**PROJECT REPORT**

****

CHEMISTRY LAB DATABASE

FATHER AGNEL SCHOOL

Board Roll NO:

INDEX

1. Certificate
2. Acknowledgement
3. Introduction
4. Requirements
5. Files used
6. Classes used
7. Non-Member Functions
8. Source Code
9. Output

**CERTIFICATE**

**Mrs. Kamaljeet Kaur**

**Computer Science Dept.**

# This is to certify that **IshaGautam** of class XII C has completed the project titled “**CHEMISTRY LAB DATABASE**” during the session 2016-17 under my guidance and supervision. .

Acknowledgement

I take this opportunity to express my sincere and heartfelt gratitude towards every person without whose help this project would have been a distant dream. Firstly, Ma’am Kamaljeet Kaur, whose timely advice and extreme useful tips made this project a lot easier than it would have been. Without her motivation and help the successful completion of this project would not have been possible.

I would like to thank Harsh and Ainika, my project partners. Without Harsh’sand Ainika’s organization and devotion this project never would have progressed as smoothly as it did.

I can’t forget to offer my sincere thanks to my classmates who helped me to carry out this project work successful and for their valuable advice.

Isha Gautam

INTRODUCTION

This project is aimed at designing an efficient system for management of chemical stock in the Chemistry Laboratory. This programme records new chemicals and its amount. Information about any chemical in stock can be searched. The record can be easily changed. The name of the manufacturer is also stored for future reference. This programme records the list of all the elements in the chemical stock in the Laboratory.

The programme also checks the availability of chemicals when the chemicals are being issued. The chemicals can also been returned anytime and the records are simultaneously updated. The last updated time is also given.

A unique feature of this programme is its notification feature which keeps a reminder amount of a chemical at the time of recording of new chemicals below which a notification is to be sent to the Lab Assistant or the person in-charge of maintaining the chemical stock. This featurehelps in replenishing the chemicals before they go out of stock. The last updated time also serves as a reminder to the Lab Assistant or the person in-charge to check the notification bar for shortage of chemicals.

This program has real-life implementations as well since it can be used in any laboratory to store records or in a pharmacy to store medicine stock.

The program can be modified to check for repeated entries and eliminate them. More features like real-time ordering of the chemicals in the notifications can be added which would enable the person using the program to replenish the required chemicals as soon as possible without much hassle.

Requirements

**Hardware Requirements:**

|  |  |
| --- | --- |
| RAM Requirements : | 512 MB (53,68,70,912 bytes) |
| Size on Hard Disk : | 80 KB (81,920 bytes) |
| Monitor Resolution : | 1366 x 768 |

**Software Requirements:**

|  |  |
| --- | --- |
| Operating System Used : | Windows 10 Home Single Language |
| Compiler : | Turbo C++ |

FILES USED

Two files named ‘Record.txt’ and ‘New.txt’ where used to implement the programme, the details of which are as follows:

**Record.txt**

This file stores ID, name, state, amount present in lab, minimum amount in the lab, the description of the named element or compound and also the name of the manufacturer.

**New.txt**

This file is used only in deletion of record (later re-named as ‘Record.txt’).

CLASSES USED

One class named “Chemical” was used to implement the program, the details of which are as follows:

**Chemical**

**PRIVATE SECTION**

**DATA MEMBERS:**

|  |  |
| --- | --- |
| **int id:** | Chemical ID (uniquely identifies a chemical) |
| **char name [400]:** | Name of the chemical |
| **char formula[100]:** | Formula of the chemical |
| **char state[20]:** | State of the chemical (solid/liquid/gas) |
| **float qty:** | Quantity of the chemical present |
| **char unit[3]:** | Unit of measurement |
| **float alert\_qty:** | Quantity below which notification is to be given |
| **char alert\_unit[3]:** | Units of measurement (g/ml) |
| **char description[1000]:** | Description of the chemical |
| **char manufacturer[100]:** | Name of the manufacturer of the chemical |

**PUBLIC SECTION**

**Member Functions:**

|  |  |
| --- | --- |
| **void IN(int):** | To enter the details of a chemical |
| **void OUTALL():** | To display only the important details of a chemical |
| **void OUTONE():** | To display all the details of the chemical |
| **int retid():** | Returns the chemical ID |
| **int retqty():** | Returns the present quantity of the chemical |
| **int retnot():** | Returns the alert quantity of the chemical |
| **char\* retname():** | Returns the name of the chemical |
| **void editname():** | To edit the name of the chemical |
| **void editformula():** | To edit the formula of the chemical |
| **void editstate():** | To edit the state of the chemical |
| **void editqty():** | To edit the quantity of the chemical |
| **void editalert():** | To edit the alert quantity of the chemical |
| **void editdscrp():** | To edit the description of the chemical |
| **void editmanu():** | To edit the name of the manufacturer of the chemical |
| **void issueqty(float):** | To enter the quantity of the chemical to be issued |
| **void returnqty(float):** | To enter the quantity of the chemical to be returned |
| **void notice():** | To display the details of the chemical with qty<alert\_qty |

