**Tutorial 11**

**Ques1. Run and analyze output of the following programs.**

(i)

#include<iostream>

using namespace std;

class Test

{

public:

Test();

};

Test::Test() {

cout << "Constructor Called \n";

}

Test t1;

int main() {

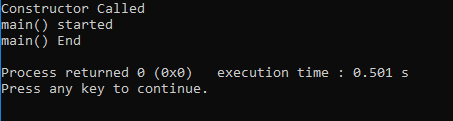
cout << "main() started\n";

cout << "main() End\n";

return 0;

}

OUTPUT:-



(ii)

#include<iostream>

using namespace std;

class Test

{

public:

Test();

};

Test::Test() {

cout << "Constructor Called \n";

}

void fun() {

Test t1;

}

int main() {

cout << "Before fun() called\n";

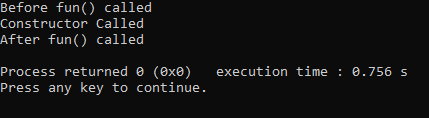
fun();

cout << "After fun() called\n";

return 0;

}

OUTPUT:-



(iii)

#include<iostream>

using namespace std;

class Test

{

public:

Test();

};

Test::Test() {

cout << "Constructor Called \n"; }

void fun() {

static Test t1; }

int main() {

cout << "Before fun() called\n";

fun();

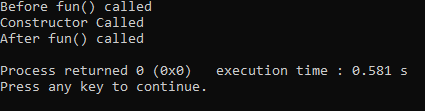
cout << "After fun() called\n";

fun(); //will the constructor be

called this time?

return 0; }

OUTPUT:-



(iv)

#include<iostream>

using namespace std;

class Test

{

public:

Test();

};

Test::Test() {

cout<<"Constructor Called \n";

}

int main()

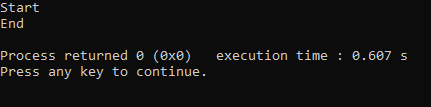
{cout<<"Start \n";

Test t1();

cout<<"End \n";

return 0;}

OUTPUT:-



(v)

#include<iostream>

using namespace std;

int &fun() {

static int a = 10;

return a;}

int main() {

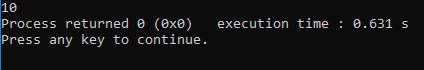
int y = fun();

y = y +30;

cout<<fun();

return 0; }

OUTPUT:-



(vi)

#include<iostream>

using namespace std;

int &fun() {

static int a = 10;

return a;

}

int main() {

int &y = fun();

y = y +30;

cout<<fun();

return 0; }

OUTPUT:-



(vii)

#include<iostream>

using namespace std;

/\* Swaps strings by swapping

pointers \*/

void swap1(char \*\*str1\_ptr, char

\*\*str2\_ptr)

{

char \*temp = \*str1\_ptr;

\*str1\_ptr = \*str2\_ptr;

\*str2\_ptr = temp;

}

int main()

{

char \*str1 = "JIIT";

char \*str2 = "NOIDA";

swap1(&str1, &str2);

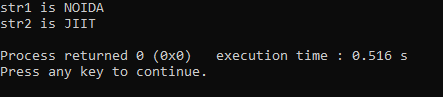
cout<<"str1 is "<<str1<<endl;

cout<<"str2 is "<<str2<<endl;

return 0;

}

OUTPUT:-



(viii)

#include<iostream>

using namespace std;

class Point {

private:

int x;

int y;

public:

Point(int i = 0, int j = 0); //

Normal Constructor

Point(const Point &t); // Copy

Constructor

};

Point::Point(int i, int j) {

x = i;

y = j;

cout << "Normal Cunstroctor

called\n";

}

Point::Point(const Point &t) {

y = t.y;

cout << "Copy constructor

called\n";

}

int main()

{

Point \*t1, \*t2;

t1 = new Point(10, 15);

t2 = new Point(\*t1);

Point t3 = \*t1;

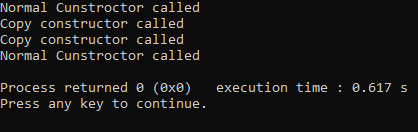
Point t4;

t4 = t3;

return 0;

}

OUTPUT:-



Ques 2.

