**Practical No. 9**

**File Handling**

**Program 9(a):** Design a class FileDemo open a file in real mode and display the total number of words and lines in the file.

**Coding:**

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

void main()

{

clrscr();

char msg[20];

ofstream o;

o.open("test()",ios::in);

o<<"HelloFriend!!"<<endl;

o<<"Bye!!"<<endl;

o.close();

ifstream i;

i.open("test()",ios::out);

i>>msg;

cout<<msg<<endl;

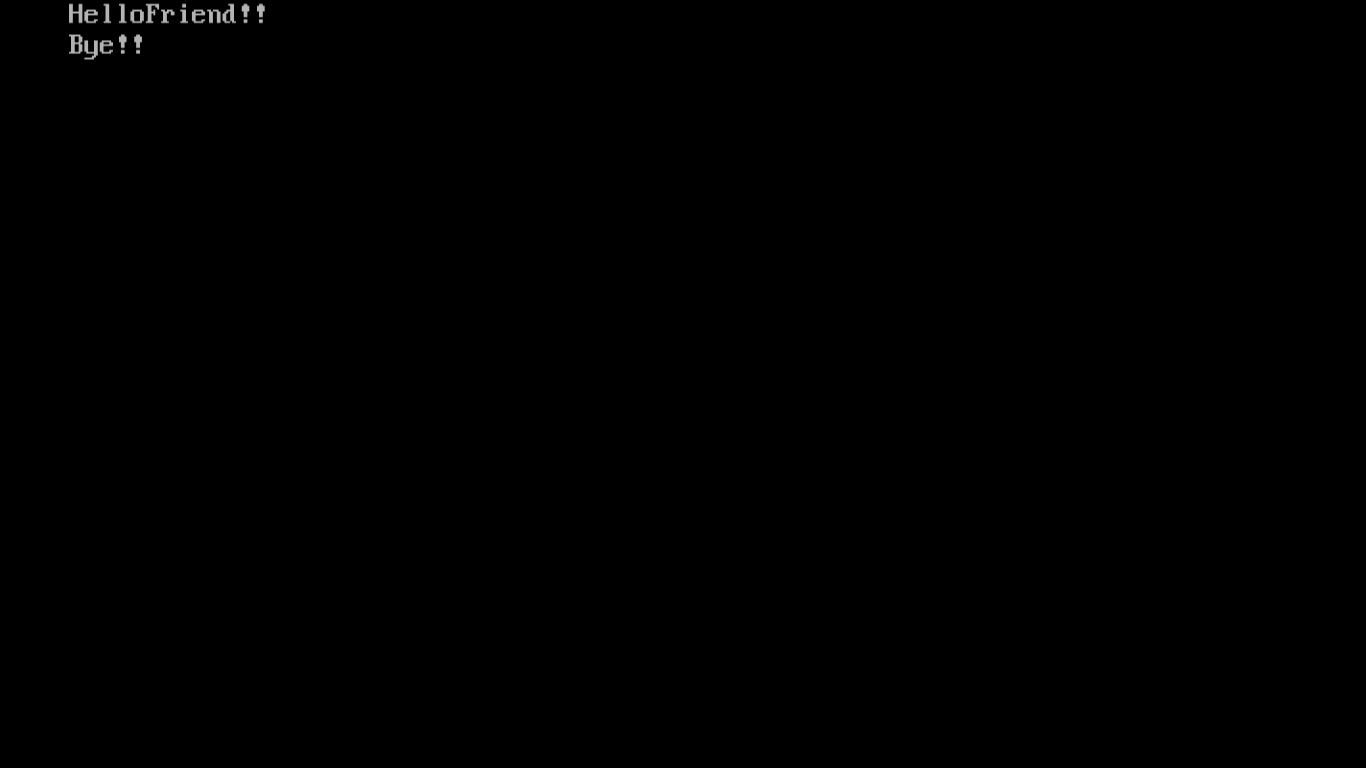
i>>msg;

cout<<msg<<endl;

getch();

}

**Output:**



**Program 9(b):** Design a class to handle multiple files and file operations.

**Example:** Write a c++ program to read character data from a file. Create one file to store all capital alphabets and another file to store all small case alphabets. Also display contents of both files.

