**1. Creating a database for secret agents:**

#include<fstream.h>

#include<conio.h>

#include<stdio.h>

#include<iomanip.h>

#include<string.h>

#include<process.h>

class agent //creating class

{

int code;

char name[11], pass[16], details[500], specialisation[21];

public:

void nishant();

void shownishant();

void nitish();

void shownitish();

void recruit();

void show();

void modify();

void showall();

int showcode() //inline function reuturning code

{

return code;

}

char \*showname() //inline function returning name

{

return name;

}

char \*showpass() //inline function returning password

{

return pass;

}

};

//special functions for creators of this program to facilitate no modification on their part by user

void agent::nishant()

{

code=1;

strcpy(name,"nishant");

strcpy(pass,"kalita");

strcpy(specialisation, "ammunition expert");

}

void agent::shownishant()

{

cout<<"\ncoded "<<code<<", agent "<<name<<" specialises in S.S.A.S.A. as "<<specialisation;

cout<<"\n details of agent are (press any key to check details):";

getch();

for(int i=9999;i>0; i--)

cout<<i;

clrscr();

cout<<"currently acts as the COMMANDER of S.S.A.S.A., i.e. head of the agency!";

cout<<"\nhe excells in strength and his guts is all it is needed to lead the team\n\nNOTABLE INCIDENTS: ";

cout<<"\n\t\t\*MISSION COBRA\*\n ";

cout<<"in his furious fight against TERROR COMMANDER, 'COBRA GOD',\nnishant's team under ex-general, ZOD";

cout<<"had succesfully apprehended\nCOBRA but nishant had to take a bullet to his head...";

cout<<"\nthis had left agent nishant paralyesd for several months\nbut later it showed miracles...";

cout<<"\nafter his recovery he was found to have excellent intellegenge\nalthough sadly, ";

cout<<"he is still known to have some level of MENTAL DISORDED....";

}

void agent::nitish()

{

code=2;

strcpy(name,"nitish");

strcpy(pass,"kashyap");

strcpy(specialisation, "sniper");

}

void agent::shownitish()

{

cout<<"\ncoded "<<code<<", agent "<<name<<" specialises in S.S.A.S.A as "<<specialisation;

cout<<"\n details of agent are (press any key to check details):";

getch();

for(int i=9999;i>0; i--)

cout<<i;

clrscr();

cout<<"currently acts as the STRATEGICAL CHIEF of S.S.A.S.A.";

cout<<"\nhe excels in speed and can analyse and coordinate any situation \n\nNOTABLE INCIDENTS: ";

cout<<"\n\t\t\*MISSION MOON\*\n ";

cout<<"\nCLASSIFIED INFORMATION: when S.S.A.S.A. sattelite detected alien aircraft 256UFO";

cout<<"crash land on LUNAR surface\nat 00:00 hours, S.S.A.S.A. launched a MISSION MOON ";

cout<<"under ex-commander ZEODIN\nto check on it, agent nitish was also inculded in ZEODIN's team \n";

cout<<"before contact with lunar surface, due to reasons unknown till now,\nspacecraft lost all connections ";

cout<<"\nafter a time of 6 days, 15 hours, spacecraft had been detected in\nearth atmosphere";

cout<<"\nall the crewmembers were found dead with the exception of agent nitish";

cout<<"\nhe has no memory of what had happened with them";

cout<<"\nalthough, it was seen that all of agent nitish's fats were sucked off his body";

cout<<"\nthus, he still has a LEAN AND THIN body";

}//end of special functions for creators

void agent::recruit() //class's function to assign initial values

{

int x, y=0;

while(y==0)

{

cout<<"\nEnter agent code (non zero and not 1 or 2): ";

cin>>x;

if(x==1||x==2 || x==0)

cout<<"\ninvalid entry. Enter code again: ";

else

{

code=x;

y=1;

}

}

cout<<"\nEnter name of agent (maximum 10 characters): ";

gets(name);

cout<<"\nEnter agent specialisation (maximum 20 characters): ";

gets(specialisation);

cout<<"\nEnter agent details (press '|' when done): \n";

cin.getline(details, 200, '|');

cout<<"\nEnter the unique password the agent will use (maximum 15 characters): ";

gets(pass);

}

void agent::show() //class's function to display details of members

{

cout<<"\ncoded "<<code<<", agent "<<name<<" specialises in S.S.A.S.A. as "<<specialisation;

cout<<"\n details of agent are (press any key to check details):";

getch();

for(int i=9999;i>0; i--)

cout<<i;

clrscr();

cout<<details;

