#include <bits/stdc++.h>

using namespace std;

class bill

{

public:

int day,month,year,amount;

string description,status;

bill(const int &a,const int &b,const int &c,const int &d,const string &s,const string &st="paid")

{

day=a;

month=b;

year=c;

amount=d;

description=s;

status=st;

}

};

class transcmp

{

public:

bool operator()(const bill &A,const bill &B)

{

if(A.year!=B.year)

return A.year<B.year;

else if(A.month!=B.month)

return A.month<B.month;

else

return A.day<B.day;

}

};

int random()

{

time\_t t;

srand((unsigned)time(&t));

return (abs(rand())%9999)+10000;

}

class patient

{

public:

int id;

string name;

string problems;

int age;

long long phone\_no;

multiset<bill,transcmp> payments;

patient()

{

}

patient(const string &name,const string &problems,const long long \*a,const vector<bill> &v)

{

this->name=name;

this->problems=problems;

id=a[0];

age=a[1];

phone\_no=a[2];

for(auto b:v)

{

payments.insert(b);

}

}

void initpatient()

{

cout<<"enter name : \n";

getline(cin,name);

cout<<"enter health issues in a single line with commas \n";

getline(cin,problems);

cout<<"enter age : ";

cin>>age;

cout<<"enter phone number : ";

cin>>phone\_no;

id=random();

cout<<"your user id is "<<id<<endl;

}

void output()

{

cout<<"\nuser id: "<<id;

cout<<"\nname : "<<name;

cout<<"\nage : "<<age;

cout<<"\nphone number : "<<phone\_no;

cout<<"\nhealth issues : "<<problems<<endl;

if(!payments.empty())

{

cout<<"\nPayments\n\n";

for(bill b:payments)

{

const char \*c1=b.status.c\_str();

const char \*c2=b.description.c\_str();

//cout<<b.day<<"/"<<b.month<<"/"<<b.year<<"\t"<<b.amount<<"\t"<<b.description<<"\t\t"<<b.status<<endl;

printf("%d/%d/%d Rs%-10d %-32s%-10s\n",b.day,b.month,b.year,b.amount,c2,c1);

}

}

else

{

cout<<"No payments till yet, why so? \n";

}

}

};

list<patient> l;

unordered\_map<int,string> m;

list<string> \*graph;

vector<string> symptoms={"cold","cough","fever","headache","stomachache","bodyache","nausea","vomiting","diarrhea","blood in sputum"};

void init\_graph()

{

//string symptoms[]={"cold","cough","fever","headache","stomachache","bodyache","nausea","vomiting","diarrhea","blood in sputum","blood in vomit","loss of consciousness","trouble in breathing","fatigue","muscle aches","palpitations","tastelessness","dehydration","dizziness","weakness","loss of appetite"};

vector<string> v1={"blood in vomit","loss of consciousness","trouble in breathing","fatigue","muscle aches","palpitations","tastelessness","dehydration","dizziness","weakness","loss of appetite"};

symptoms.insert(symptoms.end(),v1.begin(),v1.end());

vector<string> v2={"hallucinations","paralysis","constipation","weight loss","skin rash"};

symptoms.insert(symptoms.end(),v2.begin(),v2.end());

const int n=symptoms.size();

/\*

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

{

//shift this in main menu. initialize graph only once.

cout<<i<<". "<<symptoms[i]<<endl;

}

\*/

graph=new list<string>[n];

graph[18].push\_back("rabies");

graph[2].push\_back("rabies");

graph[3].push\_back("rabies");

graph[13].push\_back("rabies");

graph[14].push\_back("rabies");

graph[6].push\_back("rabies");

graph[7].push\_back("rabies");

graph[20].push\_back("rabies");

graph[21].push\_back("rabies");

graph[22].push\_back("rabies");

graph[2].push\_back("polio");

graph[3].push\_back("polio");

graph[11].push\_back("polio");

graph[13].push\_back("polio");

graph[19].push\_back("polio");

graph[22].push\_back("polio");

graph[1].push\_back("corona");

graph[2].push\_back("corona");

graph[3].push\_back("corona");

graph[8].push\_back("corona");

graph[12].push\_back("corona");

graph[13].push\_back("corona");

graph[16].push\_back("corona");

graph[19].push\_back("corona");

graph[0].push\_back("corona");

graph[2].push\_back("dengue");

graph[3].push\_back("dengue");

graph[5].push\_back("dengue");