Non-Member Functions

|  |  |
| --- | --- |
| **void NEW():** | To get the details of a new record and store them in the file ‘Records.txt’ |
| **void SEARCH():** | To search a particular chemical in the file ‘Records.txt’ using the Chemical ID and displaying all the details of the chemical including the description |
| **void EDIT():** | To modify the details of a chemical stored by first searching the chemical and then enquiring which details is to be edited and making the changes in the file |
| **void DELETE():** | To delete a whole record of a chemical from the file ‘Records.txt’ |
| **void LIST():** | To display the list of all the chemical stored in the records along with important details |
| **int AUTOGENERATE():** | To generate the Chemical ID automatically each time a new entry is made |
| **void ISSUE():** | To issue a chemical stored, i.e. making the change in the quantity stored in the file |
| **void RETURN():** | To return a chemical stored, i.e. making the change in the quantity stored in the file |
| **void NOTIFY():** | To display the list of the chemicals which are going to get out of stock. It shows the details of those chemicals which have their current quantity equal to or less than their alert quantity |
| **void STOCK():** | Used to remove the load from main() and implement the first case of the main menu, i.e. Chemical Stock |
| **void CASE2():** | Used to decrease the load of main() and implement the second case of the main menu, i.e. Issue/Return Chemical |
| **void PRINT():** | To print the name of the institution and the heading |
| **void DATE():** | To print the system date |
| **void EXIT():** | To print the message while exiting a menu |
| **void LOADING():** | To print the loading bar and the message |
| **void UPDATE():** | To print the last updated message on the top right corner of the monitor |
| **void FIRST():** | To print the welcoming screen |
| **void LAST():** | To print the message while exiting the program |

SOURCE CODE

#include<string.h>

#include<fstream.h>

#include<conio.h>

#include<stdio.h>

#include<process.h>

#include<iomanip.h>

#include<dos.h>

void NEW();

void SEARCH();

void EDIT();

void DELETE();

void LIST();

int AUTOGENERATE();

void ISSUE();

void NOTIFY();

void STOCK();

void CASE2();

void PRINT();

void DATE();

void EXIT();

void LOADING();

void UPDATE();

void RETURN();

void FIRST();

void LAST();

date d;

time t;

class chemical

{

int id;

char name[400];

char formula[100];

char state[20];

float qty;

char unit[3];

float alert\_qty;

char alert\_unit[3];

char description[1000];

char manufacturer[100];

public:

void IN(int);

void OUTALL();

void OUTONE();

int retid();

int retqty();

int retnot();

char\* retname();

void editname();

void editformula();

void editstate();

void editqty();

void editalert();

void editdscrp();

void editmanu();

void issueqty(float);

void returnqty(float);

void notice();

};

void chemical::IN(int i)

{

char option;

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<endl;

id=i;

cout<<"\n Enter the name of the chemical :\t";

gets(name);

cout<<"\n Enter the name of the manufacturer :\t";

gets(manufacturer);

cout<<"\n Enter the formula of the chemical :\t";

cin>>formula;

gotoxy(2,9);

cout<<"\n Enter the state of the chemical at room temperature :\t";

cout<<"\n a. Solid";

cout<<"\n b. Liquid";

cout<<"\n c. Gas";

cout<<"\n Enter your choice :\t";

gotoxy(28,14);

label1: cin>>option;

switch(option)

{

case 'a':

case 'A': strcpy(state,"Solid");

gotoxy(2,16);

clreol();

break;

case 'b':

case 'B': strcpy(state,"Liquid");

gotoxy(2,16);

clreol();

break;

case 'c':

case 'C': strcpy(state,"Gas");

gotoxy(2,16);

clreol();

break;

default : gotoxy(2,15);

cout<<"\n Invalid input! Please select again!";

gotoxy(27,14);

clreol();

goto label1;

}

cout<<"\n Enter the quantity present in the lab :\t";

cin>>qty;

cout<<"\n Enter the units of the measurement (ml/g) :\t";

cin>>unit;

cout<<"\n Enter the quantity for which you wish to be notified :\t";

cin>>alert\_qty;

cout<<"\n Enter the units of measurement (ml/g) :\t";

cin>>alert\_unit;

textbackground(6);

cout<<"\n Enter the description of the chemical";

cout<<" (please enter \'@\' symbol when you are finished) :"<<endl;

cin.getline(description,1000,'@');

}

void chemical::OUTALL()

{

cout<<endl;

cout<<setw(4)<<id<<setw(30)<<name<<setw(15)<<manufacturer;

cout<<setw(10)<<formula<<setw(8)<<state<<setw(7)<<qty<<" "<<unit;