}

void agent::modify() //classes function to modify existing members

{

cout<<"\nagent name and password can not be changed";

int x, y=0;

while(y==0)

{

cout<<"\nEnter agent's new code (non zero and not 1 or 2): ";

cin>>x;

if(x==1||x==2 || x==0)

cout<<"\ninvalid entry. Enter code again: ";

else

{

code=x;

y=1;

}

}

cout<<"\nenter agent's new specialisation (maximum 20 characters): ";

gets(specialisation);

cout<<"\nEnter agent's new details (press '|' when done): \n";

cin.getline(details, 200, '|');

}

void agent::showall() //class's function to show list of all members

{

cout<<code<<setw(15)<<" "<<name<<setw(15)<<" "<<specialisation<<endl;

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

}//end of class and its function definations

//start of functions that will be summoned my a function 'menu'

void add()

{

agent x;

ofstream f;

f.open("agent.dat",ios::binary|ios::app);

x.recruit();

f.write((char \*) &x, sizeof(agent));

f.close();

cout<<"\nAgent added ";

cout<<"\npress any key to continue: ";

getch();

}

void info(int n)

{

agent x;

int found=0;

ifstream f;

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

if(!f)

{

cout<<"File could not be open !! Press any Key...";

return;

}

cout<<"\nAGENT DETAILS\n";

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

{

if(x.showcode()==n)

{

if(n==1)

x.shownishant();

else if(n==2)

x.shownitish();

else

x.show();

found=1;

}

}

f.close();

if(found==0)

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

cout<<"\npress any key to continue ";

getch();

}

void modify(int n)

{

int found=0;

agent x;

fstream f;

f.open("agent.dat",ios::binary|ios::in|ios::out);

if(!f)

{

cout<<"Agent not found !! Press any Key...";

return;

}

while(f.read((char \*) &x, sizeof(agent))&& found==0)

{

if(x.showcode()==n)

{

if(n==1)

cout<<"\nagent nishant's data cannot be modified as he is creator of program: ";

else if(n==2)

cout<<"\nagent nitish's data cannot be modified as he is creator of program: ";

else

{

cout<<"\n\nEnter The New Details of agent"<<endl;

x.modify();

int position=(-1)\*sizeof(agent);

f.seekp(position,ios::cur);

f.write((char \*) &x, sizeof(agent));

cout<<"\n\n\t Agent Info Updated";

}

found=1;

}

}

f.close();

if(found==0)

cout<<"\n\n Agent not Found ";

cout<<"\npress any key to continue: ";

getch();

}

void expel(int n)

{

agent x;

ifstream fi;

ofstream fo;

fi.open("agent.dat",ios::binary);

if(!fi)

{

cout<<"File could not be open !! Press any Key...";

return;

}

fo.open("temp.dat",ios::binary);

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

while(fi.read((char \*) &x, sizeof(agent)))

{

if(x.showcode()!=n)

{

fo.write((char \*) &x, sizeof(agent));

}

}

fi.close();

fo.close();

remove("agent.dat");

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

}

void all()

{

agent x;

ifstream f;

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

if(!f)

{

cout<<"File could not be open !! Press any Key...";

return;

}

cout<<"\n\n\t\tSSASA AGENT LIST\n\n";

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<"Agent Code. Name Specialisation\n";

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

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

{

x.showall();

}

f.close();

cout<<"\n\npress any key to continue: ";

getch();

}

void help()

{

cout<<"\nhello! this database is created by NISHANT KALITA and NITISH KASHYAP";

cout<<"\nit is easy to use, first, u must have accessed with the correct password";

cout<<"\nu have in total 3 chances to input correct password and name";

cout<<"\n(note: each name has unique password)";

cout<<"\nafter which u encounter a menu";

cout<<"\nthe menu provides 7 functions as described in the menu";

cout<<"\nu must also notice that agent nitish and agent nishant can not be expelled\nand their";

cout<<" data can not be modified as they are the creators of this program";

cout<<"\nthank you for using this program";

cout<<"\n\npress any key to continue ";

getch();

}

//end of functions that will be called by 'menu' function

void initial() //a function to assign creators directly when program starts

{

agent x;

ofstream f;

f.open("agent.dat", ios::binary|ios::app);

x.nishant();

f.write((char \*) &x, sizeof(agent));

x.nitish();

f.write((char \*) &x, sizeof(agent));

f.close();