A Book Shop Inventory uses a personal computer to maintain the inventory of books. The list includes details

such as author, title, price, publisher, stock position etc. Whenever a customer wants a book, the shop Keeper

inputs the title and author of the book and the system replies whether it is in list or not. If it is not, an appropriate

message is displayed. If book is in the list, then the system displays the book details and asks for the no. of

copies. If the requested copies are available, the total cost of books displayed, and otherwise the message

“required copies not in stock” is displayed. Use appropriate Data structure of your choice to minimize search

time and implement the system

CODE:-

#include<iostream>

#include<string.h>

#include<cstdio>

using namespace std;

class Bookshop

{

public:

char author[100],title[100],publisher[100];

int price,NB;

void getdata();

void display();

};

void Bookshop::getdata()

{

cout<<"Enter the Author name:";

fflush(stdin);

cin.getline(author,100);

cout<<"Enter the Title of book:";

fflush(stdin);

cin.getline(title,100);

cout<<"Enter the Publisher of book:";

fflush(stdin);

cin.getline(publisher,100);

cout<<"Enter the Price of book:";

cin>>price;

cout<<"Enter the Number of books:";

cin>>NB;

cout<<endl;

}

void Bookshop::display()

{

cout<<endl;

cout<<"Author Name:"<<author<<endl;

cout<<"Title of Book:"<<title<<endl;

cout<<"Publisher:"<<publisher<<endl;

cout<<"Price of book:"<<price<<endl;

cout<<"Number of books:"<<NB<<endl;

cout<<endl;

}

void compare(int n,Bookshop t[])

{

int i,f=0,cp;

char tt[100],an[100];

cout<<"Enter the Author name:";

fflush(stdin);

cin.getline(an,100);

cout<<"Enter the Title of book:";

fflush(stdin);

cin.getline(tt,100);

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

{

if(strcmp(tt,t[i].title)==0 && strcmp(an,t[i].author)==0)

{

f=1;

t[i].display();

cout<<endl<<"Enter the required no. of copies:";

cin>>cp;

if(cp<=t[i].NB)

cout<<"Total cost books:"<<t[i].price\*cp<<endl;

else

cout<<"Required copies not in stock"<<endl;

cout<<endl;

}

}

if(f==0)

cout<<"Book is not available!!!"<<endl;

}

int main()

{

int n,ch,i;

cout<<"Enter the no of Books present in Bookshop:";

cin>>n;

Bookshop t[n];

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

t[i].getdata();

cout<<"Choices:-"<<endl;

cout<<"1. Book Required"<<endl;

cout<<"2. Display details of all books"<<endl;

cout<<"3. Exit"<<endl;

while(1)

{

cout<<"Enter your choice:";

cin>>ch;

switch(ch)

{

case 1:

compare(n,t);

break;

case 2:

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

t[i].display();

case 3:

break;

}

if(ch==3)

break;

