**INTRODUCTION**

The use of technology has risen exponentially in all fields that serve the humans. This is mainly due to the comfort and efficiency it provides.

We know, education is need of today. We just can’t be independent in life without education. It empowers us to do various creative things to contribute in society.

So, why not use the technology to ease a student’s work?

STAP has the following tools to aid students:

1. Calculator: simple addition, multiplication, subtract and divide.
2. Contacts: save, edit and delete important contacts.
3. JobCal: calculates the amount of work to be done in each time frame to finish the job before deadline.
4. To-do lists: add and prioritize tasks, search tasks by priority and delete tasks.

**SOURCE CODE**

#include<fstream.h>

#include<conio.h>

#include<dos.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<process.h>

#include<iomanip.h>

int s=0;

char xyz[100];

void loginid();

void mainmenu();

class login

{

char usern[50], passw[50], name[50],ppp[50];

char momn[50],u[50];

char m[50];

public:

char\* return\_usern()

{

return usern;

}

char\* return\_passw()

{

return passw;

}

char\* return\_ppp()

{

return ppp;

}

void getpass();

void check();

void setup()

{

char passw2[10];

cout<<"\n Enter your name : ";

gets(name);

cout<<"\n Enter a Username : ";

gets(usern);

check();

getpass();

strcpy(passw,ppp);

if(strlen(passw)<6)

{

cout<<"\n\n\n Password too short! "<<endl;

getpass();

strcpy(passw,ppp);

}

cout<<"\n PASSWORD CONFIRMATION : ";

getpass();

strcpy(passw2,ppp);

if(strcmp(passw,passw2)==0)

{

cout<<"\n SETUP SUCCESFULLY DONE ! "<<endl;

verify();

}

else

{

cout<<"\n The password you entered is wrong............ ";

cout<<"\n Enter the password again for confirmation CORRECTLY : ";

getpass();

strcpy(passw2,ppp);

if(strcmp(passw,passw2)==0)

{

cout<<"\n SETUP SUCCESFULLY DONE ! "<<endl;

verify();

}

else

{

cout<<"\n SETUP NOT SUCCESSFULL "; delay(1000); exit(0);

}

}

}

void verify()

{

cout<<"\n\n\n Security question in case you forget your password " ;

cout<<"\n\n\n Enter your mobile number : ";

gets(momn);

delay(200);

}

char\* return\_momn()

{

return momn;

}

char\* return\_name()

{

return name;

}

};

void login::getpass()

{

char pss[10],c=' ';

int i=0;

cout<<"\n Enter the password [min length 6] : ";

for(i=0; pss[i]!='\r';i++)

{

pss[i]=getch();

c=pss[i];

if(c==13) break;

else cout<<"\*";

}

pss[i]='\0';

strcpy(ppp,pss);

}

void login::check()

{

login l;

int n;

ifstream f;

f.open("log1.dat", ios::binary);

f.seekg(0, ios::end);

n=f.tellg();

n=n/sizeof(l);

f.seekg(0, ios::beg);

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

{

f.read((char\*)&l, sizeof(l));

if(strcmp(usern,l.return\_usern())==0)

{

cout<<"\n The Username Already Exists! \n\n Enter a different username : ";

gets(usern);

i=-1;

f.seekg(0);

}

}

f.close();

}

void setupid()

{

clrscr();

login l;

ofstream f1;

f1.open("log1.dat",ios::binary|ios::app);

int i=0;

textcolor(YELLOW);

cprintf("\n ENTER THE DETAILS TO SIGNUP : ");

while(i==0)

{

l.setup();

f1.write((char\*)&l, sizeof(l));

i++;

}

f1.close();

loginid();

}

void loginid()

{

clrscr();

login l;

ifstream f1;

f1.open("log1.dat", ios::binary);

char un[50],pw[50],y, vn[50];

int f=0;

textcolor(YELLOW);

cprintf("\n ENTER THE DETAILS TO LOGIN : ");

cout<<"\n\n Enter your username : ";

gets(un);

l.getpass();

strcpy(pw,l.return\_ppp());

f1.seekg(0);

while(f1.read((char\*)&l, sizeof(l)))

{

if(strcmp(un,l.return\_usern())==0 && strcmp(pw,l.return\_passw())==0)

{

f++;

strcpy(xyz,l.return\_name());

cout<<"Login Successful!" <<"\nWELCOME "<<xyz;

delay(500);

mainmenu(); break;

}

else if((strcmp(un,l.return\_usern())==0 && strcmp(pw,l.return\_passw())!=0) || (strcmp(pw,l.return\_passw())==0 && strcmp(un,l.return\_usern())!=0))

{

f++;

cout<<"\n\n\n\n Have you forgotten your password/username?(y/n) : ";

cin>>y;

if (y=='y')

{

cout<<"\n\n\n\n\n Enter your mobile number :";

gets(vn);

if( strcmp(vn,l.return\_momn())==0)

{

cout<<"WELCOME---> "<<l.return\_name();

mainmenu();

strcpy(xyz,l.return\_name()); break;

}

}

}

}

if(f==0)

{

cout<<"\n\n\n Loginid doesnt exist, please create one. ";

delay(1500);

f1.close();

setupid();

}

f1.close();

}

void cal()

{

clrscr();

float a, b, res;

char choice, ch;

do

{

cout<<"1.Addition\n";

cout<<"2.Subtraction\n";

cout<<"3.Multiplication\n";

cout<<"4.Division\n";

cout<<"5.Exit to MainMenu \n\n";

cout<<"Enter Your Choice : ";

cin>>choice;

switch(choice)

{

case '1' : cout<<"Enter two number : ";

cin>>a>>b;

res=a+b;

cout<<"Result = "<<res;

break;

case '2' : cout<<"Enter two number : ";

cin>>a>>b;

res=a-b;

cout<<"Result = "<<res;

break;

case '3' : cout<<"Enter two number : ";

cin>>a>>b;

res=a\*b;

cout<<"Result = "<<res;

break;

case '4' : cout<<"Enter two number : ";

cin>>a>>b;

res=a/b;

cout<<"Result = "<<res;

break;

case '5' : mainmenu();

break;

default : cout<<"Wrong Choice..!!";

break;

}

cout<<"\n------------------------------------\n";

}

while(choice!=5 && choice!=getchar());

}

void cmain();

void ccmain();

class contact

{

char ph[100];

char name[50],add[100],email[50];

public:

void create\_contact()

{

cout<<"Phone: ";

gets(ph);

cout<<"Name: ";

gets(name);

cout<<"Address: ";

gets(add);

cout<<"Email address: ";

gets(email);

cout<<"\n";

}

void show\_contact()

{

cout<<endl<<"Phone: "<<ph;

cout<<endl<<"Name: "<<name;

cout<<endl<<"Address: "<<add;

cout<<endl<<"Email Address : "<<email;

}

char\* getPhone()

{

return ph;

}

char\* getName()

{

return name;

}

char\* getAddress()

{

return add;

}

char\* getEmail()

{

return email;

}

};

fstream fp;

contact cont;

void save\_contact()

{

fp.open("cb.dat",ios::out|ios::app);

char ch='y';

while(ch=='y')

{cont.create\_contact();

fp.write((char\*)&cont,sizeof(cont));

cout<<" Do you want to add one more?(y/n) ";

cin>>ch;}

fp.close();

cout<<endl<<endl<<"Contact(s) Has Been Sucessfully Created...";

ccmain();

}

void show\_all\_contacts()

{

clrscr();

cout<<"\n\t\t================================\n\t\t\tLIST OF CONTACTS\n\t\t================================\n";

fp.open("cb.dat",ios::binary|ios::in);

while(fp.read((char\*)&cont,sizeof(cont)))

{

cont.show\_contact();

cout<<endl<<"=================================================\n"<<endl;

}

fp.close();

int opt;

cout<<"\n\n\n..::Enter the Choice:\n\n\t[1] Contact Menu\t\t[0] Exit\n";

cin>>opt;

switch (opt)

{

case 1: clrscr();

cmain();

case 0: mainmenu();

}

}

void display\_contact(char num[100])

{

int found;

int ch;

found=0;

fp.open("cb.dat",ios::binary|ios::in);

while(fp.read((char\*)&cont,sizeof(cont)))

{

if(strcmp(cont.getPhone(),num)==0)

{

clrscr();

cont.show\_contact();

found=1;

}

}

fp.close();

if(found == 0)

{ cout<<"\n\nNo record found...";

}

int opt;

cout<<"\n\n\n..::Enter the Choice:\n\n\t[1] Contact Menu\t\t[0] Exit\n";

cin>>opt;

switch (opt)

{

case 1: clrscr();

cmain();

case 0: mainmenu();

}

}

void edit\_contact()

{

char num[100];

int found=0;

clrscr();

cout<<"..::Edit contact\n===============================\n\n\t..::Enter the number of contact to edit:";

cin>>num;

fp.open("cb.dat",ios::in|ios::out);

while(fp.read((char\*)&cont,sizeof(cont)) && found==0)

{

if(strcmp(cont.getPhone(),num)==0)

{

cont.show\_contact();

cout<<"\nPlease Enter The New Details of Contact: "<<endl;

cont.create\_contact();

int pos=-1\*sizeof(cont);

fp.seekp(pos,ios::cur);

fp.write((char\*)&cont,sizeof(cont));

cout<<endl<<endl<<"\t Contact Successfully Updated...";

found=1;

}

}

fp.close();

if(found==0)

cout<<endl<<endl<<"Contact Not Found...";

ccmain();

}

void delete\_contact()

{

ifstream f("cb.dat",ios::binary);

ofstream ff("tt.dat",ios::binary);

char num[100];

int g=0;

clrscr();

cout<<endl<<endl<<"Please Enter The contact #: ";

gets(num);

while(f.read((char\*)&cont,sizeof(cont)))

{

if(strcmp(cont.getPhone(),num)!=0)

{

ff.write((char\*)&cont,sizeof(cont));

g++;

}

}

if(g==0)

cout<<"\n\n Contact does not exist ";

else

cout<<"\nContact with ph. no."<<num<<" has been deleted."<<endl;

f.close();

ff.close();

remove("cb.dat");

rename("tt.dat","cb.dat");

int opt;

cout<<"\n\n\n..::Enter the Choice:\n\n\t[1] Contact Menu\t\t[0] Exit\n";

cin>>opt;

switch (opt)

{

case 1: clrscr();

cmain();

case 0: mainmenu();

}

}

void ccmain()

{ clrscr();

int opt;

cout<<"\n\n\n..::Enter the Choice:\n\n\t[1] Contact Menu\t\t[0] Exit\n";

cin>>opt;

switch (opt)

{

case 1: clrscr();

cmain();

case 0: mainmenu();

}

}

void cmain()

{ clrscr();

int ch;

cout<<"\n\t Welcome to Contact Management System ";

cout<<"\n\n\n\t\t\tMAIN MENU\n\t\t=====================\n\t\t[1] Add a new Contact\n\t\t[2] List all Contacts\n\t\t[3] Search for contact\n\t\t[4] Edit a Contact\n\t\t[5] Delete a Contact\n\t\t[0] Exit to MainMenu\n\t\t=================\n\t\t";

cout<<"Enter the choice:";

cin>>ch;

switch(ch)