}

void intro() //opening of project

{

cout<<"\n\n\twelcome to C++ program 'CREATING A DATABASE FOR SECRET AGENTS'\n\n";

cout<<"\t\tclass 12 project for computer science for the \n\t\tpractical fulfilment of AISSCE,2014 in Science";

cout<<"\n\n\n\t\tcreated by, NISHANT KALITA and NITISH KASHYAP";

cout<<"\n\npress any key to continue: ";

getch();

}

void menu(char \*z) //main function called 'menu' which acts as doorway to other working functions

{

int n, m;

char w;

do

{

clrscr();

cout<<"\twelcome agent "<<z;

cout<<"\n\n\nwhat would you like to do? (select option from menu) ";

cout<<"\n\n\t1. Info on agents\n\n\t2. Add new agent\n\n\t3. Modify agent info\n\n\t4. Expel agent";

cout<<"\n\n\t5. Show all agents\n\n\t6. Help\n\n\t7. Exit"<<endl;

cin>>m;

clrscr();

switch(m)

{

case 1: cout<<"\nenter agent code: ";

cin>>n;

info(n);

break;

case 2: add();

break;

case 3: cout<<"\nenter code of agent to be modified: ";

cin>>n;

modify(n);

break;

case 4: cout<<"\nenter code of agent to be expeled: ";

cin>>n;

if(n==1)

{

cout<<"\nagent nishant cannot be expeled as he is creator of program: ";

getch();

}

else if(n==2)

{

cout<<"\nagent nitish cannot be expeled as he is creator of program: ";

getch();

}

else

{

cout<<"\nare u sure? (y/n)";

cin>>w;

if(w=='y')

{

expel(n);

cout<<"\n\n\tAgent Has Been Removed ...";

getch();

}

}

break;

case 5: all();

break;

case 6: help();

break;

case 7:

clrscr();

cout<<"\n\n\n\t\t\tEXITTING DATABASE";

getch();

expel(1);//two function call deleting both creators (as they will be initialised in next program run)

expel(2);

exit(0); //exitting the wole program

default:

cout<<"\nincorrect selection: ";

break;

}

}while(m!=7);

}

void main()

{

int i=0, entry=0;

char n[11], p[16];

clrscr();

intro(); //opening page

initial(); //initialisation of the creators of the program

while(i<3 && entry==0) //a loop to check match of password and name of agent giving three chances

{

clrscr();

cout<<"================================================================================";

cout<<"\n\t\tWELCOME TO SSA SECRET AGENCY (S.S.A.S.A) DATABASE";

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

cout<<"\n\n\t\t\tenter agent name: ";

gets(n);

cout<<"\n\t\t\tenter unique password: ";

gets(p);

agent x;

ifstream f;

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

clrscr();

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

{

if((strcmp(x.showname(),n)==0) && (strcmp(x.showpass(),p)==0))

{

entry=1;

menu(n);

}

}

if(entry!=1)

{

cout<<"\n\t\tmismatch.. you have "<<2-i<<" try(s) left";

cout<<"\npress any key";

getch();

}

f.close();

i++;

}

if(entry==0) //if unable to input matching set of name and passwords

{

cout<<"\n\n\t\t\tINTRUDER ALERT";

cout<<"\nprogram closing.. any press of key will shut down program";

getch();

for(int j=9999;j>0; j--)

cout<<j;

clrscr();

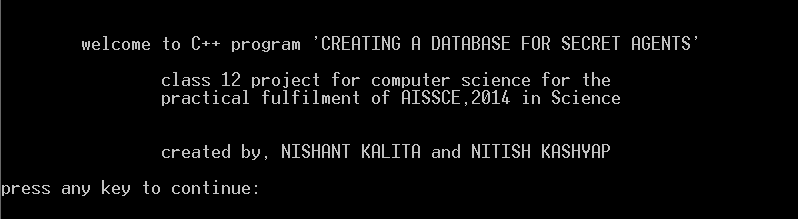
expel(1); //functions similar to ones used when exiting from ‘menu’ function

expel(2);