}

void chemical::notice()

{

cout<<endl;

cout<<setw(4)<<id<<setw(30)<<name<<setw(15)<<manufacturer;

cout<<setw(10)<<formula<<setw(8)<<qty<<" "<<unit<<setw(8)<<alert\_qty<<" "<<alert\_unit;

}

void chemical::OUTONE()

{

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<endl;

cout<<"\n ID :\t"<<id;

cout<<"\n Name :\t"<<name;

cout<<"\n Manufacturer :\t"<<manufacturer;

cout<<"\n Formula :\t"<<formula;

cout<<"\n Physical state at room temperature :\t"<<state;

cout<<"\n Current quantity present :\t"<<qty<<" "<<unit;

cout<<"\n General Description :\t"<<endl;

cout<<description;

}

int chemical::retid()

{

return id;

}

int chemical::retqty()

{

return qty;

}

int chemical::retnot()

{

return alert\_qty;

}

char\* chemical::retname()

{

char \*NAME=new char;

strcpy(NAME,name);

return NAME;

}

void chemical::editname()

{

cout<<"\n The current name of the chemical is :\t"<<name;

cout<<"\n Enter the new name of the chemical :\t";

gets(name);

}

void chemical::editformula()

{

cout<<"\n The current formula of the chemical is :\t"<<formula;

cout<<"\n Enter the new formula of the chemical :\t";

cin>>formula;

}

void chemical::editstate()

{

char option;

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<endl;

cout<<"\n The current state stored is :\t"<<state;

gotoxy(2,8);

cout<<"\n Enter the state of the chemical at room temperature :\t";

cout<<"\n a. Solid";

cout<<"\n b. Liquid";

cout<<"\n c. Gas";

cout<<"\n Enter your choice :\t";

gotoxy(28,13);

label1: cin>>option;

switch(option)

{

case 'a':

case 'A': strcpy(state,"Solid");

gotoxy(2,15);

clreol();

break;

case 'b':

case 'B': strcpy(state,"Liquid");

gotoxy(2,15);

clreol();

break;

case 'c':

case 'C': strcpy(state,"Gas");

gotoxy(2,15);

clreol();

break;

default : gotoxy(2,14);

cout<<"\n Invalid Input! Please select again!";

gotoxy(27,13);

clreol();

goto label1;

}

}

void chemical::editmanu()

{

cout<<"\n The current manufacturer is :\t"<<manufacturer;

cout<<"\n Enter the new name of the manufacturer :\t";

gets(manufacturer);

}

void chemical::editdscrp()

{

cout<<"\n Enter the new description of the chemical :\n";

cin.getline(description,1000,'@');

}

void chemical::editqty()

{

cout<<"\n Enter the current quantity of the chemical present in the lab :\t";

cin>>qty;

cout<<"\n Enter the units of measurement (g/ml) :\t";

cin>>unit;

}

void chemical::editalert()

{

cout<<"\n Enter the quantity of the chemical for which you wish to be notified :\t";

cin>>alert\_qty;

cout<<"\n Enter the units of measurement (g/ml) :\t";

cin>>alert\_qty;

}

void chemical::issueqty(float q)

{

qty-=q;

}

void chemical::returnqty(float q)

{

qty+=q;

}

long size=sizeof(chemical);

void main()

{

clrscr();

char choice;

FIRST();

while(1)

{

clrscr();

textbackground(6);

textcolor(0);

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<"\n\n MAIN-MENU";

cout<<"\n\n 1. Chemical Stock";

cout<<"\n 2. Issue/Return Chemical";

cout<<"\n 3. Notifications";

cout<<"\n 4. Exit";

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

label3: gotoxy(25,12);

clreol();

cin>>choice;

switch(choice)

{

case '1': LOADING();

STOCK();

break;

case '2': LOADING();

CASE2();

break;

case '3': LOADING();

NOTIFY();

getch();

break;

case '4': LAST();

break;

default : gotoxy(3,14);

cout<<"\n Invalid Input!";

goto label3;

break;

}

}

}

void STOCK()

{

char choice;

while(1)

{

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<"\n\n CHEMICAL STOCK";

cout<<"\n\n 1.1 New Entry";

cout<<"\n 1.2 Search Chemical";

cout<<"\n 1.3 Edit Chemical";

cout<<"\n 1.4 Delete Chemical";

cout<<"\n 1.5 List All Chemicals";

cout<<"\n 1.6 Exit";

cout<<"\n\n Please enter your choice :";

label4: gotoxy(34,14);

clreol();

cin>>choice;

switch(choice)

{

case '1': NEW();

break;

case '2': SEARCH();

getch();

break;

case '3': EDIT();

break;

case '4': DELETE();

break;

case '5': LOADING();

LIST();

getch();

break;

case '6': EXIT();

return;

default: gotoxy(3,35);

cout<<"\n Invalid Input!";

goto label4;