{

case 0: mainmenu();

break;

case 1: save\_contact();

break;

case 2: show\_all\_contacts();

break;

case 3: char num[100];

clrscr();

cout<<"\n\n\tPhone: ";

cin>>num;

display\_contact(num);

break;

case 4: edit\_contact();

break;

case 5: delete\_contact();

break;

default:break;

}

}

void jm1();

void jm();

int round(float);

int round1(float);

void jobcal()

{ clrscr();

float a, b,c,d ;

cout<<"Enter time frame (number of days, hours or minutes) :" ;

cin>>d ;

cout<<"\nEnter number of tasks :" ;

cin>>c ;

b=c/d ;

a=d/c ;

int e , f ;

e=round(a) ;

f=round1(b) ;

cout<<"\nNumber of tasks to be done in a time frame:"<<f;

cout<<"\nTime frame alloted for each task: " <<e<<endl;

jm1();

}

int round1(float w)

{

if((w+0.9)>=(int (w)+1))

w=w+1 ;

else

w=int (w);

return(w) ;

}

int round(float w)

{

if((w+0.5)>=(int (w)+1))

w=w+1 ;

else

w=int (w);

return(w) ;

}

void jm()

{ clrscr();

int tm;

cout<<"\n\n 1.CALCULATE ";

cout<<"\n\n 2.BACK TO MAIN MENU";

cout<<"\n Choose:";

cin>>tm;

switch(tm)

{

case 1 : jobcal();

break;

case 2 : mainmenu();

break;

}

}

void jm1()

{ int tm;

cout<<"\n\n 1.CALCULATE ";

cout<<"\n\n 2.BACK TO MAIN MENU";

cout<<"\n Choose:";

cin>>tm;

switch(tm)

{

case 1 : jobcal();

break;

case 2 : mainmenu();

break;

}

}

class mtd

{ public:

char description[100];

int priority;

char dueDate[100];

void accept()

{

cout << "\n\nwhat is the description of the activity?" << endl;

gets(description);

cout << "how urgent is this? rank 1-5. 1 for not really important, 5 for very important" << endl;

cin >> priority;

cout << "when is the due date? is day/month format" << endl;

cin >> dueDate;

}

void display()

{

cout<<"\n\nTask description: "<<description<<endl;

cout<<"Priority:"<<priority<<endl;

cout<<"Duedate:"<<dueDate<<endl;

}

};

void addToList()

{ ofstream f4;

mtd d;

f4.open("list1.dat",ios::binary);

char ch='y';

while(ch=='y')

{

d.accept();

f4.write((char\*)&d,sizeof(d));

cout<<"DO you wanna comtinue";cin>>ch;

}

f4.close();

}

void getByPriority()

{ int n;

mtd m;

cout<<" Enter the priority number to be searched ";

cin>>n;

ifstream f5;

f5.open("list1.dat",ios::binary);

int flag=0;

while(f5.read((char\*)&m,sizeof(m)))

{

if(n==m.priority)

{

m.display();

flag=1;}

}

if(flag==0)

cout<<" No priority of "<<n<<"number!";

f5.close();

}

void del()

{

ifstream f1("list1.dat",ios::binary);

ofstream f2("temp.dat",ios::binary);

mtd del;

char task[100];

f1.seekg(0);

cout<<"\n\nEnter the description of the task to be deleted:"<<endl;

gets(task);

int record=0;

while(f1.read((char\*)&del,sizeof(del)))

{

if(strcmp(del.description,task)!=0)

{

f2.write((char\*)&del,sizeof(del));

record++;

}

}

if(record==0)

cout<<"\n\nTask does not exist"<<endl;

else

cout<<"\nTask with description "<<task<<" has been deleted."<<endl;

f1.close();

f2.close();

remove("list1.dat");

rename("temp.dat","list1.dat");

}

void todo()

{

clrscr();

int choice = 0;

char nextmove = 'y';

while (nextmove = 'y')

{

cout << "\n\n\t\tToDo Menu" << endl;

cout << "Choose (1-5)" << endl;

cout << "1. Add to my to do list" << endl;

cout << "2. Search by Specific Priority" << endl;

cout << "3. Delete a task "<< endl;

cout << "4. Exit To MainMenu"<<endl;

cin >> choice;

if (choice == 1)

addToList();

else if(choice==2)

getByPriority();

else if(choice==3)

del();

else if(choice==4)

mainmenu();

}

}

void mainmenu()

{

clrscr();

textcolor(RED);

cprintf("\n\n MAIN MENU ");

cout<<"\n\n 1.CALCULATOR ";

cout<<"\n\n 2.CONTACTS ";

cout<<"\n\n 3.JOBCAL ";

cout<<"\n\n 4.ToDo ";

cout<<"\n\n 5. Quit";

int tm;

cout<<"\n\n CHOOSE(1-5) : ";

cin>>tm;

switch(tm)

{

case 1 : cal();

break;

case 2 : cmain();

break;

case 3 : jm();

break;

case 4 : todo(); break;

case 5: exit(0);

default: cout<<"\n\n INVALID INPUT!!!! SO WE MIGHT AS WELL QUIT :-) ;-P !!!!!!!!"; delay(1300); exit(0);

}

}

void main()

{

clrscr();

char www[]="WELCOME TO STAP";

int x=0;

cout<<"\n ";

textcolor(RED);

while(www[x]!='\0')

{

if(www[x]==' ' && www[x+1]==' ')

{

cout<<"\n\n ";

x=x+2;

}

else

{

cout<<www[x];

delay(75);

x++;

}

}

delay(2000);

int t;

cout<<"\n\n";

cprintf(" 1.LOGIN ");

cout<<"\n\n";

cprintf(" 2.SIGNUP ");

cout<<"\n\n";

cprintf(" Enter your choice : ");

cin>>t;

switch(t)

{

case 1: loginid();

break;

case 2: setupid();

break;

default: cout<<"\n\n INVALID INPUT!"; delay(1000); exit(0);

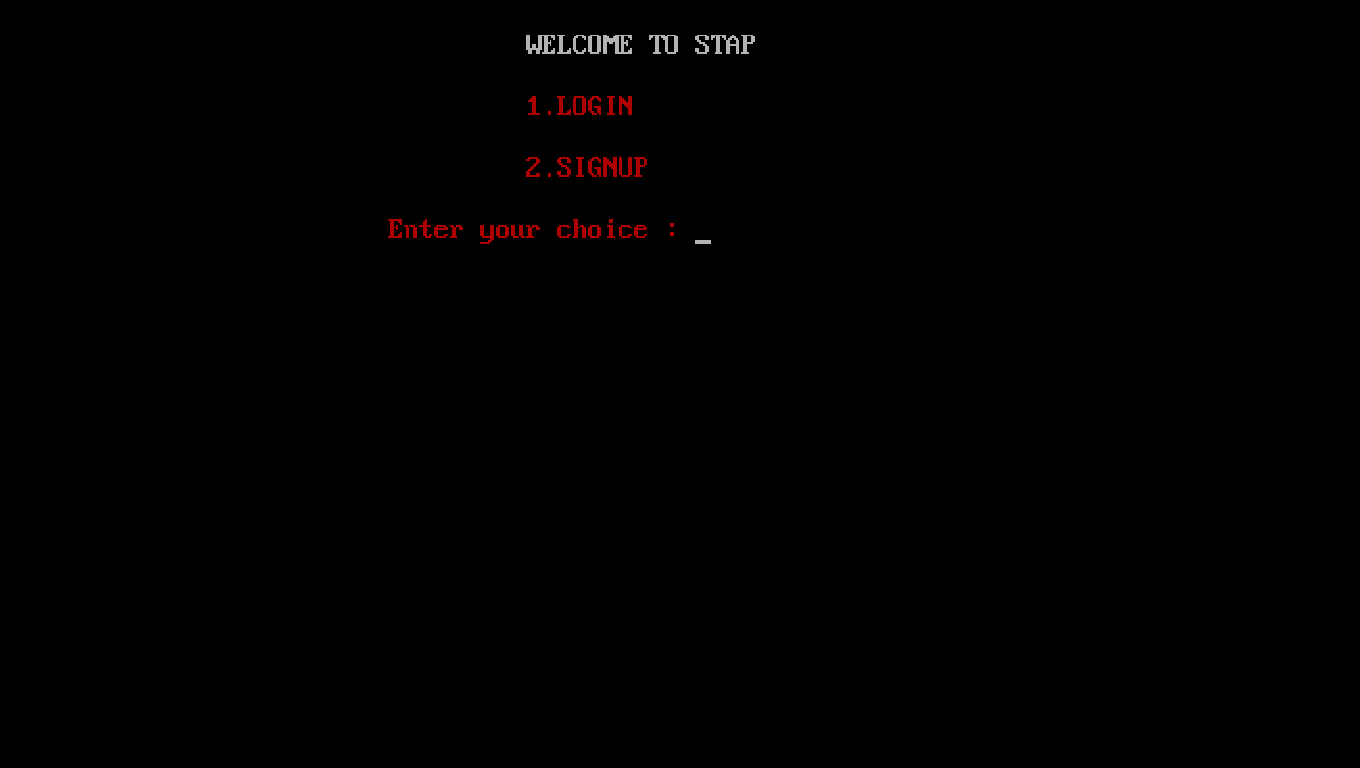
}

getch();