}

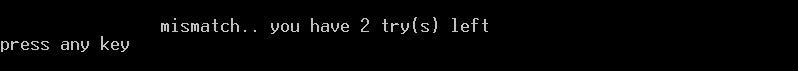
}

**Output:**



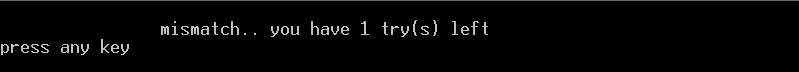
----------------------------------------------------screen cleared--------------------------------------------------------------

If wrong name and password used:

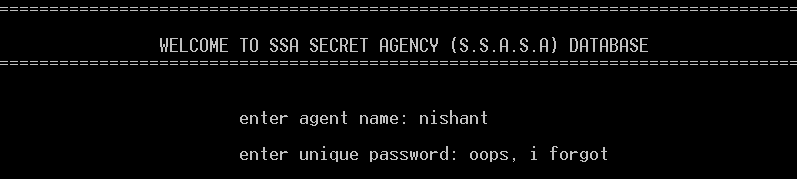
----------------------------------------------------screen cleared--------------------------------------------------------------

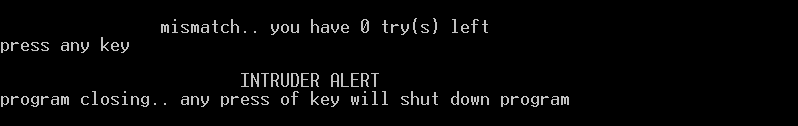
----------------------------------------------------screen cleared--------------------------------------------------------------

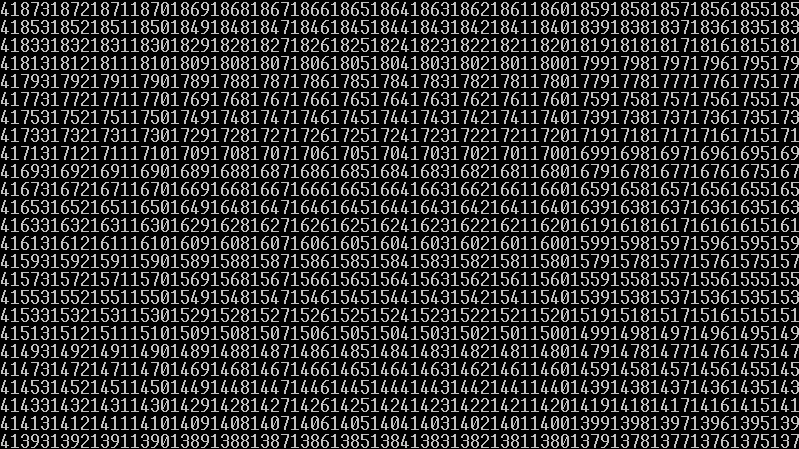
Another mismatch

----------------------------------------------------screen cleared--------------------------------------------------------------

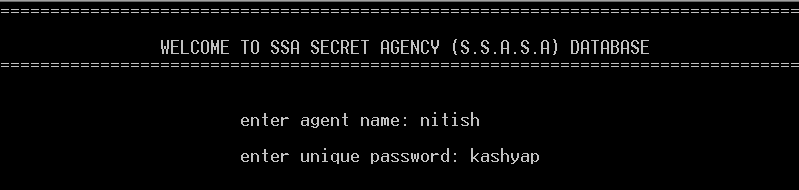
----------------------------------------------------screen cleared--------------------------------------------------------------

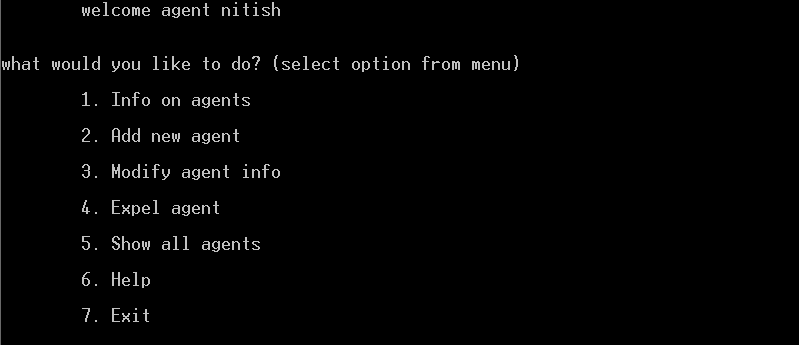
Another wrong entry

----------------------------------------------------screen cleared--------------------------------------------------------------