}

}

}

void CASE2()

{

char choice;

while(1)

{

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<"\n\n ISSUE/RETURN";

cout<<"\n\n 2.1 Issue Chemical";

cout<<"\n 2.2 Return Chemical";

cout<<"\n 2.3 Quit";

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

label5: gotoxy(27,11);

clreol();

cin>>choice;

switch(choice)

{

case '1': ISSUE();

break;

case '2': RETURN();

break;

case '3': EXIT();

return;

default : gotoxy(3,12);

cout<<"\n Invalid Input";

goto label5;

}

}

}

void NEW()

{

chemical c;

int i;

clrscr();

PRINT();

DATE();

UPDATE();

i=AUTOGENERATE();

gotoxy(3,3);

c.IN(i);

textbackground(6);

fstream f;

f.open("Record.txt",ios::app|ios::binary);

if(f.fail())

{

cerr<<"\n\n File cannot be opened!";

getch();

exit(1);

}

f.write((char\*)&c,size);

f.close();

textbackground(6);

cout<<"\n Chemical stored!";

getdate(&d);

gettime(&t);

}

void SEARCH()

{

int i,flag=0;

chemical c;

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<"\n\n Enter the id of the chemical you wish to search :\t";

cin>>i;

fstream f;

f.open("Record.txt",ios::in|ios::binary);

if(f.fail())

{

cerr<<"\n File cannot be opened!";

getch();

exit(1);

}

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

while(f.read((char\*)&c,size))

{

if (c.retid()==i)

{

flag=1;

break;

}

}

if(flag==1)

c.OUTONE();

else

cout<<"\n Chemical not found!";

}

void EDIT()

{

int i,flag=0;

char option,choice;

chemical c;

clrscr();

PRINT();

DATE();

UPDATE();

fstream f;

f.open("Record.txt",ios::in|ios::out|ios::binary);

if(f.fail())

{

cerr<<"\n File cannot be opened!";

getch();

exit(1);

}

gotoxy(3,3);

cout<<"\n\n Enter the id of the chemical you wish to edit :\t";

cin>>i;

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

while(f.read((char\*)&c,size))

{

if(c.retid()==i)

{

cout<<"\n What do you wish to edit?";

cout<<"\n a. Name of the chemical";

cout<<"\n b. Name of the manufacturer";

cout<<"\n c. Formula of the chemical";

cout<<"\n d. State of the chemical at room temperature";

cout<<"\n e. Quantity present in the lab";

cout<<"\n f. Quantity for which you wish to be notified";

cout<<"\n g. General description of the chemical";

cout<<"\n\n Press Q to quit";

gotoxy(2,15);

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

label2: gotoxy(27,17);

cin>>option;

textbackground(6);

switch(option)

{

case 'a':

case 'A': c.editname();

gotoxy(2,21);

clreol();

break;

case 'b':

case 'B': c.editmanu();

gotoxy(2,21);

clreol();

break;

case 'c':

case 'C': c.editformula();

gotoxy(2,21);

clreol();

break;

case 'd':

case 'D': c.editstate();

gotoxy(2,21);

clreol();

break;

case 'e':

case 'E': c.editqty();

gotoxy(2,21);

clreol();

break;

case 'f':

case 'F': c.editalert();

gotoxy(2,21);

clreol();

break;

case 'g':

case 'G': c.editdscrp();

gotoxy(2,21);

clreol();

break;

case 'Q':

case 'q': EXIT();

return;

default : gotoxy(2,19);

cout<<"\n Invalid Input! Please select again";

gotoxy(26,18);

clreol();

goto label2;

}

f.seekp(-size,ios::cur);

f.write((char\*)&c,size);

getdate(&d);

gettime(&t);

flag=1;

break;

}

}

f.close();

textbackground(6);

if(flag==1)

{

cout<<"\n Chemical edited!";

getch();

}

else

{

cout<<"\n Chemical not found!";

getch();

}

}

void DELETE()

{

chemical c;

int i,flag=0;

char option;

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<"\n\n Are you sure you want to delete a chemical from the database? (Y/N)\t ";

cin>>option;

if(option=='N'|option=='n')

return;

cout<<"\n Enter the id of the chemical you wish to delete :\t";

cin>>i;

fstream f1;

f1.open("Record.txt",ios::in|ios::binary);

if(f1.fail())

{

cerr<<"\n File cannot be opened!";

getch();

exit(1);

}

fstream f2;

f2.open("New.txt",ios::out|ios::binary);

if(f2.fail())

{

cerr<<"\n File cannot be opened!";

getch();

exit(1);

}

while (f1.read((char\*)&c,size))

{

if(c.retid()!=i)

f2.write((char\*)&c,size);

else if(c.retid()==i)

{

cout<<"\n Chemical deleted:\t"<<c.retname();

flag=1;