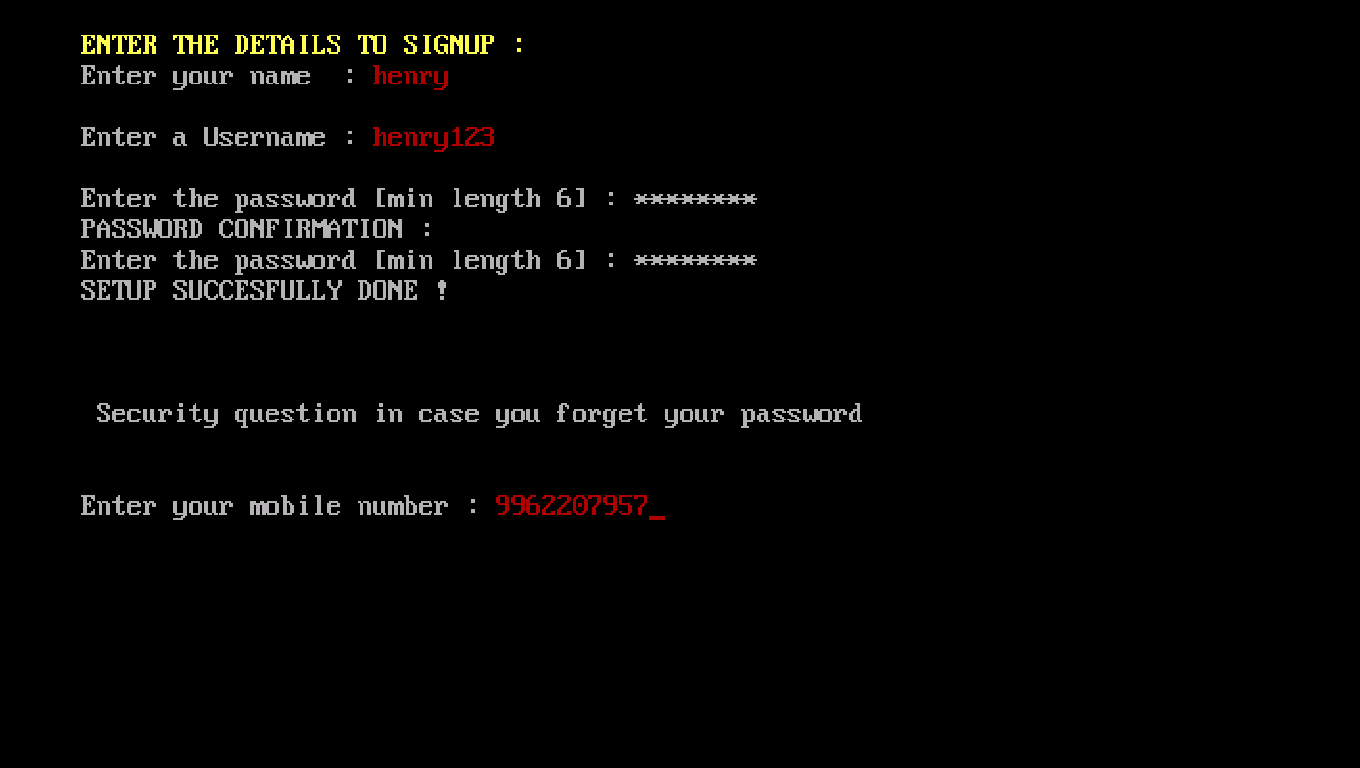
}

**OUTPUT SCREEN**

**Welcome page**



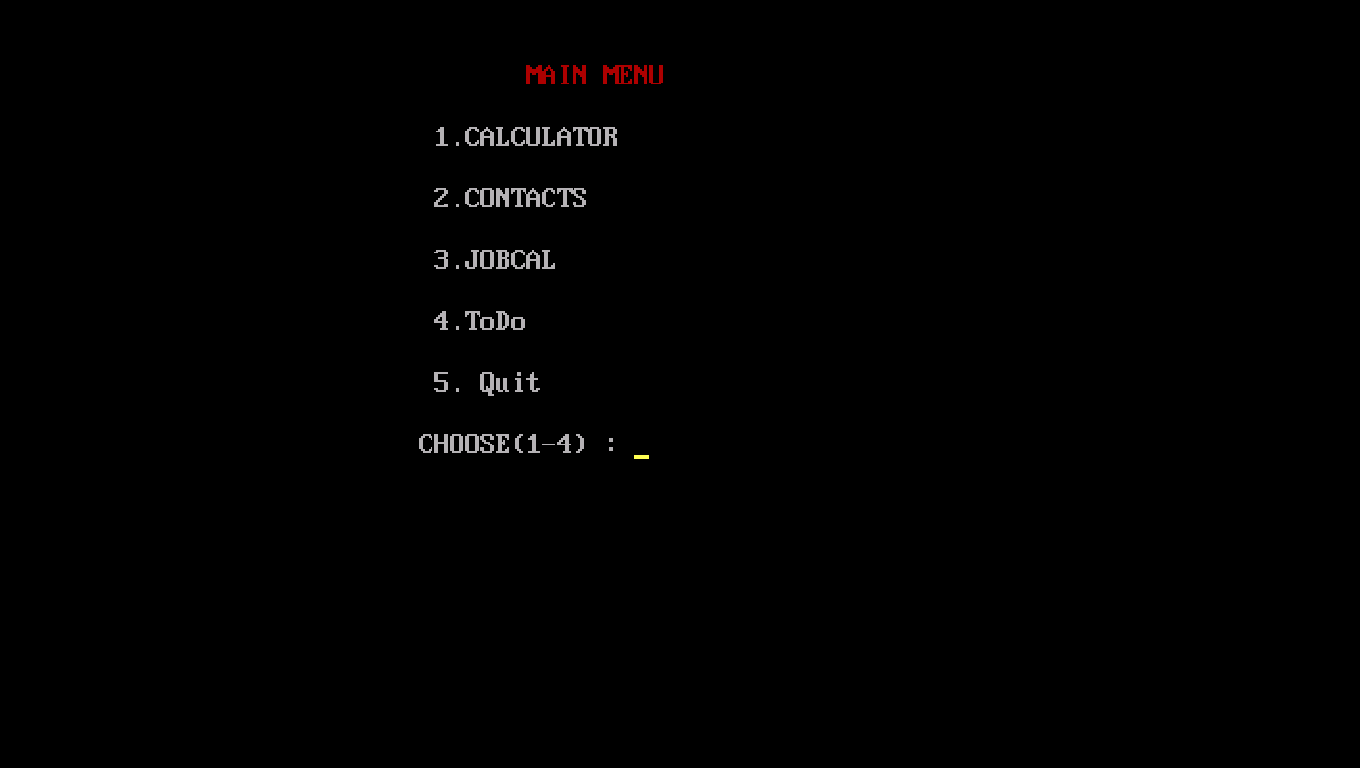
**Signup page**



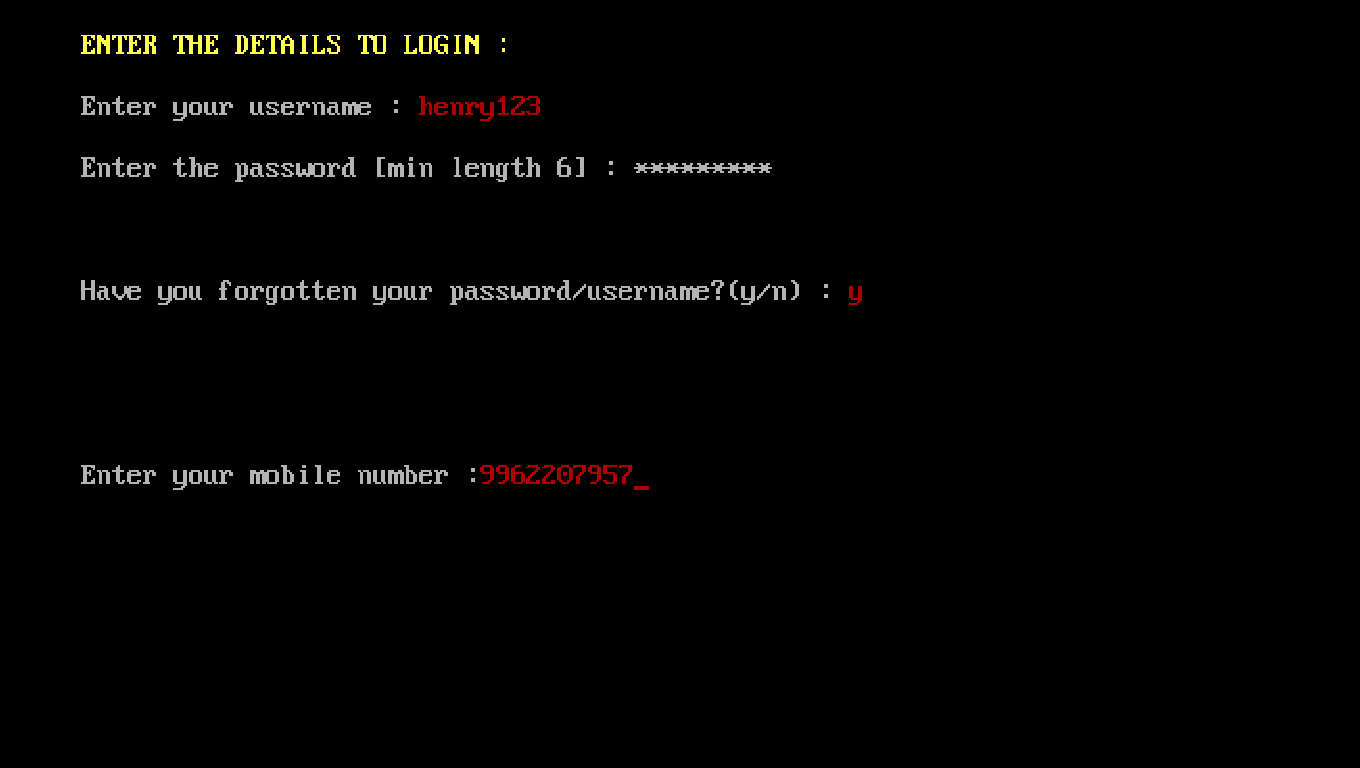
**Login Page**



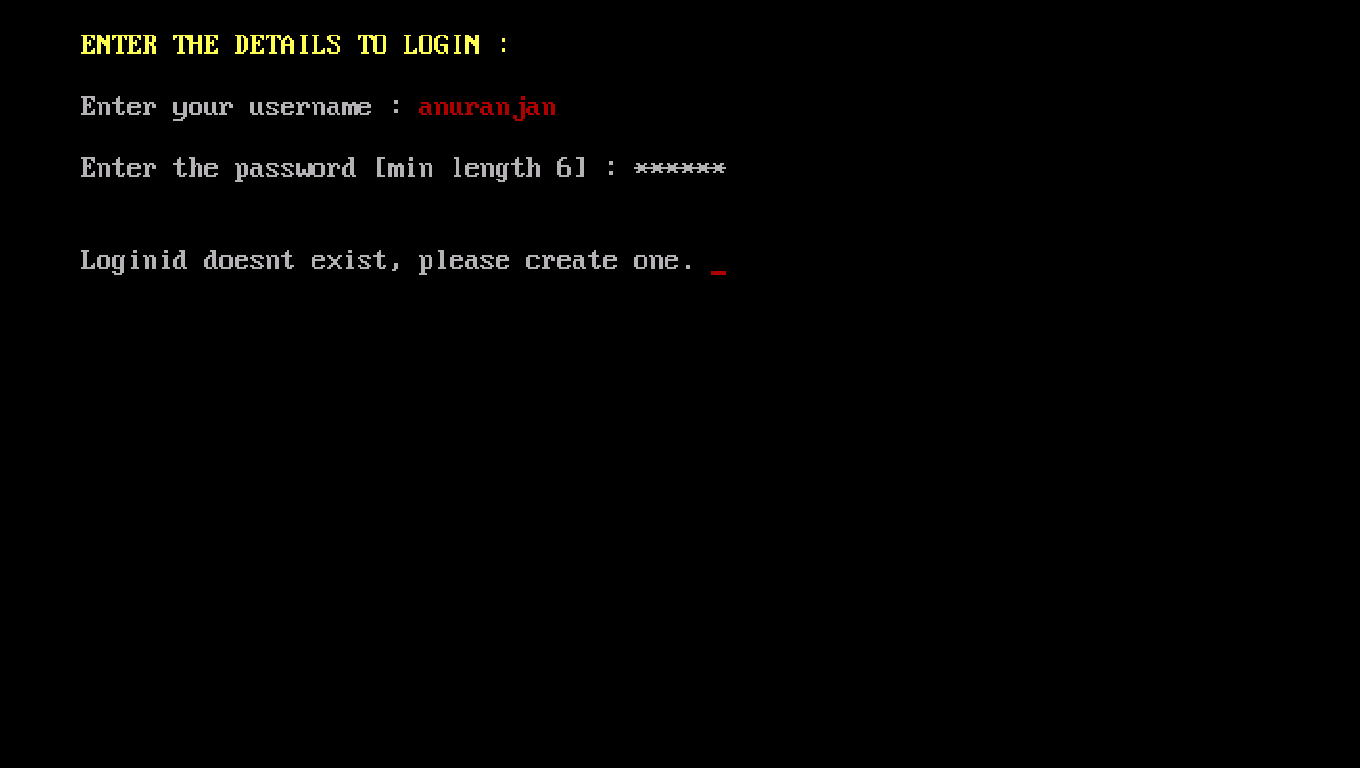