**Coding:**

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

void main()

{

clrscr();

char msg;

int i;

ofstream o;

o.open("test0",ios::in);

for(i=0;i<=25;i++)

{

o<<char('A'+i)<<endl;

}

o.close();

o.open("test0",ios::in);

for(i=0;i<=25;i++)

{

o<<char('A'+i)<<endl;

}

o.close();

o.open("test01",ios::in);

for(i=0;i<=25;i++)

{

o<<char('a'+i)<<endl;

}

o.close();

ifstream in;

in.open("test0",ios::out);

cout<<"Reading from file 1:\n";

for(i=1;i<=26;i++)

{

in>>msg;

cout<<msg<<"\t";

}

cout<<"\n\n\nReading from file 2:"<<endl;

in.close();

in.open("test01",ios::out);

for(i=1;i<=26;i++)

{

in>>msg;

cout<<msg<<"\t";

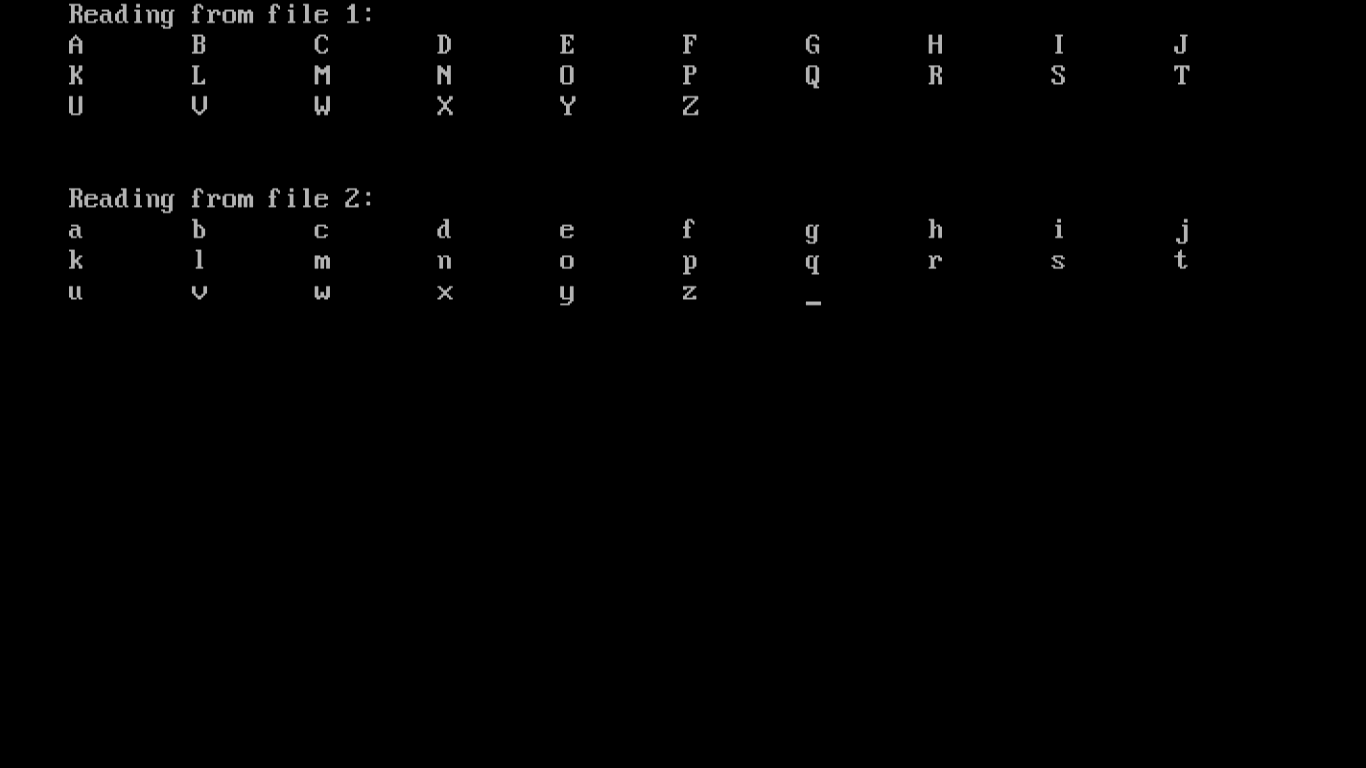
}

in.close();

getch();

}

**Output:**



**Program 9(c):** Design an editor for appending and editing the files.

**Example:** Write a program to write string, integer and float to a file and read it as binary data.

**Coding:**

#include<iostream.h>

#include<fstream.h>

#include<conio.h>

void main()

{

clrscr();

char name[20],ch;

int roll;

float fee;

ofstream o;

o.open("test0",ios::in);

o<<"Ajay"<<endl;

o<<24<<endl;

o<<55988.45<<endl;

o.close();

ifstream i;

i.open("test0",ios::out);

while(i)

{

i.get(ch);

cout<<ch;

}

getch();

}

**Output:**