graph[6].push\_back("dengue");

graph[7].push\_back("dengue");

graph[13].push\_back("dengue");

graph[14].push\_back("dengue");

graph[20].push\_back("dengue");

graph[0].push\_back("malaria");

graph[2].push\_back("malaria");

graph[3].push\_back("malaria");

graph[4].push\_back("malaria");

graph[6].push\_back("malaria");

graph[7].push\_back("malaria");

graph[8].push\_back("malaria");

graph[13].push\_back("malaria");

graph[14].push\_back("malaria");

graph[0].push\_back("typhoid");

graph[2].push\_back("typhoid");

graph[3].push\_back("typhoid");

graph[4].push\_back("typhoid");

graph[6].push\_back("typhoid");

graph[7].push\_back("typhoid");

graph[8].push\_back("typhoid");

graph[13].push\_back("typhoid");

graph[14].push\_back("typhoid");

graph[19].push\_back("typhoid");

graph[20].push\_back("typhoid");

graph[23].push\_back("typhoid");

graph[24].push\_back("typhoid");

graph[25].push\_back("typhoid");

graph[1].push\_back("tuberculosis");

graph[2].push\_back("tuberculosis");

graph[5].push\_back("tuberculosis");

graph[9].push\_back("tuberculosis");

graph[12].push\_back("tuberculosis");

graph[13].push\_back("tuberculosis");

graph[20].push\_back("tuberculosis");

graph[24].push\_back("tuberculosis");

graph[2].push\_back("gastrointestinal disease");

graph[4].push\_back("gastrointestinal disease");

graph[6].push\_back("gastrointestinal disease");

graph[7].push\_back("gastrointestinal disease");

graph[8].push\_back("gastrointestinal disease");

graph[10].push\_back("gastrointestinal disease");

graph[20].push\_back("gastrointestinal disease");

graph[23].push\_back("gastrointestinal disease");

graph[24].push\_back("gastrointestinal disease");

graph[2].push\_back("food poisoning");

graph[3].push\_back("food poisoning");

graph[4].push\_back("food poisoning");

graph[6].push\_back("food poisoning");

graph[7].push\_back("food poisoning");

graph[8].push\_back("food poisoning");

graph[13].push\_back("food poisoning");

graph[17].push\_back("food poisoning");

graph[18].push\_back("food poisoning");

graph[19].push\_back("food poisoning");

graph[20].push\_back("food poisoning");

graph[0].push\_back("sinus");

graph[2].push\_back("sinus");

graph[3].push\_back("sinus");

graph[12].push\_back("sinus");

graph[13].push\_back("sinus");

graph[2].push\_back("chickenpox");

graph[3].push\_back("chickenpox");

graph[13].push\_back("chickenpox");

graph[20].push\_back("chickenpox");

graph[25].push\_back("chickenpox");

graph[4].push\_back("cirrhosis");

graph[6].push\_back("cirrhosis");

graph[7].push\_back("cirrhosis");

graph[12].push\_back("cirrhosis");

graph[13].push\_back("cirrhosis");

graph[10].push\_back("cirrhosis");

graph[19].push\_back("cirrhosis");

graph[20].push\_back("cirrhosis");

graph[24].push\_back("cirrhosis");

graph[2].push\_back("pancreatitis");

graph[4].push\_back("pancreatitis");

graph[6].push\_back("pancreatitis");

graph[7].push\_back("pancreatitis");

graph[8].push\_back("pancreatitis");

graph[15].push\_back("pancreatitis");

graph[20].push\_back("pancreatitis");

graph[24].push\_back("pancreatitis");

graph[2].push\_back("appendicitis");

graph[4].push\_back("appendicitis");

graph[6].push\_back("appendicitis");

graph[7].push\_back("appendicitis");

graph[8].push\_back("appendicitis");

graph[20].push\_back("appendicitis");

graph[23].push\_back("appendicitis");

graph[5].push\_back("arthritis");

graph[13].push\_back("arthritis");

graph[14].push\_back("arthritis");

graph[19].push\_back("arthritis");

graph[22].push\_back("arthritis");

graph[1].push\_back("asthma");

graph[5].push\_back("asthma");

graph[12].push\_back("asthma");

graph[15].push\_back("asthma");

graph[0].push\_back("bronchitis");

graph[1].push\_back("bronchitis");

graph[2].push\_back("bronchitis");

graph[3].push\_back("bronchitis");