**Main Menu**



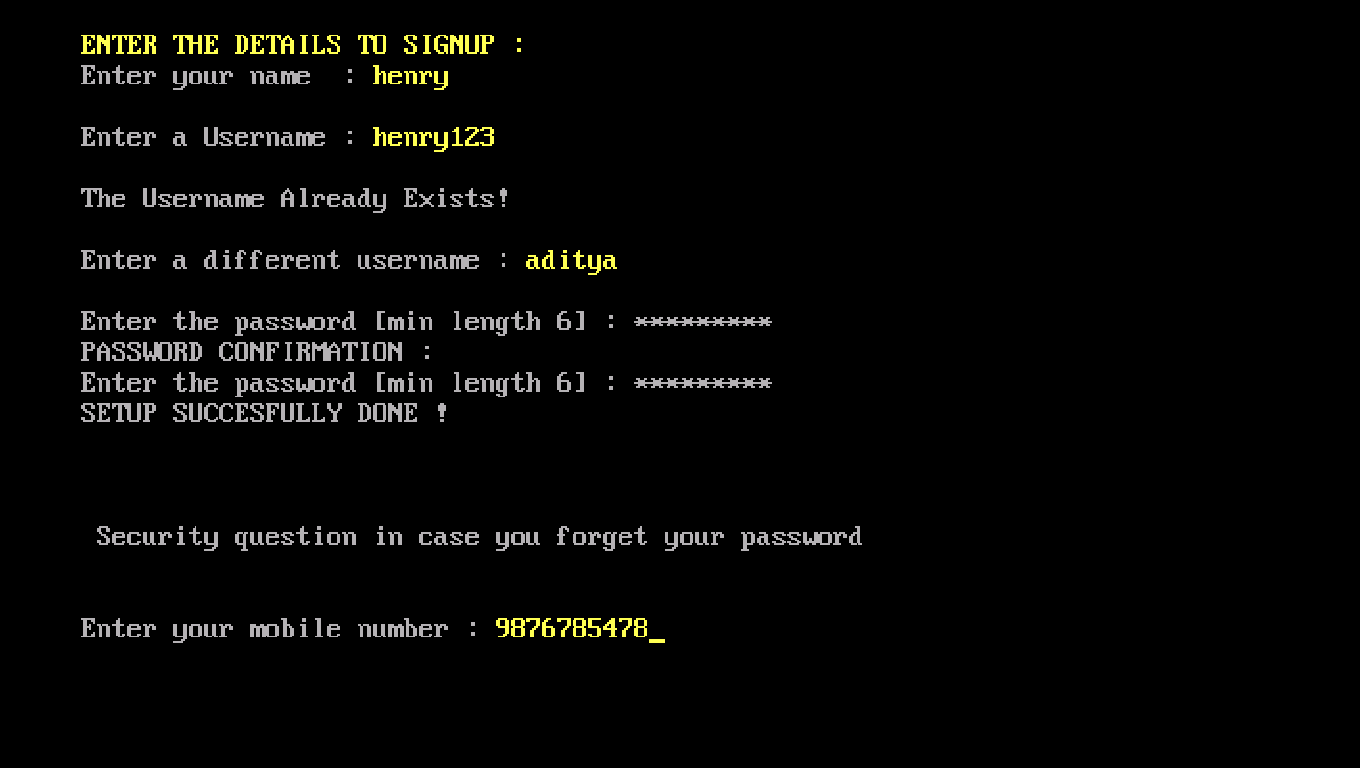
**Forgot login details?**



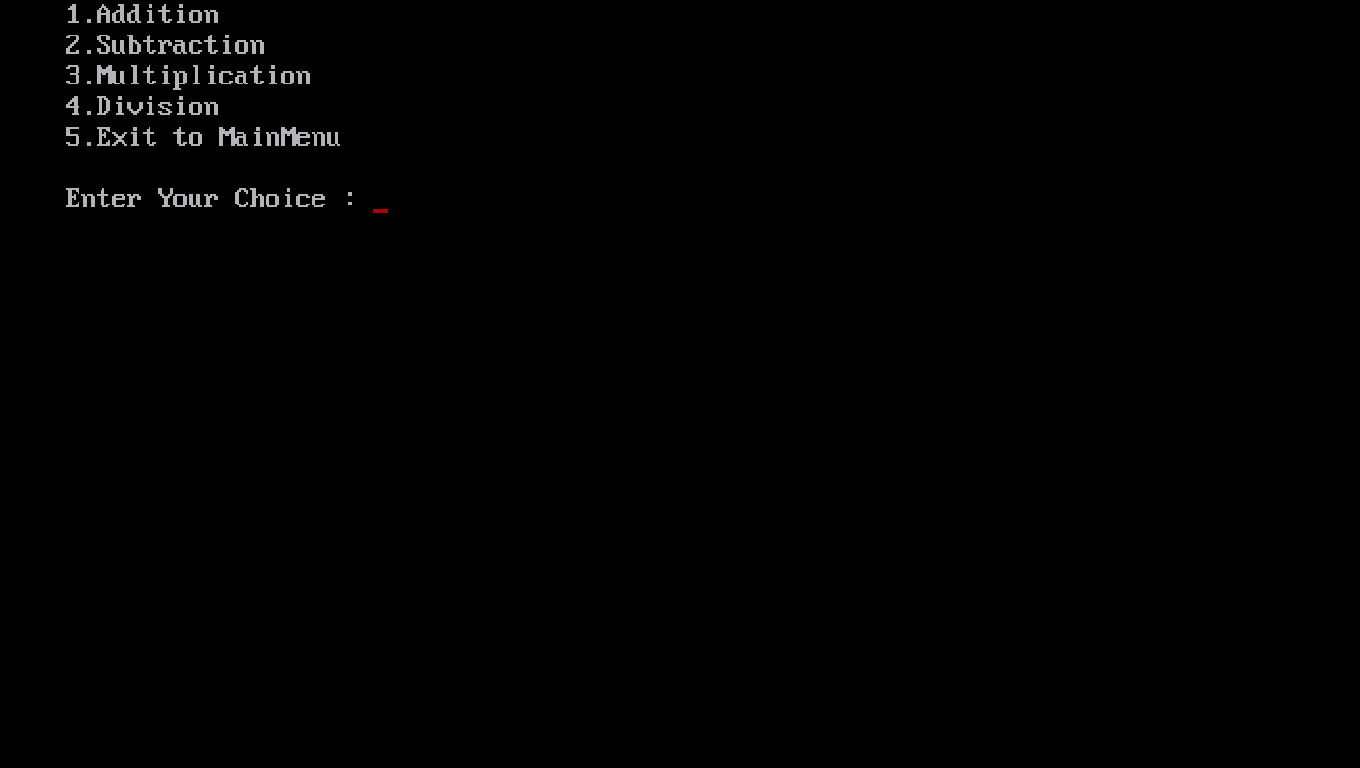
**You don’t have a login account**



**Login id already exists**



**Calculator**



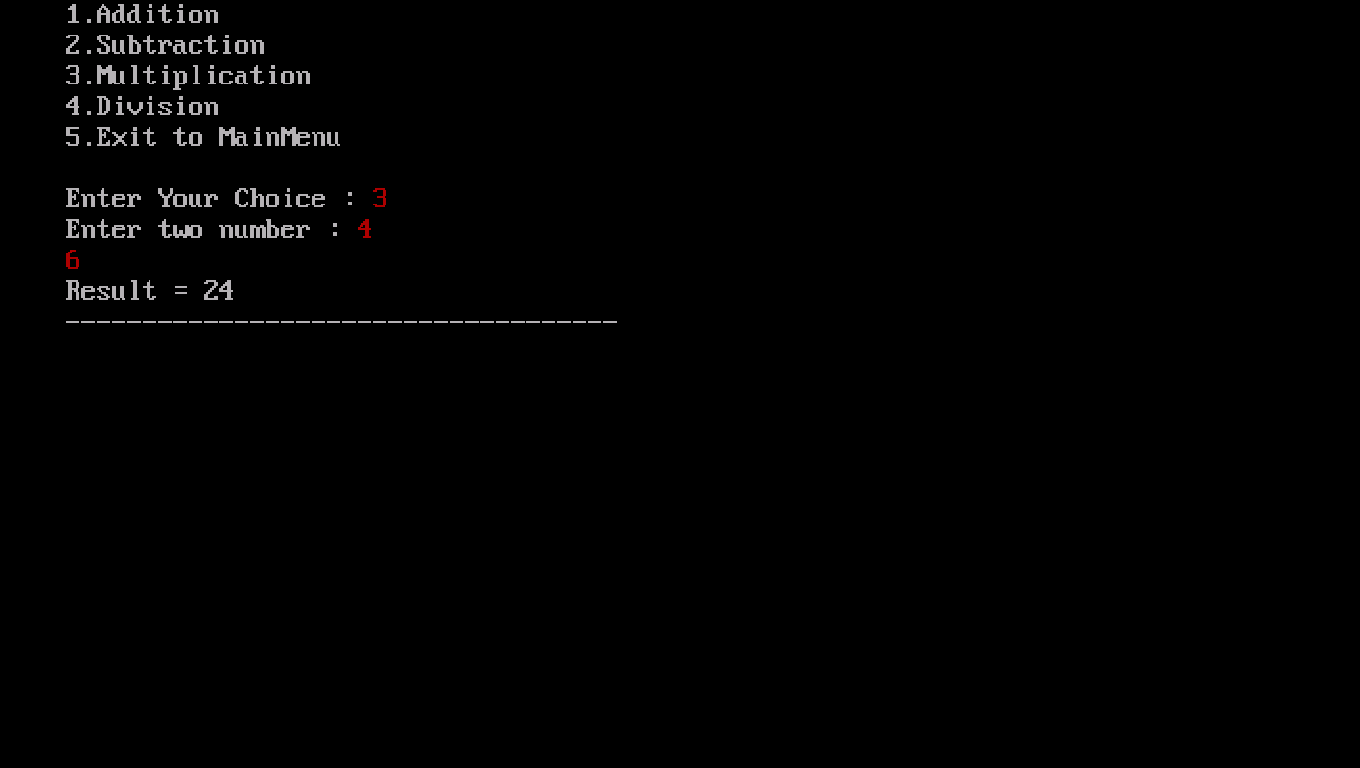