}

}

textbackground(6);

f1.close();

f2.close();

remove("Record.txt");

rename("New.txt","Record.txt");

if (flag==0)

cout<<"\n Chemical not found!";

else

{

getch();

getdate(&d);

gettime(&t);

}

}

void LIST()

{

fstream f;

chemical c;

clrscr();

PRINT();

DATE();

UPDATE();

cout<<endl;

f.open("Record.txt",ios::in|ios::binary);

if(f.fail())

{

cerr<<"\n File cannot be opened!";

getch();

exit(1);

}

cout<<endl;

cout<<setw(4)<<"ID"<<setw(30)<<"Name"<<setw(15)<<"Manufacturer";

cout<<setw(10)<<"Formula"<<setw(8)<<"State"<<setw(7)<<"Qty";

cout<<endl;

textbackground(6);

while(f.read((char\*)&c,size))

c.OUTALL();

f.close();

}

int AUTOGENERATE()

{

fstream f;

chemical c;

clrscr();

PRINT();

DATE();

UPDATE();

f.open("Record.txt",ios::in|ios::out|ios::binary);

if(f.fail())

{

cerr<<"\n File cannot be opened!";

getch();

exit(1);

}

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

long t=f.tellg();

int i;

if(t==0)

i=1;

else

{

f.seekg(-size,ios::end);

f.read((char\*)&c,size);

i=c.retid();

i=i+1;

}

f.close();

return i;

}

void ISSUE()

{

int i,flag;

float qty;

chemical c;

char option;

clrscr();

flag=0;

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<"\n\n\n Enter the id of the chemical you wish to issue :\t";

cin>>i;

fstream f;

f.open("Record.txt",ios::in|ios::out|ios::binary);

if(f.fail())

{

cerr<<"\n\n File cannot be opened!";

getch();

exit(1);

}

textbackground(6);

while(f.read((char\*)&c,size))

{

if(c.retid()==i)

{

flag=1;

break;

}

}

if(flag==0)

{

cout<<"\n Chemical not found!";

getch();

return;

}

cout<<"\n Enter the amount of quantity you wish to issue (< "<<c.retqty()<<" g/ml):\t";

cin>>qty;

if(qty>c.retqty())

{

cout<<"\n\n Issuing quantity must be less than the quantity present in the lab!";

getch();

exit(0);

}

cout<<"\n Are you sure you want to issue this quantity of the chemical? (Y/N)";

cin>>option;

if(option=='y'||option=='Y')

{

c.issueqty(qty);

f.seekp(-size,ios::cur);

f.write((char\*)&c,size);

f.close();

}

textbackground(6);

if(option=='n'||option=='N')

{

EXIT();

return;

}

cout<<"\n Quantity Issued!";

getch();

getdate(&d);

gettime(&t);

}

void RETURN()

{

int i,flag=0;

float qty;

chemical c;

char option;

fstream f;

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<"\n\n\n Enter the id of the chemical you wish to return :\t";

cin>>i;

f.open("Record.txt",ios::in|ios::out|ios::binary);

if(f.fail())

{

cerr<<"\n\n File cannot be opened!";

getch();

exit(1);

}

while(f.read((char\*)&c,size))

{

if(c.retid()==i)

{

flag=1;

break;

}

}

if(flag==0)

{

cout<<"\n Chemical not found!";

getch();

return;

}

cout<<"\n Enter the amount of quantity you wish to return (g/ml):\t";

cin>>qty;

cout<<"\n Are you sure you want to return this quantity of the chemical? (Y/N)";

cin>>option;

if(option=='n'||option=='N');

{

EXIT();

return;

}

if (option=='y'||option=='Y')

{

c.returnqty(qty);

f.seekp(-size,ios::cur);

f.write((char\*)&c,size);

f.close();

}

cout<<"\n Quantity Returned!";

getch();

getdate(&d);

gettime(&t);

}

void NOTIFY()

{

fstream f;

chemical c;

clrscr();

PRINT();

DATE();

UPDATE();

gotoxy(3,3);

cout<<endl;

f.open("Record.txt",ios::in|ios::binary);

if(f.fail())

{

cerr<<"\n File cannot be opened!";

getch();

exit(1);

}

cout<<"\n The following chemicals are going to get\n out of stock soon!";

cout<<"\n Please replenish as soon as possible!";

cout<<endl<<endl<<endl;

cout<<setw(4)<<"ID"<<setw(30)<<"Name"<<setw(15)<<"Manufacturer";

cout<<setw(10)<<"Formula"<<setw(8)<<"Qty"<<setw(8)<<"Alert Qty";

cout<<endl;

textbackground(6);

while(f.read((char\*)&c,size))

{

if(c.retqty()-c.retnot()<=10)

c.notice();

}

f.close();

}

void PRINT()