graph[12].push\_back("bronchitis");

graph[13].push\_back("bronchitis");

graph[3].push\_back("brain tumor");

graph[6].push\_back("brain tumor");

graph[7].push\_back("brain tumor");

graph[11].push\_back("brain tumor");

graph[13].push\_back("brain tumor");

graph[18].push\_back("brain tumor");

graph[21].push\_back("brain tumor");

graph[22].push\_back("brain tumor");

graph[13].push\_back("type 1 diabetes");

graph[24].push\_back("type 1 diabetes");

graph[2].push\_back("ebola");

graph[3].push\_back("ebola");

graph[4].push\_back("ebola");

graph[5].push\_back("ebola");

graph[6].push\_back("ebola");

graph[7].push\_back("ebola");

graph[8].push\_back("ebola");

graph[9].push\_back("ebola");

graph[10].push\_back("ebola");

graph[13].push\_back("ebola");

graph[14].push\_back("ebola");

graph[17].push\_back("ebola");

graph[19].push\_back("ebola");

graph[20].push\_back("ebola");

graph[25].push\_back("ebola");

//20 till this line

graph[2].push\_back("gallstones");

graph[4].push\_back("gallstones");

graph[7].push\_back("gallstones");

graph[6].push\_back("gallstones");

graph[8].push\_back("gallstones");

graph[15].push\_back("gallstones");

graph[20].push\_back("gallstones");

graph[2].push\_back("kidney stones");

graph[4].push\_back("kidney stones");

graph[6].push\_back("kidney stones");

graph[5].push\_back("kidney stones");

graph[7].push\_back("kidney stones");

graph[2].push\_back("stomach flu");

graph[3].push\_back("stomach flu");

graph[4].push\_back("stomach flu");

graph[6].push\_back("stomach flu");

graph[7].push\_back("stomach flu");

graph[8].push\_back("stomach flu");

graph[13].push\_back("stomach flu");

graph[14].push\_back("stomach flu");

graph[15].push\_back("stomach flu");

graph[17].push\_back("stomach flu");

graph[18].push\_back("stomach flu");

graph[19].push\_back("stomach flu");

graph[20].push\_back("stomach flu");

graph[24].push\_back("stomach flu");

graph[6].push\_back("low blood pressure");

graph[11].push\_back("low blood pressure");

graph[13].push\_back("low blood pressure");

graph[15].push\_back("low blood pressure");

graph[17].push\_back("low blood pressure");

graph[18].push\_back("low blood pressure");

graph[2].push\_back("hepatitis A");

graph[4].push\_back("hepatitis A");

graph[6].push\_back("hepatitis A");

graph[8].push\_back("hepatitis A");

graph[13].push\_back("hepatitis A");

graph[14].push\_back("hepatitis A");

graph[20].push\_back("hepatitis A");

graph[23].push\_back("hepatitis A");

graph[24].push\_back("hepatitis A");

graph[7].push\_back("hepatitis A");

graph[7].push\_back("hepatitis B");

graph[2].push\_back("hepatitis B");

graph[4].push\_back("hepatitis B");

graph[6].push\_back("hepatitis B");

graph[13].push\_back("hepatitis B");

graph[19].push\_back("hepatitis B");

graph[20].push\_back("hepatitis B");

graph[2].push\_back("hepatitis C");

graph[4].push\_back("hepatitis C");

graph[6].push\_back("hepatitis C");

graph[5].push\_back("hepatitis C");

graph[13].push\_back("hepatitis C");

graph[14].push\_back("hepatitis C");

graph[20].push\_back("hepatitis C");

graph[24].push\_back("hepatitis C");

graph[0].push\_back("HIV");

graph[1].push\_back("HIV");

graph[2].push\_back("HIV");

graph[3].push\_back("HIV");

graph[6].push\_back("HIV");

graph[7].push\_back("HIV");

graph[8].push\_back("HIV");

graph[13].push\_back("HIV");

graph[14].push\_back("HIV");

graph[20].push\_back("HIV");

graph[24].push\_back("HIV");

graph[25].push\_back("HIV");

graph[3].push\_back("anemia");

graph[12].push\_back("anemia");

graph[13].push\_back("anemia");

graph[15].push\_back("anemia");

graph[18].push\_back("anemia");

graph[20].push\_back("anemia");

graph[4].push\_back("high blood sugar");

graph[6].push\_back("high blood sugar");