**Addition**



**Subtraction**



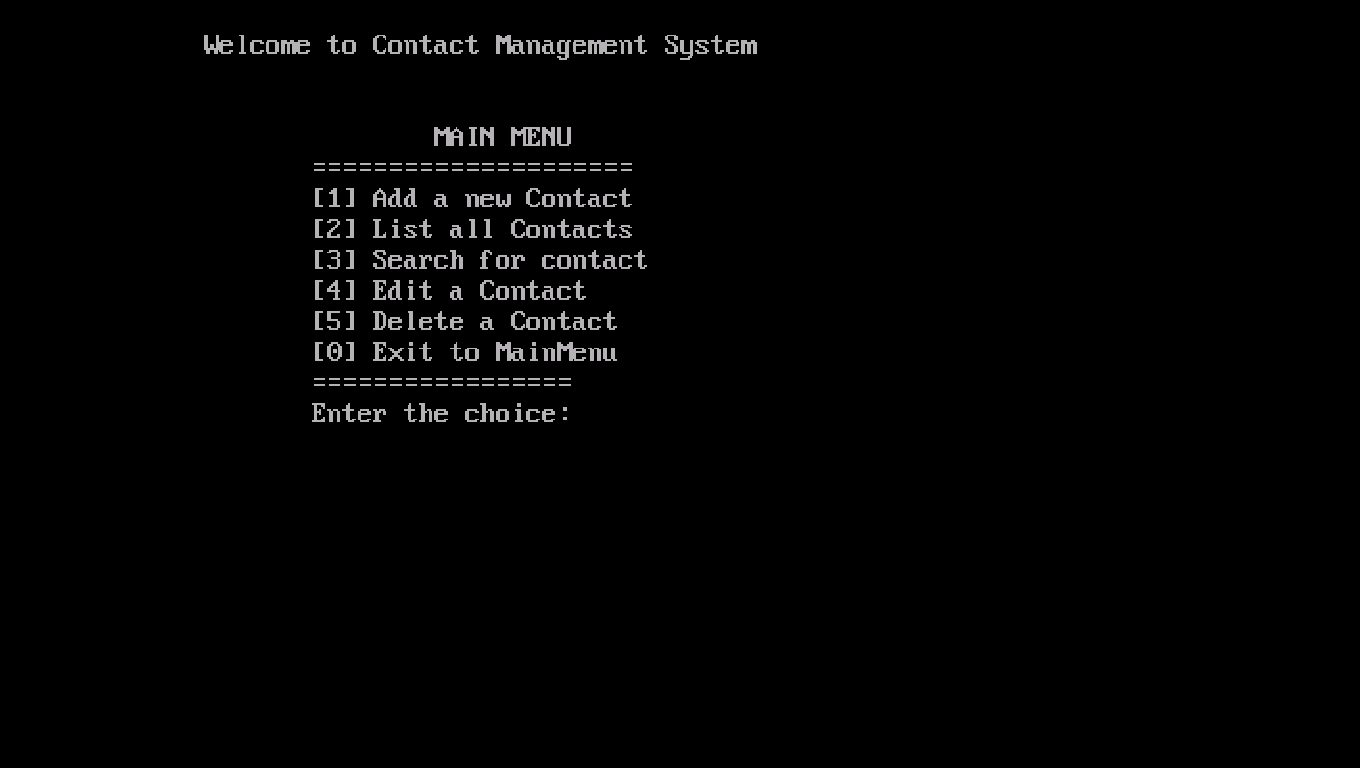
**Multiplication**



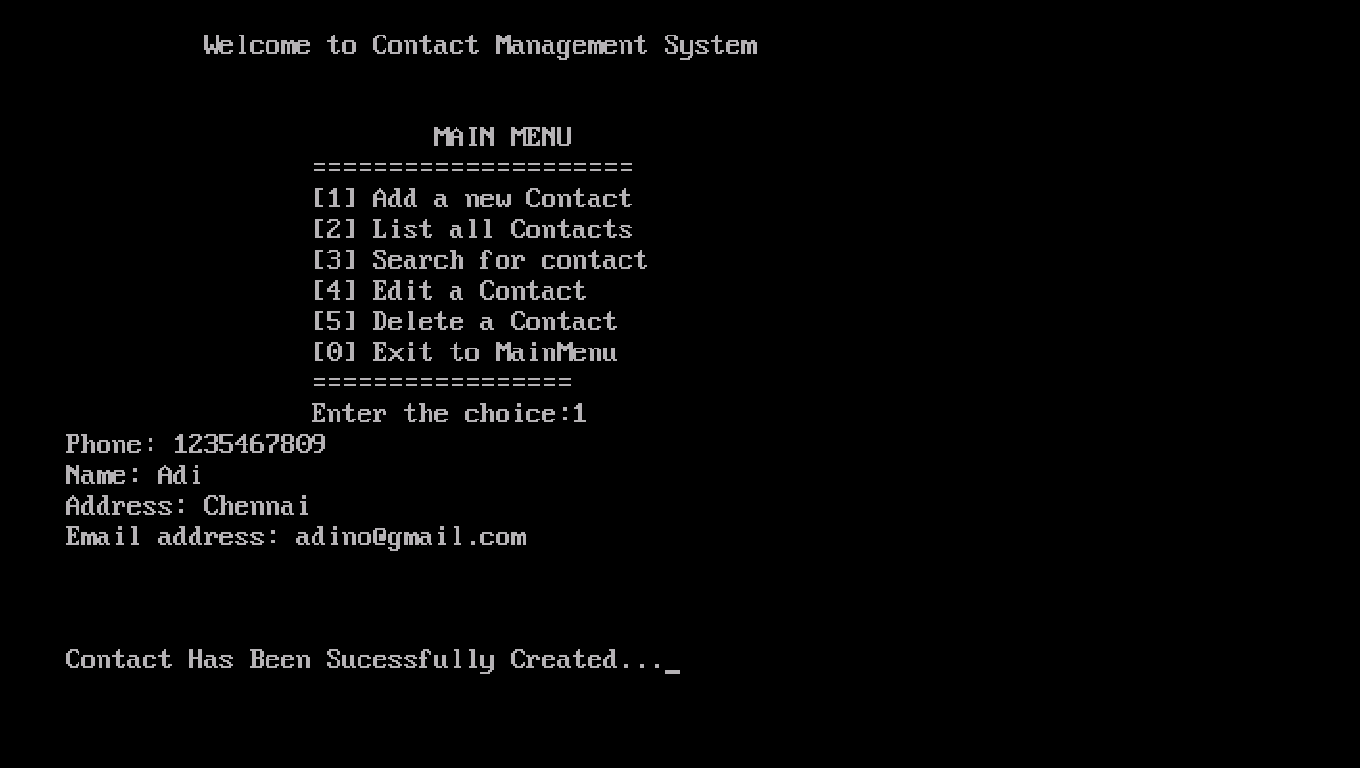
**Division**



**Contact Manager**



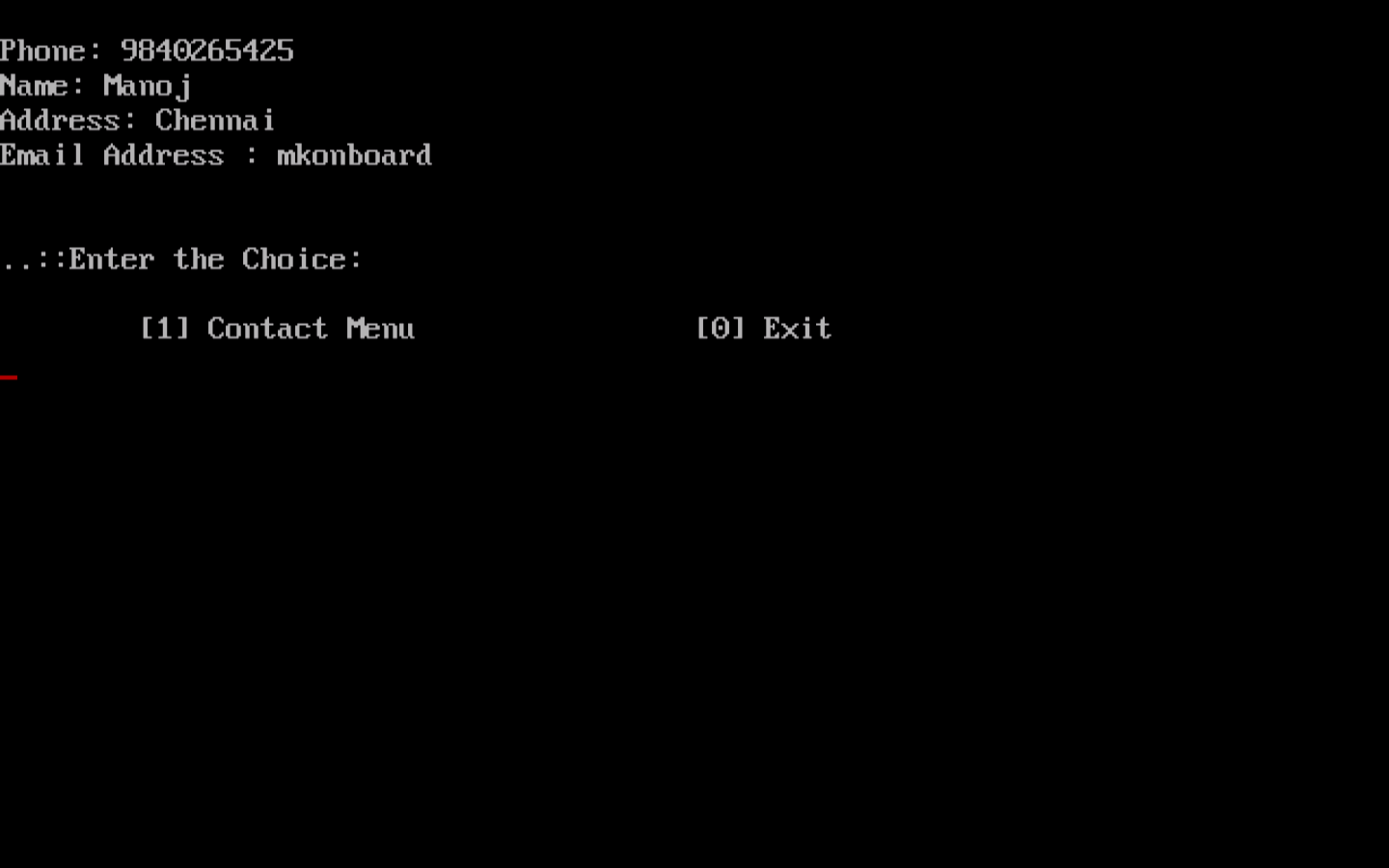
