application <: ";
                                                delay(1000);
                                                exit(1);
                                        }
                                }
                        }
                        else
                        {

                        cout<<"Invalid Key ";
                        }
                        break;

                }
                case 54:
                {
                        clrscr();
                        ln_drw(1);
                        mrgn_sqncr(5,2);
                        cout<<"ABOUT \n";
                        ln_drw(1);
                        encom_lgo();
                        cout<<"\n\n";
                        hedr_tle1();
                        devlprs();
                        cout<<"  Email ID : harshityadav95@gmail.com ";
                        cout<<"\n  Visit : http:\\actroidnotex.blogspot.in \n\n";
                        fwrd_msg();
                        getch();
```

```
                                        break;
                                }
                                case 48:
                                {
                                        exit(1);
                                }

                        }
                        cout<<"\n\n\nExit To Main Menu (y/n) : ";
                        cin>>start;
                }while(start=='y');
                getch();
}
```