{

clrscr();

gotoxy(23,1);

cout<<"Fr. Agnel School, New Delhi";

gotoxy(29,2);

cout<<"Chemistry Lab";

}

void DATE()

{

gotoxy(60,3);

date d;

getdate(&d);

cout<<(int)d.da\_day<<" / "<<(int)d.da\_mon<<" / "<<d.da\_year;

}

void EXIT()

{

clrscr();

int i;

gotoxy(30,8);

cout<<"Exiting";

for(i=0;i<4;i++)

{

cout<<".";

delay(300);

}

delay(500);

cout<<".";

delay(200);

cout<<".";

}

void LOADING()

{

clrscr();

int i;

gotoxy(20,4);

cout<<"Please wait...";

gotoxy(21,5);

cout<<"LOADING";

gotoxy(21,6);

for(i=0;i<5;i++)

{

cout<<"|";

delay(250);

}

delay(700);

cout<<"|";

}

void UPDATE()

{

gotoxy(60,5);

cout<<"Last updated on:";

gotoxy(60,6);

cout<<(int)d.da\_day<<" / "<<(int)d.da\_mon<<" / "<<(int)d.da\_year;

gotoxy(60,7);

cout<<(int)t.ti\_hour<<":"<<(int)t.ti\_min<<":"<<(int)t.ti\_sec;

}

void FIRST()

{

clrscr();

int i;

textbackground(6);

textcolor(0);

gotoxy(33,2);

cout<<"WELCOME!";

for(i=19;i>14;i--)

{

gotoxy(i,5);

cout<<"=";

delay(150);

}

for(i=6;i<10;i++)

{

gotoxy(14,i);

cout<<"||";

delay(150);

}

for(i=15;i<20;i++)

{

gotoxy(i,10);

cout<<"=";

delay(150);

}

for(i=5;i<10;i++)

{

gotoxy(32,i);

cout<<"||";

delay(150);

}

for(i=33;i<39;i++)

{

gotoxy(i,10);

cout<<"=";

delay(150);

}

gotoxy(45,5);

textcolor(9);

cout<<"|| \\\\";

delay(150);

gotoxy(45,6);

textcolor(10);

cout<<"|| \\\\";

delay(150);

gotoxy(45,7);

textcolor(11);

cout<<"|| ||";

delay(150);

gotoxy(45,8);

textcolor(12);

cout<<"|| ||";

delay(150);

gotoxy(45,9);

textcolor(13);

cout<<"|| //";

delay(150);

gotoxy(45,10);

textcolor(14);

cout<<"|| //";

delay(800);

gotoxy(21,9);

cout<<"HEMISTRY";

gotoxy(40,9);

cout<<"AB";

gotoxy(53,9);

cout<<"ATABASE";

delay(500);

gotoxy(22,14);

cout<<"\*\*\*Press Any Key To Continue\*\*\*";

gotoxy(36,17);

getch();

return;

}

void LAST()

{

clrscr();

gotoxy(20,10);

cout<<"Thank You For Using This Program!";

gotoxy(35,12);

cout<<"^o^";

gotoxy(36,14);

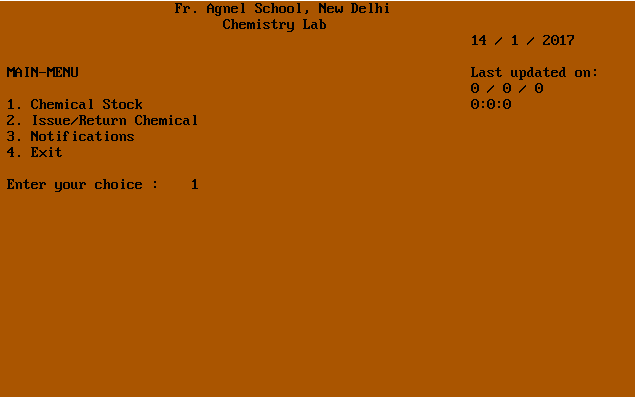
delay(2000);

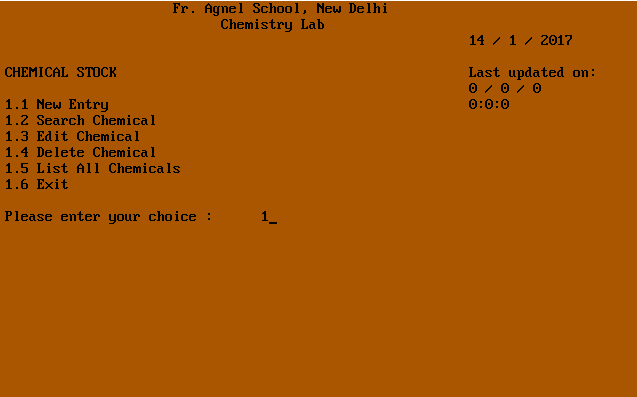
exit(0);