**Add a new contact**



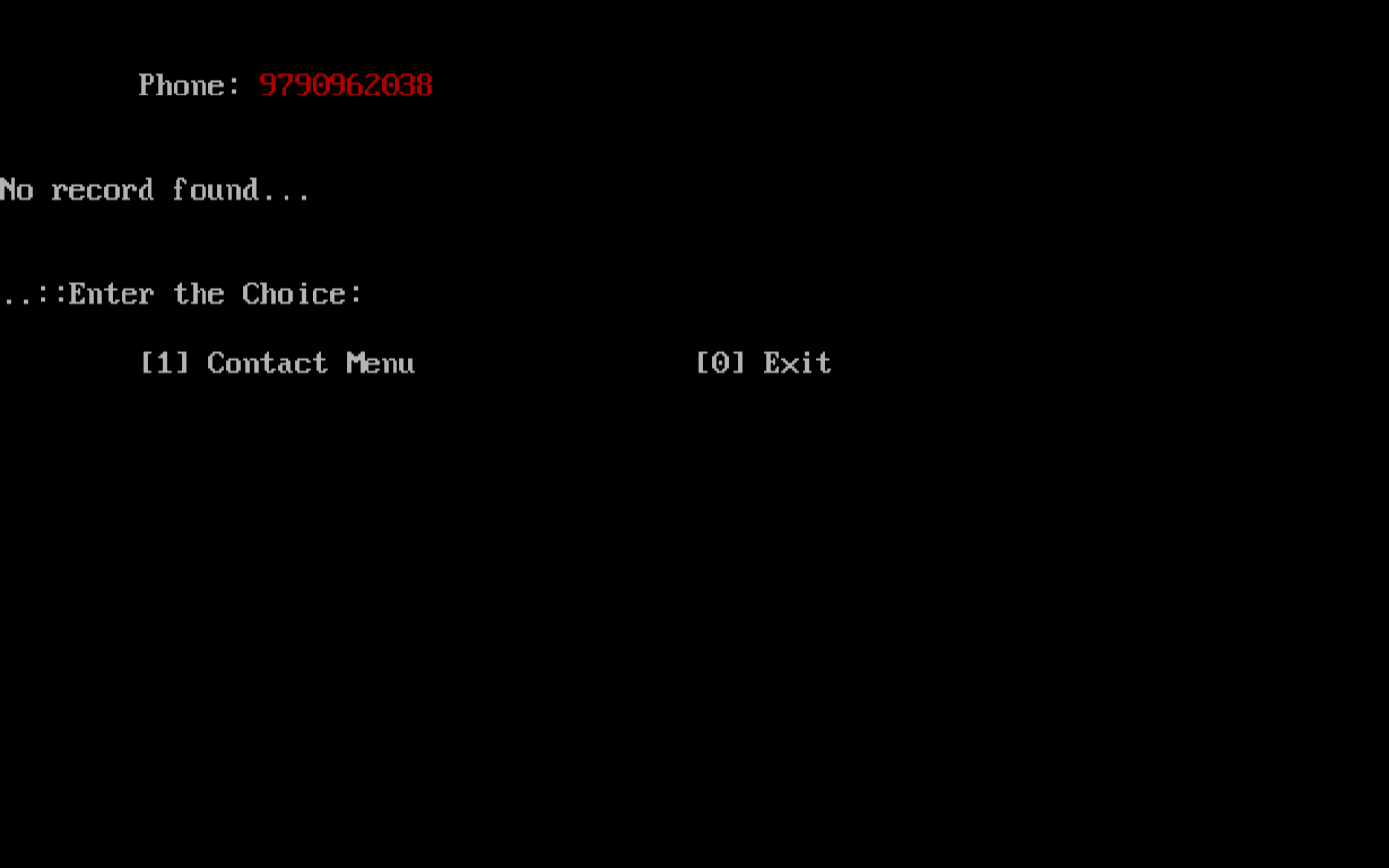
**Print all contacts**

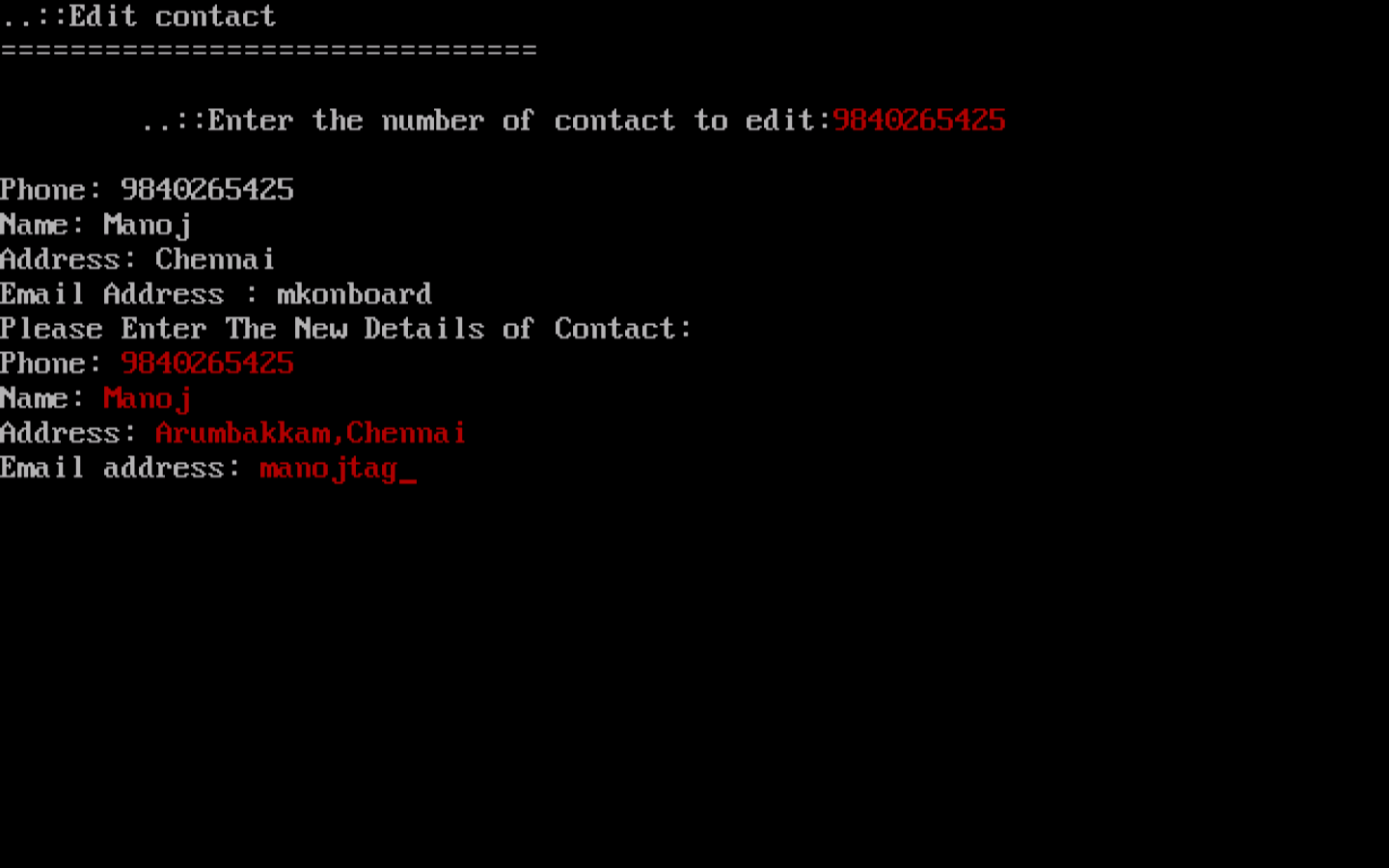


**Search for contacts**

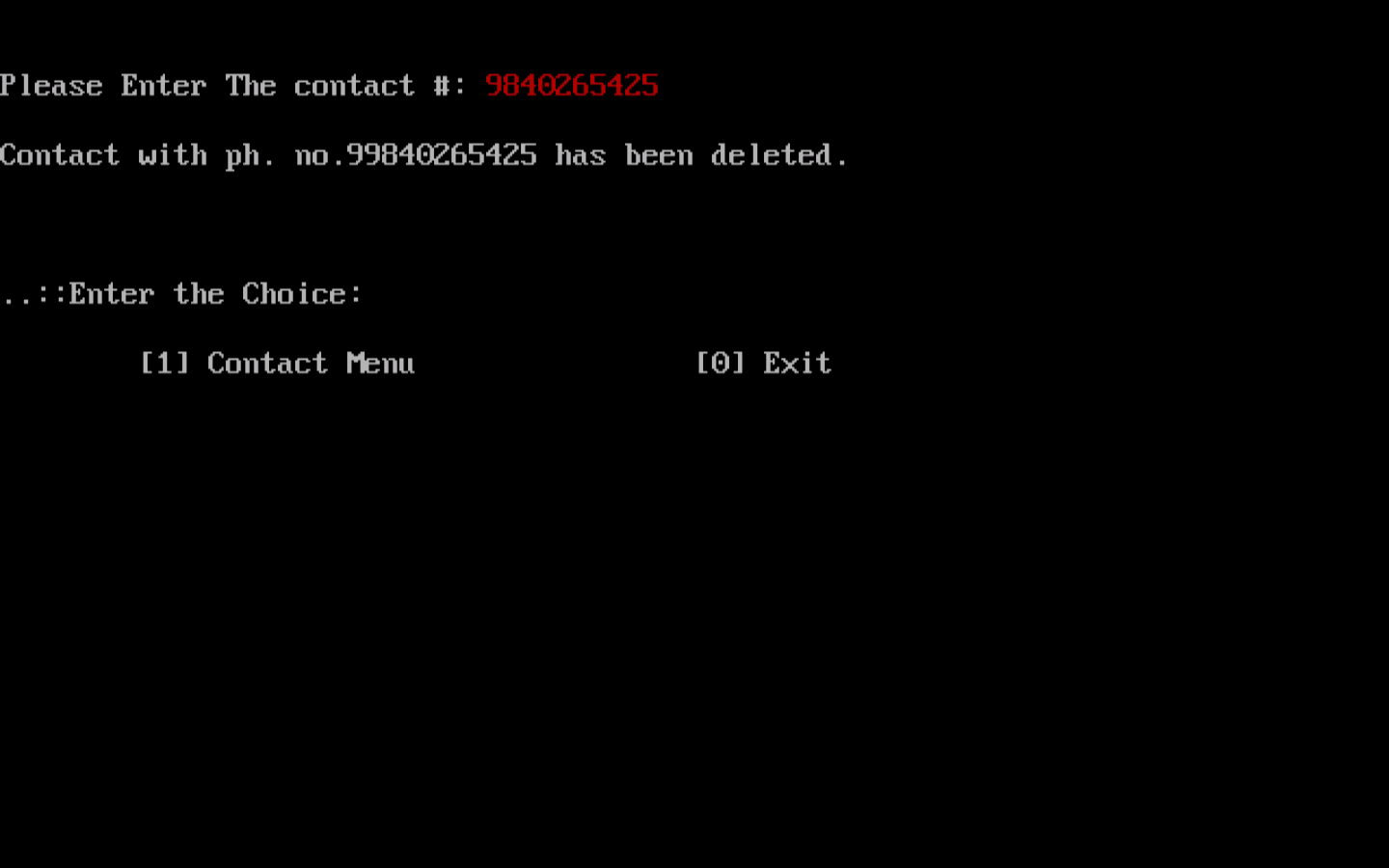
****

**Contact searched do not exist**