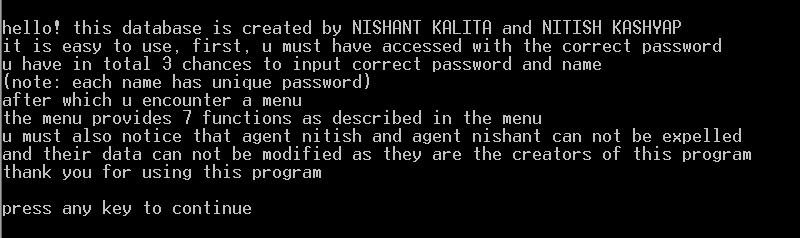
After press of a key

And program ends

If suppose correct and matching name and password:

----------------------------------------------------screen cleared--------------------------------------------------------------

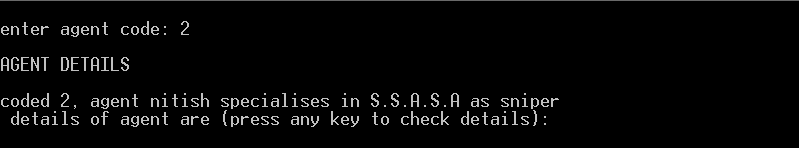
Suppose 6 is entered

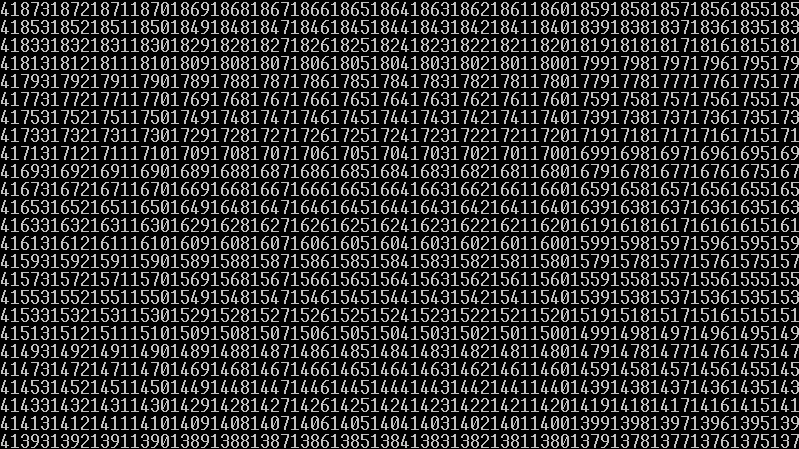
----------------------------------------------------screen cleared--------------------------------------------------------------

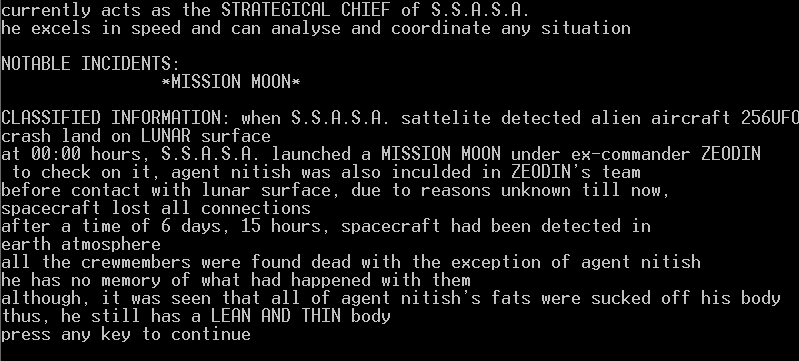
----------------------------------------------------screen cleared--------------------------------------------------------------

After press of any key, the menu is displayed again

If ‘1’ is pressed

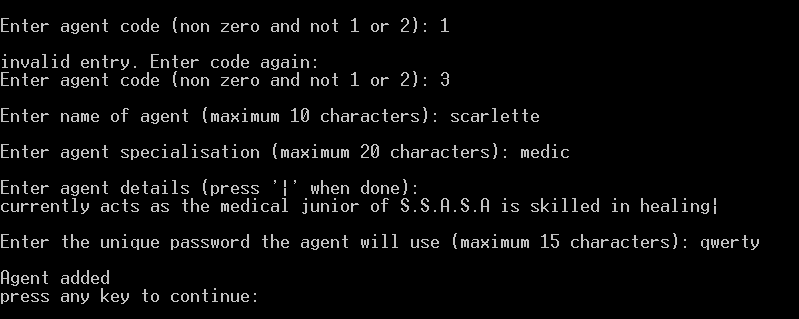


After press of any key

----------------------------------------------------screen cleared--------------------------------------------------------------