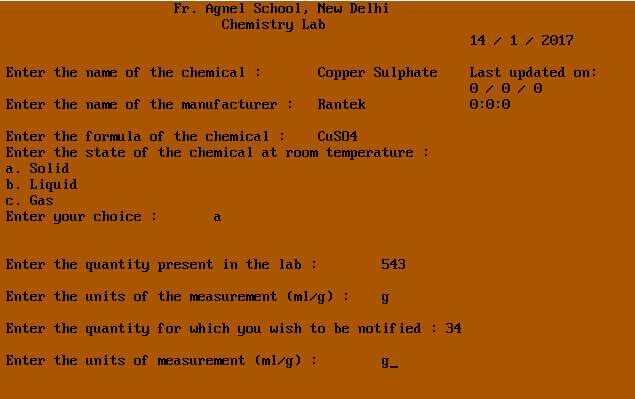
}

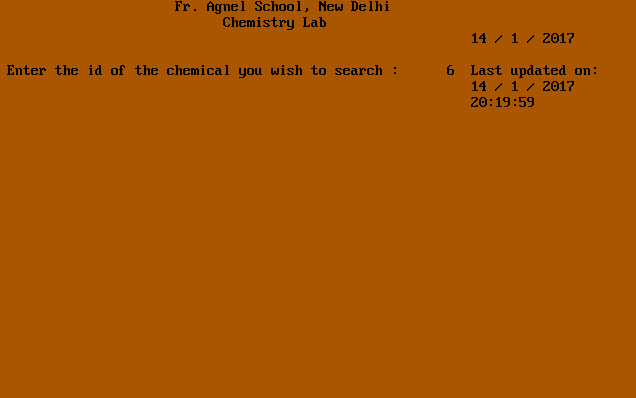
OUTPUT

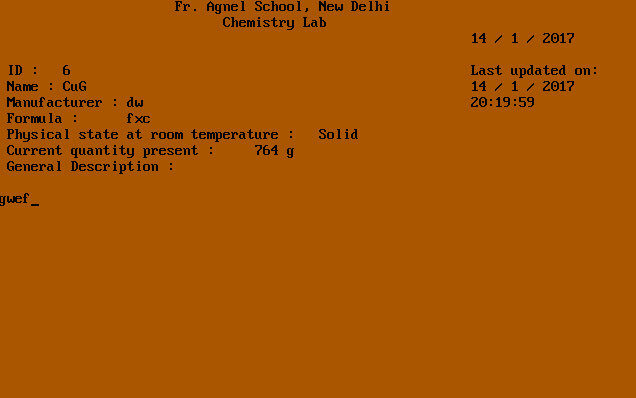
****

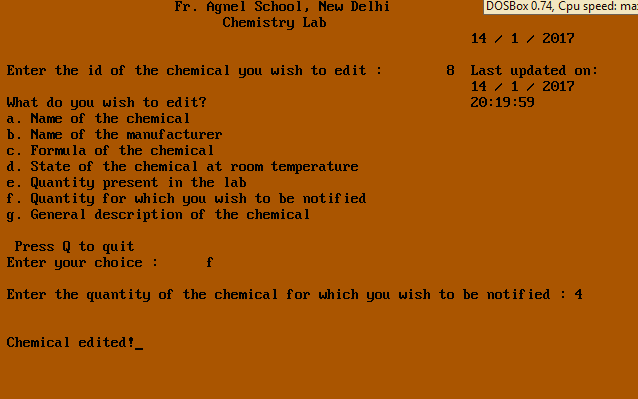
****

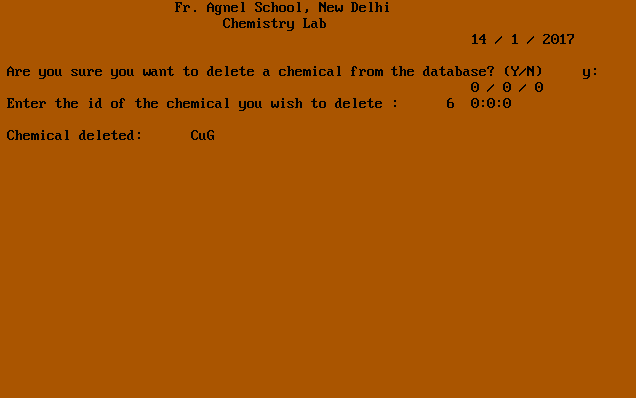
****

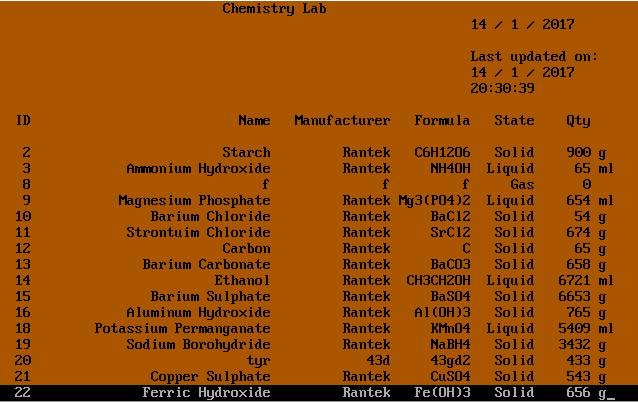