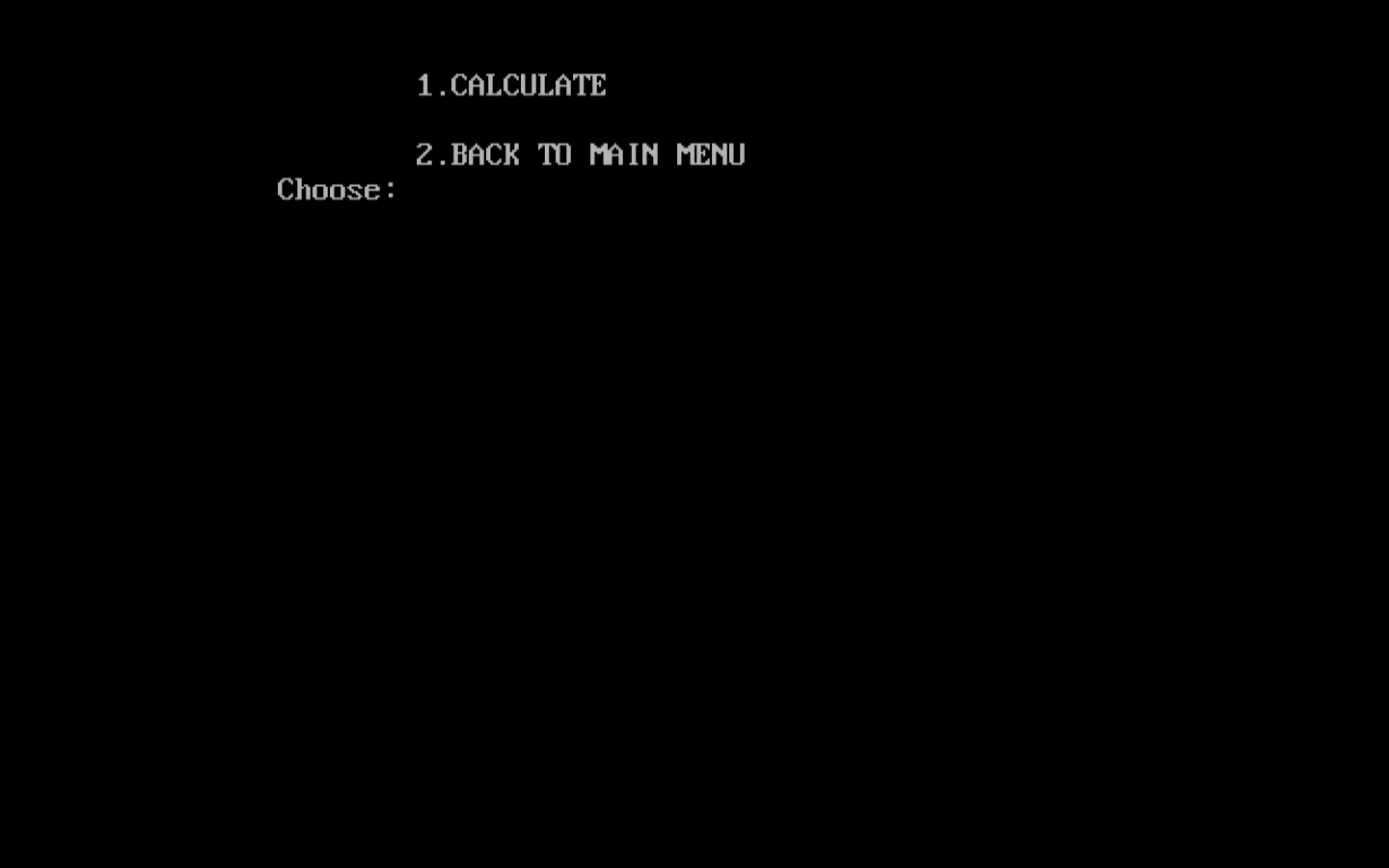
****

**Editing a contact**

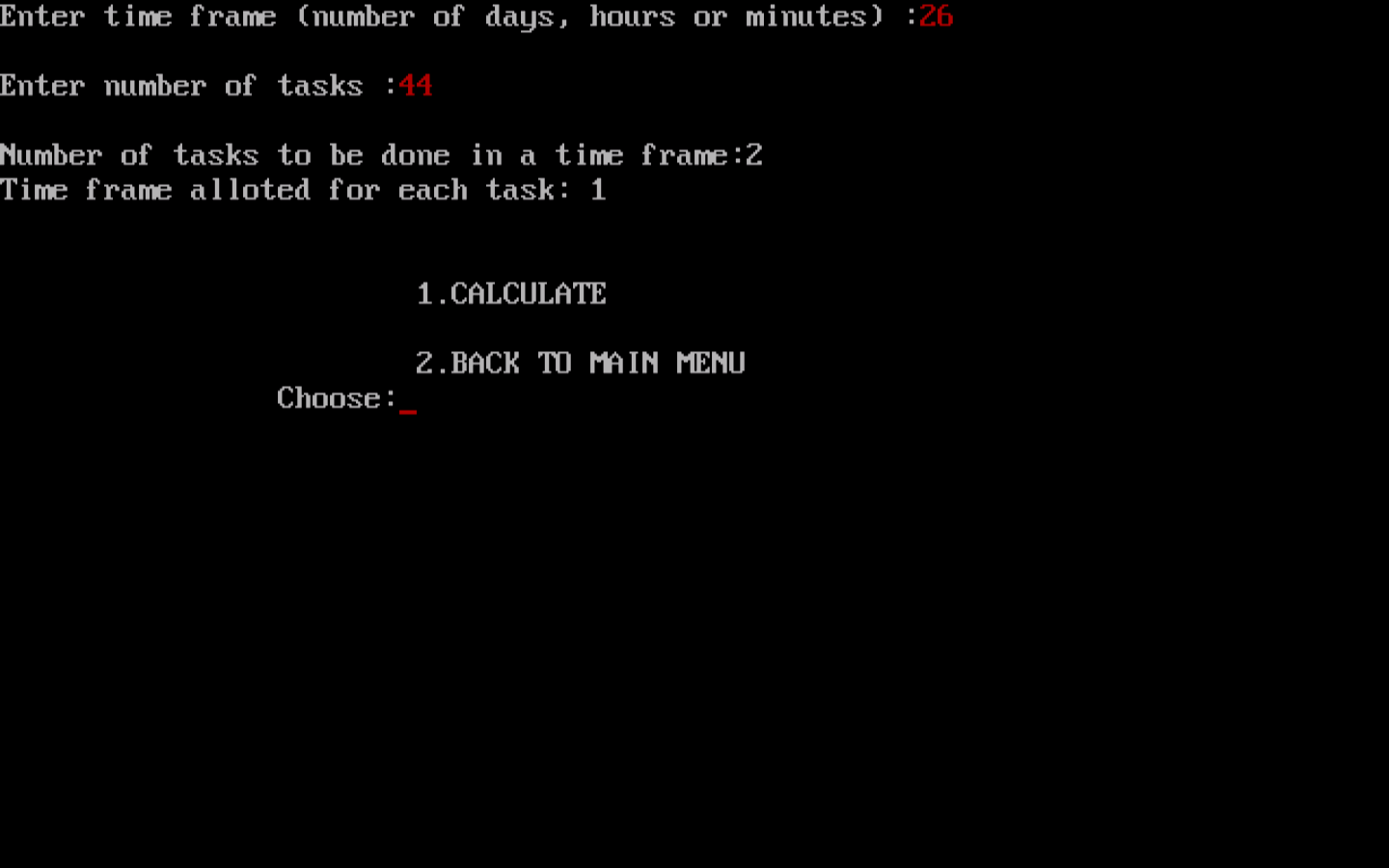
**Deleting a contact**

****

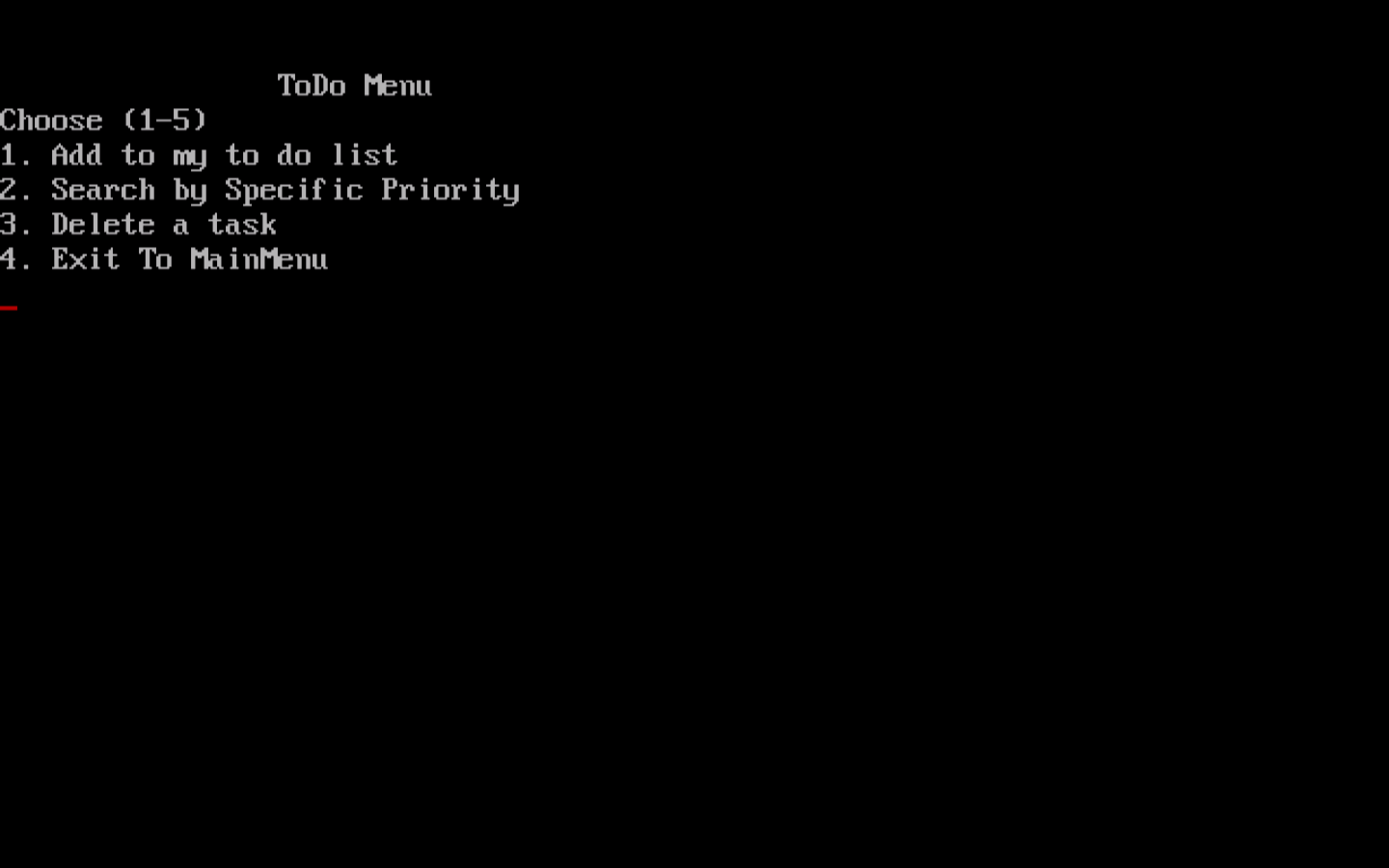
**Jobcal Menu**

****

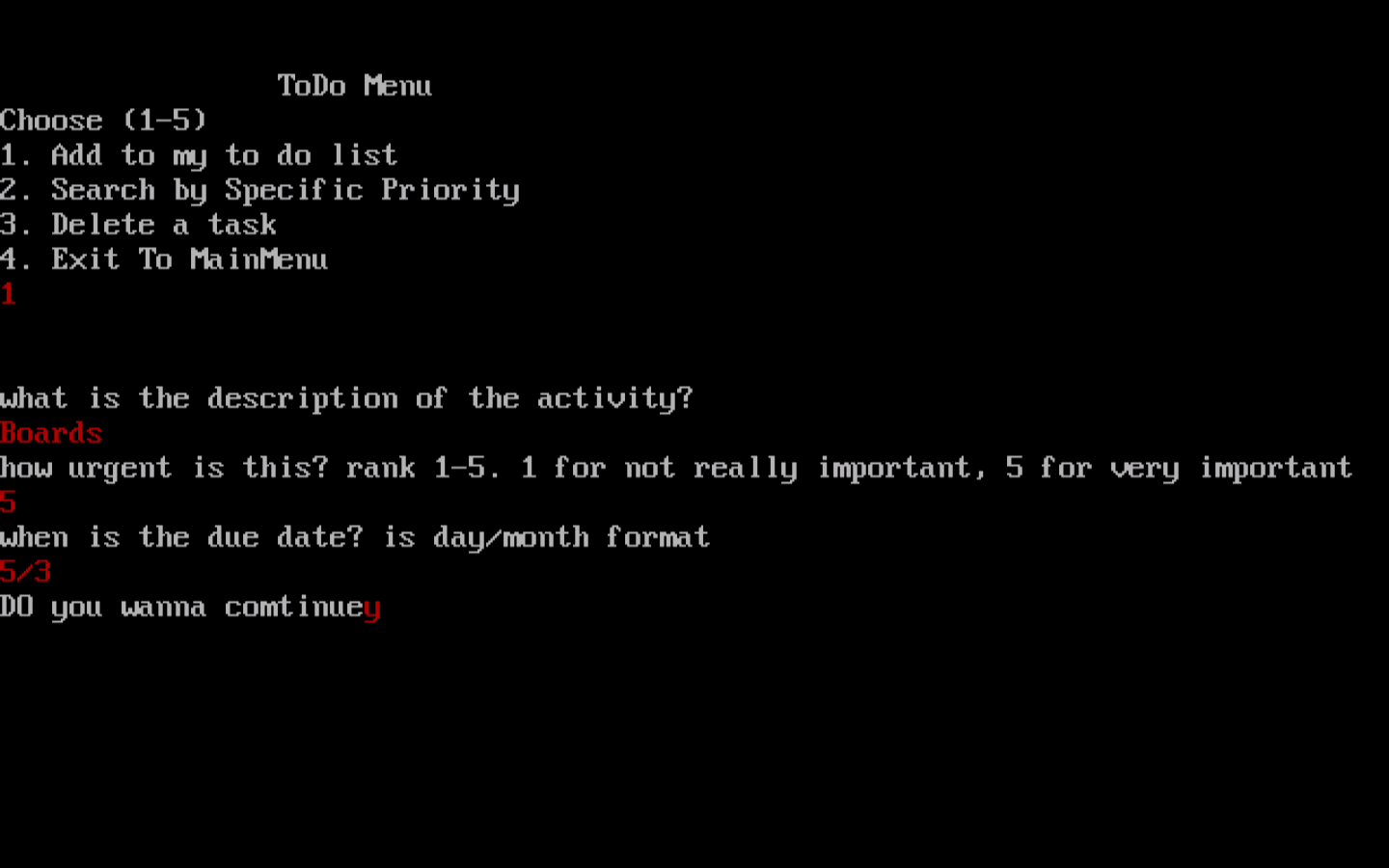