graph[7].push\_back("high blood sugar");

graph[12].push\_back("high blood sugar");

graph[13].push\_back("high blood sugar");

graph[15].push\_back("high blood sugar");

graph[6].push\_back("low blood sugar");

graph[11].push\_back("low blood sugar");

graph[13].push\_back("low blood sugar");

graph[18].push\_back("low blood sugar");

graph[19].push\_back("low blood sugar");

graph[2].push\_back("kidney infection");

graph[8].push\_back("kidney infection");

graph[13].push\_back("kidney infection");

graph[19].push\_back("kidney infection");

graph[20].push\_back("kidney infection");

graph[1].push\_back("pneumonia");

graph[2].push\_back("pneumonia");

graph[3].push\_back("pneumonia");

graph[6].push\_back("pneumonia");

graph[7].push\_back("pneumonia");

graph[9].push\_back("pneumonia");

graph[12].push\_back("pneumonia");

graph[13].push\_back("pneumonia");

graph[14].push\_back("pneumonia");

graph[15].push\_back("pneumonia");

graph[20].push\_back("pneumonia");

}

void menu(patient &p)

{

system("cls");

int t;

do{

cout<<"Welcome "<<p.name<<". How can we serve you?\n\n";

cout<<"1. See your details and transactions\n2. Make an appointment for a checkup\n3. Make an appointment for a scan\n";

cout<<"4. Admit yourself in the nearest hospital\n5. Pay pending bills\n6. Change your password\n7. Logout\n8. Link your health insurance";

cout<<"\n9. Apply for government healthcare policies\n10. Feeling unwell? tell us the symptoms and we may be able to diagnose you\n";

cin>>t;

string s;

switch(t)

{

case 1:

{

p.output();

system("pause");

//menu(p);

system("cls");

break;

}

case 2:{

cout<<"Please state details of your discomfort the doctor will be allocated accordingly \n";

cin>>s;

cout<<"\nAn appointment has been made at Fortis Hospital on Friday at 3pm. Please make a payment of 500rs.\n ";

bill b1(13,10,2020,500,"appointment at Fortis Hospital","pending");

p.payments.insert(b1);

system("pause");

//menu(p);

system("cls");

break;

}

case 3:{

cout<<"\nWhat test do you want?\n ";

cin>>s;

cout<<"\nAn appointment has been made at Dr.Lal Path Labs on Friday at 3pm. Please make a payment of 800rs.\n ";

bill b2(13,10,2020,800,"test at Dr.Lal Path Labs","pending");

p.payments.insert(b2);

system("pause");

//menu(p);

system("cls");

break;

}

case 4:{

cout<<"\nPlease state details of your discomfort \n ";

cin>>s;

cout<<"\nA room has been allocated to you at Fortis hospital at 3000rs/day \n";

bill b3(13,10,2020,3000,"room at Fortis Hospital","pending");

p.payments.insert(b3);

system("pause");

system("cls");

//menu(p);

break;

}

case 5:{

auto b=p.payments.begin();

while(b!=p.payments.end())

{

if(b->status=="pending")

{

const char \*c1=b->status.c\_str();

const char \*c2=b->description.c\_str();

printf("%d/%d/%d Rs%-10d %-32s%-10s\n",b->day,b->month,b->year,b->amount,c2,c1);

bill temp(b->day,b->month,b->year,b->amount,c2,"paid");

p.payments.insert(temp);

b=p.payments.erase(b);

}

else

{

b++;

}

}

cout<<"\nAll bills have been paid Thank You.\n";

system("pause");

system("cls");

//menu(p);

break;

}

case 6:{

cout<<"\nPlease enter the old password\n";

cin.get();

getline(cin,s);

while(s!=m[p.id])

{

cout<<"\nwrong password try again\n";

getline(cin,s);

}

cout<<"\nPlease enter the new password\n";

getline(cin,s);

m[p.id]=s;

system("pause");

system("cls");

//menu(p);

break;

}

case 7:{

return;

break;

}

case 8:

{

cout<<"\nenter your insurance provider ";

cin>>s;

cout<<"enter policy number ";

int polno;

cin>>polno;

cout<<"Please enter the OTP sent to your registered mobile number ";

cin>>polno;

cout<<"The process has been initiated, it may take a little while to verify the details\n it is our pleasure to serve you\n";

system("pause");

system("cls");

break;

}

case 9:

{

cout<<"enter aadhar number ";

int ano;

cin>>ano;

cout<<"Please enter the OTP sent to your registered mobile number ";

cin>>ano;

cout<<"The process has been initiated, it may take a little while to verify the details\n it is our pleasure to serve you\n";

system("pause");

system("cls");

break;

}

case 10:

{

const int n=symptoms.size();

//loop for seeing the graph

/\*

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

{

cout<<i<<". "<<symptoms[i]<<" : ";

for(auto x:graph[i])

{

cout<<x<<",";

}

cout<<endl;

}

\*/

map<string,int> disfreq;

int ptr=0;

system("cls");

for(string sym:symptoms)

{

cout<<ptr<<". "<<symptoms[ptr]<<endl;

ptr++;

}

cout<<"please enter the indices of the symptoms you're facing and press -1 when done\n" ;

vector<int> sympsidx;

cin>>ptr;

while(ptr!=-1)

{

sympsidx.push\_back(ptr);

cin>>ptr;

}

int maxfreq=0;

for(int idx:sympsidx)

{

for(string disease:graph[idx])

{

disfreq[disease]++;

maxfreq=max(maxfreq,disfreq[disease]);

}

}

//cout<<maxfreq;

cout<<"\nMost likely:\n\n";

for(auto dispr:disfreq)

{

if(dispr.second==maxfreq)

{

cout<<dispr.first<<endl;

}

}

maxfreq--;

ptr=5;

if(maxfreq>0)

{

cout<<"\nless likely:\n\n";

for(auto dispr:disfreq)

{

if(dispr.second==maxfreq)

{

cout<<dispr.first<<endl;

ptr--;

if(ptr==0)

break;

}

}

}

cout<<"\nplease feel free to book an appointment through the main menu\n";

system("pause");

system("cls");

break;

}

}

}while(true);

}

void login()

{

system("cls");

cout<<"\t\t ONE NATION ONE HEALTH CARD\n1.Login\n2.Sign up\n";

int choice;

cin>>choice;

if(choice==1)

{

cout<<"enter user id ";

int x;

cin>>x;

cin.get();

cout<<"enter password\n";

string s;

getline(cin,s);

if(m.count(x)==0)

{

cout<<"No such user id in the database \n";

system("pause");

login();

}

else if(m[x]!=s)

{

cout<<"Wrong password \n";

system("pause");

login();

}

else

{

for(patient temp:l)

{

if(temp.id==x)

{

// temp.output();

menu(temp);

break;

}

}

}

}

else

{

patient newpatient;

cin.get();

newpatient.initpatient();

string s;

cin.get();

cout<<"Enter password \n";

getline(cin,s);

m[newpatient.id]=s;

l.push\_back(newpatient);

cout<<"You have been added to our database \n";

system("pause");

login();

}

}

void loaddata()

{

m[56237]="iostream";

m[83248]="iostream";

m[43678]="iostream";

bill b1(17,10,2015,332253,"heart bypass");

bill b2(17,11,2015,800,"full body checkup");

bill b3(22,10,2020,2300,"mri scan","pending");

vector<bill> v;

v.push\_back(b1);

v.push\_back(b2);

v.push\_back(b3);

long long \*a=new long long[3]{56237,72,9999567289};

patient p1("Ram Kishor","heart,brain",a,v);

l.push\_back(p1);

bill b12(10,10,2020,300,"X-Ray");

bill b22(12,10,2020,180600,"Leg surgery");

bill b32(19,10,2020,300,"X-Ray");

v.clear();

v.push\_back(b12);

v.push\_back(b22);

v.push\_back(b32);

delete []a;

a=new long long[3]{83248,24,9859567889};

patient p2("Karan Dua","Ligament tear in Leg",a,v);

l.push\_back(p2);

bill a12(5,6,2020,250000,"Corona treatment");

v.clear();

v.push\_back(a12);

delete []a;

a=new long long[3]{43678,57,9857943256};

patient p3("Rakesh Kumar","Corona",a,v);

l.push\_back(p3);

}

int main()

{

system("color 5F");

init\_graph();

loaddata();

login();

return 0;

}

/\*

for(bill b:p.payments)

{

const char \*c1=b.status.c\_str();

const char \*c2=b.description.c\_str();

if(b.status=="pending")

{

printf("%d/%d/%d Rs%-10d %-32s%-10s\n",b.day,b.month,b.year,b.amount,c2,c1);

b.status="paid";

}

}

\*/