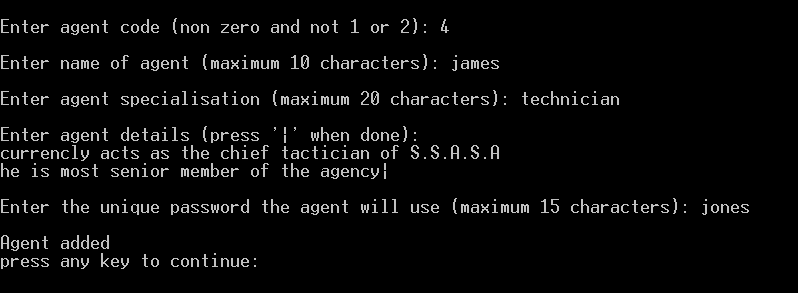
----------------------------------------------------screen cleared--------------------------------------------------------------

It returns to the menu

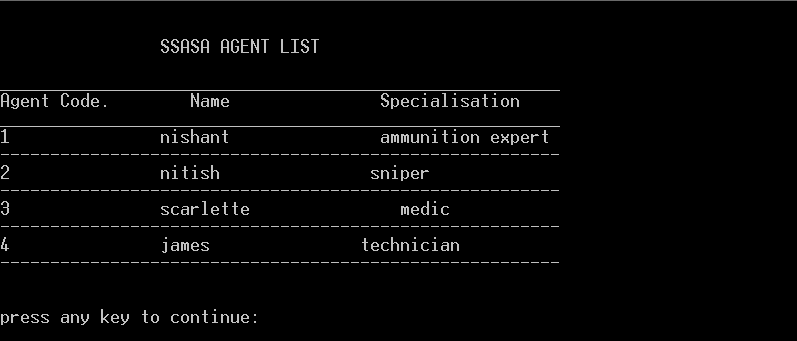
If ‘2’ is pressed

----------------------------------------------------screen cleared--------------------------------------------------------------

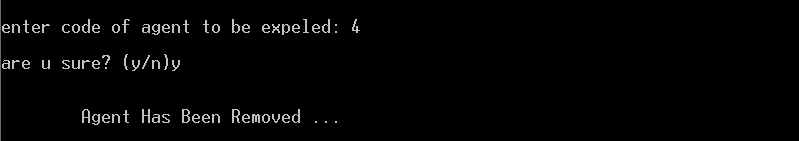
Again in menu if ‘2’ is pressed



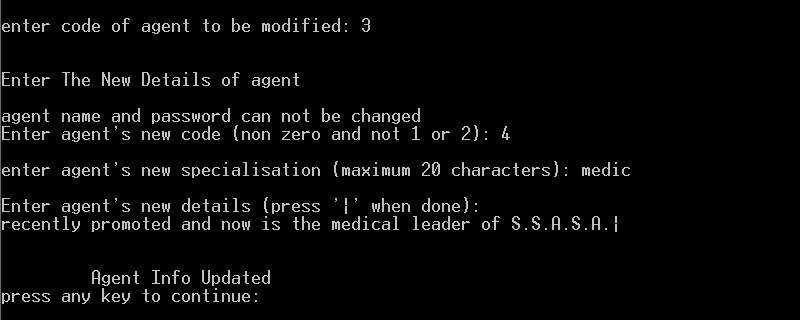
----------------------------------------------------screen cleared--------------------------------------------------------------

If in menu, ‘5’ is pressed

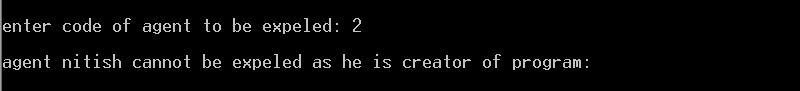
----------------------------------------------------screen cleared--------------------------------------------------------------

If in menu, ‘4’ is pressed

----------------------------------------------------screen cleared--------------------------------------------------------------

If in menu, now ‘3’ is pressed

----------------------------------------------------screen cleared--------------------------------------------------------------

If in menu ‘4’ is pressed

----------------------------------------------------screen cleared--------------------------------------------------------------

Returns to menu, if ‘7’ is pressed

Note that next time when program is opened, agent 3’s name and password can also be used