**Calculating tasks**

****

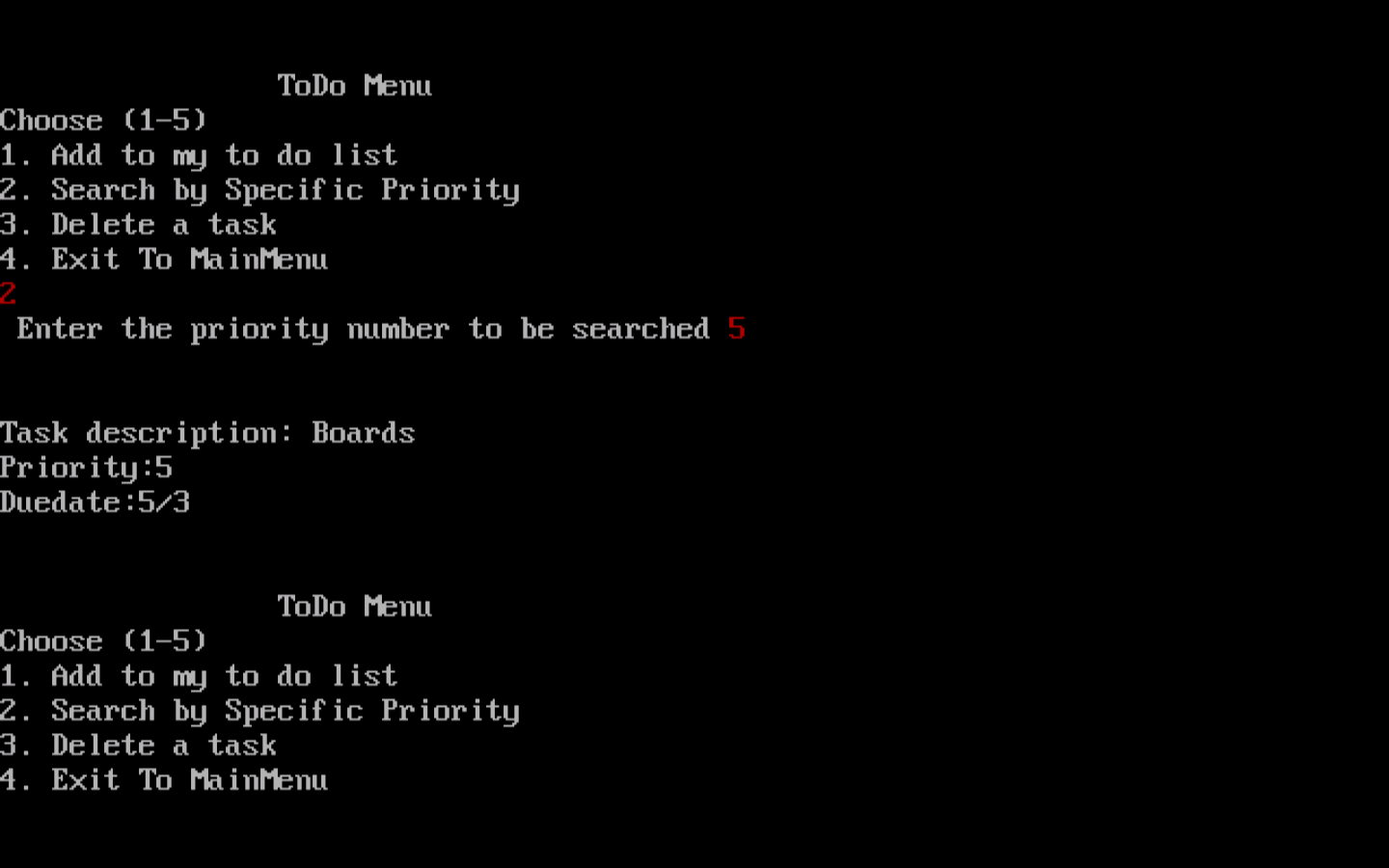
**To-do Menu**

****

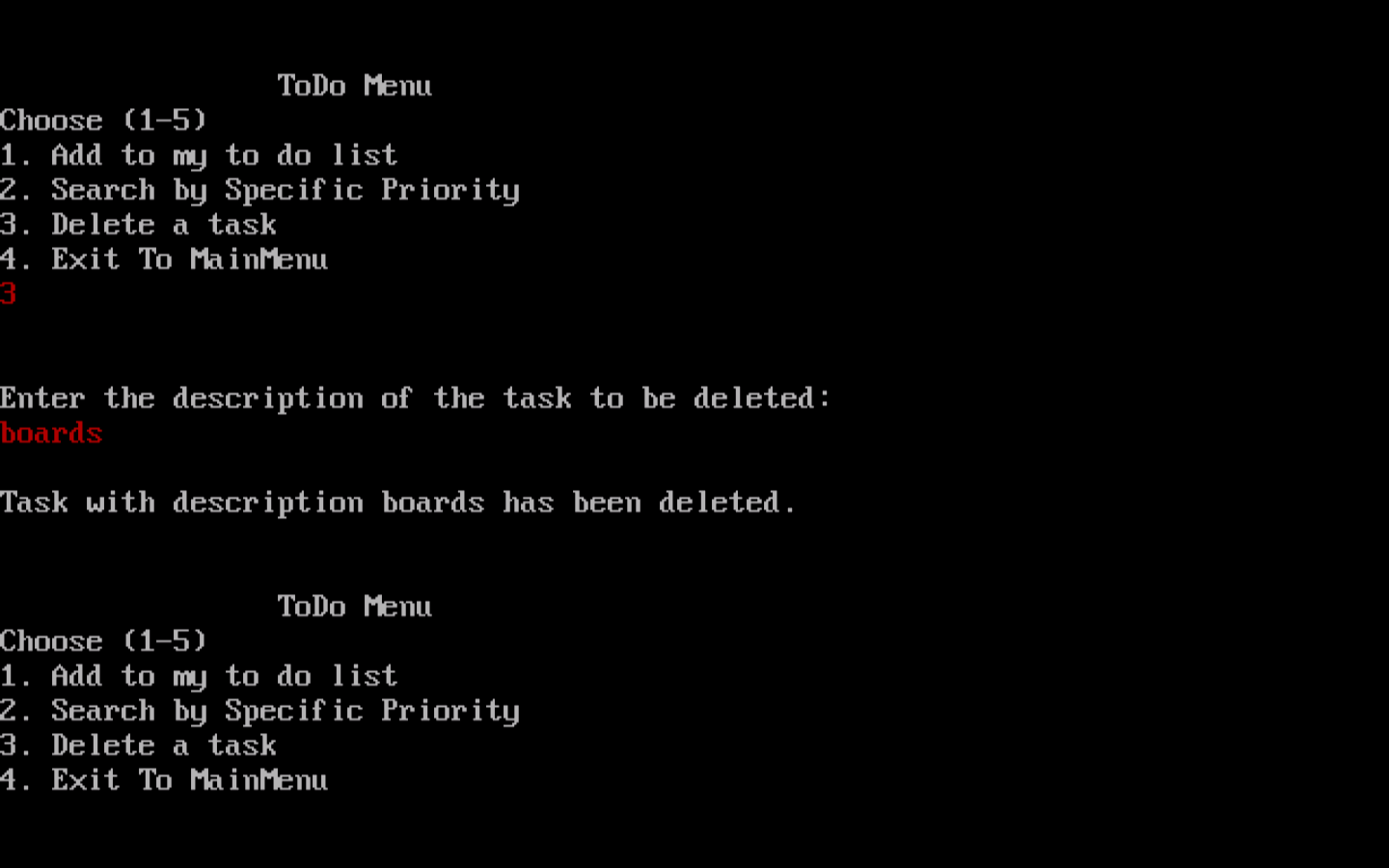
**Adding a task**

****

**Searching a task by its priority**

****

**Deleting the task**

****