****

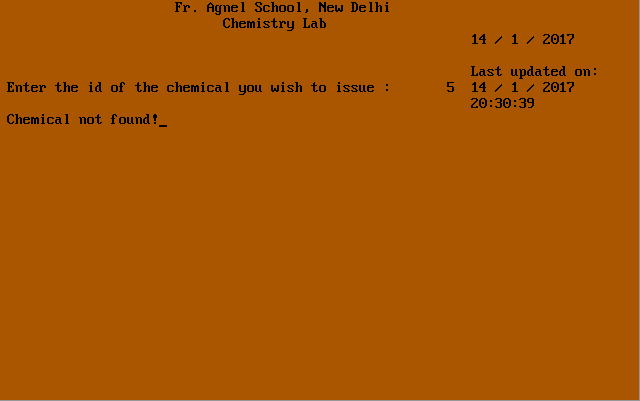


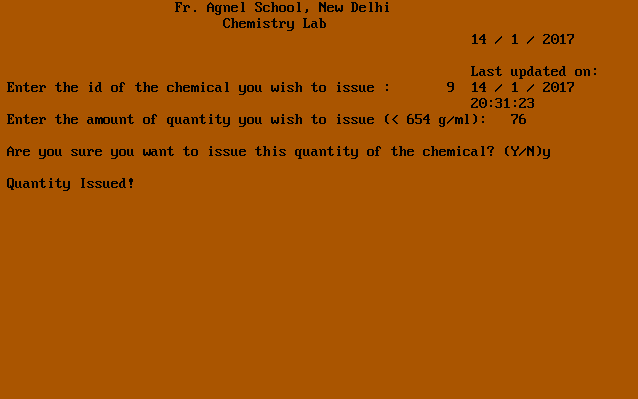


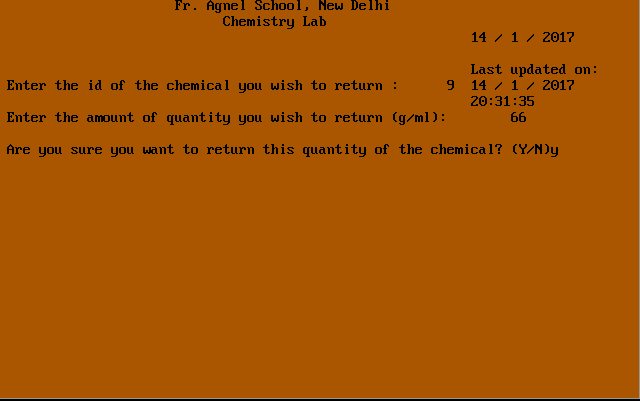
****

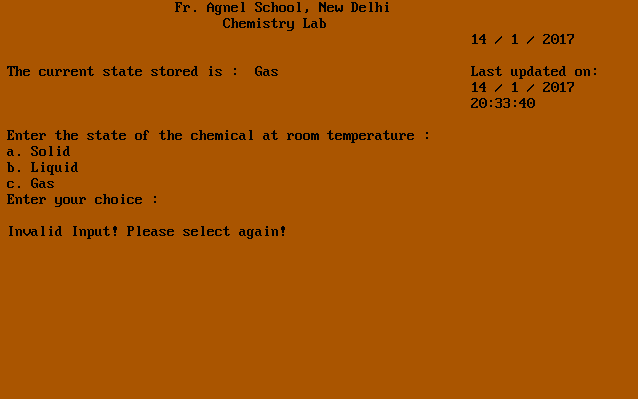
****

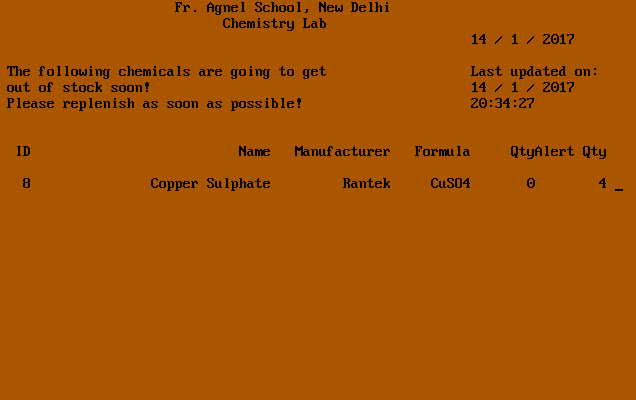
****

****

****

****

****

****