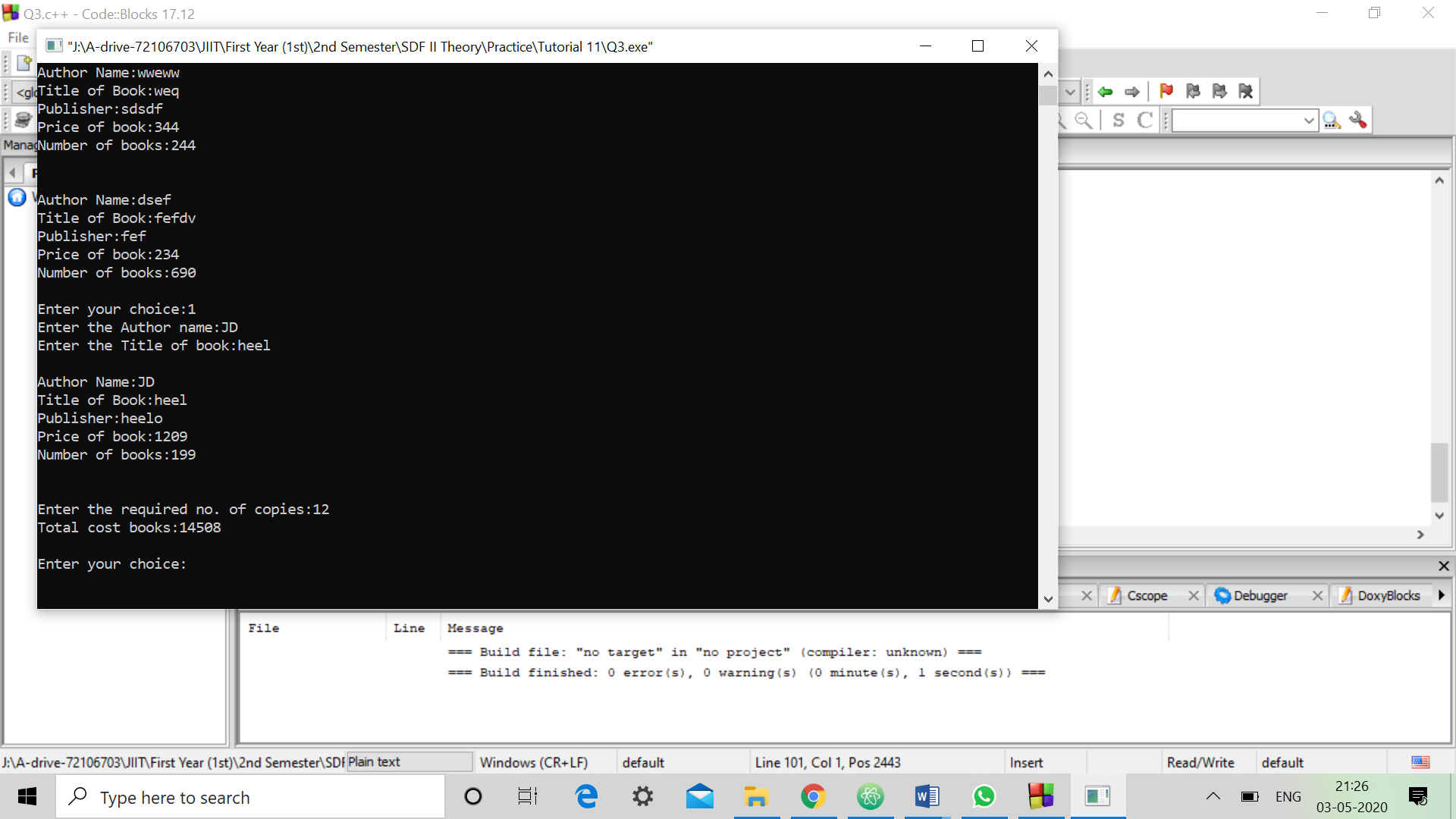
i++;

}

return 0;

}

OUTPUT:-



Q.3 Create a class “Employee” that includes three pieces of information as data members—a first name (type

string), a last name (type string) and a monthly salary (type int). Your class should have a constructor that

initializes the three data members. Provide a set and a get function for each data member. Create two Employee objects and display each object‟s yearly salary. Then give each Employee a 10 percent raise and display each Employee‟s yearly salary again. The class should be able to give the count of all the default objects. Write a test program that demonstrates class Employee capabilities.

CODE:-

#include<iostream>

#include<string.h>

#include<cstdio>

using namespace std;

static int count=0;

class employee

{

public:

char fn[100],ln[100];

int ms;

employee()

{

count++;

strcpy(fn,"FIRST NAME");

strcpy(ln,"Last name");

ms=0;

}

void set()

{

cout<<"Enter First Name: ";

fflush(stdin);

cin.getline(fn,100);

cout<<"enter last name: ";

fflush(stdin);

cin.getline(ln,100);

fflush(stdin);

cout<<"Enter monthly salary: ";

cin>>ms;

cout<<endl;

}

void get()

{

cout<<"First Name: "<<fn<<endl;

cout<<"Last Name: "<<ln<<endl;

cout<<"Monthly Salary: "<<ms<<endl;

cout<<endl;

}

};

int main()

{

employee e1,e2,e3;

cout<<"Enter details of first employee"<<endl;

e1.set();

cout<<"Enter details of second employee"<<endl;

e2.set();

cout<<"Yearly salary of first employee is: "<<e1.ms\*12<<endl;

cout<<"Yearly salary of second employee is: "<<e2.ms\*12<<endl;

cout<<endl<<"After 10% rise,salary of the two is: "<<endl;

float x,y;

x=0.1\*(float)e1.ms+e1.ms;

y=0.1\*(float)e2.ms+e2.ms;

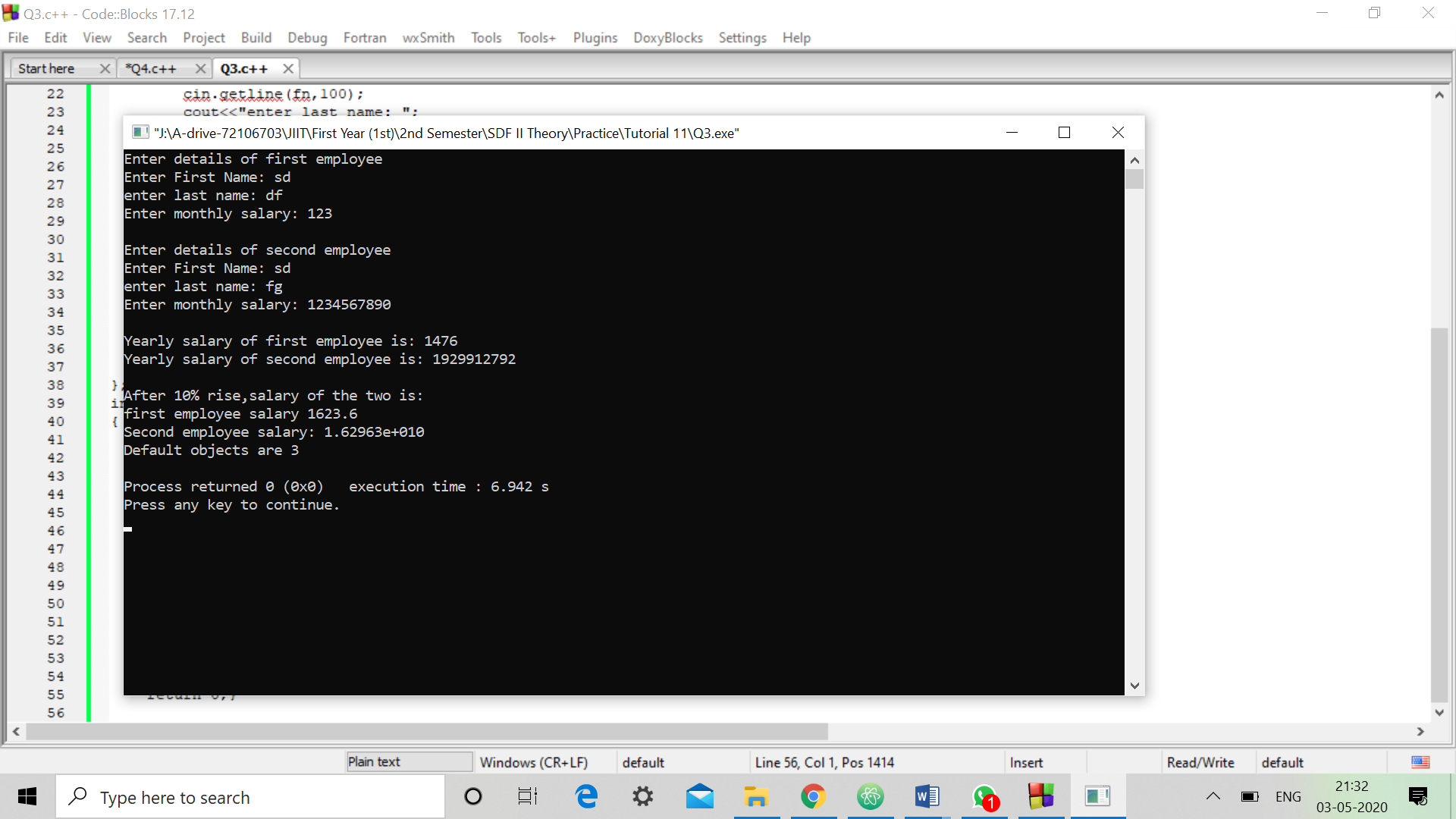
cout<<"first employee salary "<<x\*12<<endl;

cout<<"Second employee salary: "<<y\*12<<endl;

cout<<"Default objects are "<<count<<endl;

return 0;}

OUTPUT:-



Q.4 Define a class named “Document” that contains a member variable of type string named „text‟ that store any textual content for the document. Create functions named „getText‟ and „setText‟ that gets and sets text field. Define a class for „Email‟ that is derived from „Document‟ and that includes member variables for the sender, recipient and title of an email message. Implement appropriate get and set functions. The body of the email message should be stored in the inherited variable text. Define a class for „File‟ that is derived from „Document‟ and that includes a member variable for the pathname. Implement appropriate get and set function for the pathname. Create several sample objects of type „Email‟ and „File‟ in the main function to test the program. Also, write an appropriate function to search a keyword entered by user in Email/File.

CODE:-

#include<iostream>

#include<string.h>

#include<cstdio>

using namespace std;

class Document

{

private:

string text;

public:

void settext(string);

string gettext();

void fikey();

};

string Document::gettext()

{

return text;

}

void Document::settext(string x)

{

text=x;

}

class Email:public Document

{

private:

string member,recipient,title;

public:

void setmember(string);

string getmember();

void setrecipient(string);

string getrecipient();

void settitle(string);

string gettitle();

};

string Email::getmember()

{

return member;

}

void Email::setmember(string x)

{

member=x;

}

string Email::getrecipient()

{

return recipient;

}

void Email::setrecipient(string x)

{

recipient=x;

}

string Email::gettitle()

{

return title;

}

void Email::settitle(string x)

{

title=x;

}

class File: public Document

{

private:

string pathname;

public:

void setpathname(string);

string getpathname();

};

string File::getpathname()

{

return pathname;

}

void File::setpathname(string x)

{

pathname=x;

}

void Document::fikey()

{

string f;

cout<<"Enter the keyword:";

cin>>f;

if(text.find(f,0)!=string::npos)

cout<<"Keyword is found"<<endl;

else

cout<<"Keyword is not found"<<endl;

}

int main()

{

Email Doc;

string s;

cout<<"Entry in email:-"<<endl;

cout<<"Enter Member name:";

fflush(stdin);

getline(cin,s);

Doc.setmember(s);

cout<<"Enter Recipient:";

fflush(stdin);

getline(cin,s);

Doc.setrecipient(s);

cout<<"Enter Title:";

fflush(stdin);

getline(cin,s);

Doc.settitle(s);

cout<<"Enter the Message:-"<<endl;

fflush(stdin);

getline(cin,s);

Doc.settext(s);

File f;

cout<<"Entry in File:-"<<endl;

cout<<"Enter Pathname:";

fflush(stdin);

getline(cin,s);

f.setpathname(s);

cout<<endl<<"OUTPUT:-"<<endl;

cout<<"Text:"<<Doc.gettext()<<endl;

cout<<"Member:"<<Doc.getmember()<<endl;

cout<<"Recipient:"<<Doc.getrecipient()<<endl;

cout<<"Title:"<<Doc.gettitle()<<endl;

cout<<"File Pathname:"<<f.getpathname()<<endl;

int ch;

cout<<"Choices:-"<<endl;

cout<<"1.Search"<<endl;

cout<<"2.Exit"<<endl;

while(1)

{

cout<<"Enter the choice:";

cin>>ch;

switch(ch)

{

case 1: Doc.fikey();

break;

}

if(ch==2)

break;

}

return 0;

}

OUTPUT:-

