C++ project

[Type your address] ⦁ [Type your phone number] ⦁ [Type your e-mail address]

C++ project



BY:

Ruchir Jain

with

Kuber Rawat

St. Columba’s School

Roll no : 18

**XII-E**

Certificate

***This is to certify that Ruchir Jain of St.columbas School, Class 12th has completed this project ‘File Manager’ under my supervision , and completed to my satisfaction.***

**Signature**

*Aim*

The Aim of the project is to build a basic, user friendly file manager which helps you to open, rename, copy, move and organize your files in a smarter way. This application contains a multi-clipboard attribute which helps you to copy or move file from multiple locations to one location simultaneously; a feature which is not present in most of the other file managers. Hence saves a lot of time.  
This project was not created for only board purposes, but also for use in daily life.

*Header Files*

|  |  |  |
| --- | --- | --- |
| **Header files** | **Functions used** | **Used to** |
| fstream.h | cin | get input |
| cout | get output |
| open() | open file |
| close() | close file |
| conio.h | wherex() | get x coordinate |
| wherey() | get y coordinate |
| window() | define output window |
| textbackground() | set background colour |
| textcolor() | set text color |
| clrscr() | clear the screen |
| \_setcursortype() | sets cursor design |
| gotoxy() | goto specific location |
| cprintf() | print on screen |
| getch() | get a character from user |
| clreol() | clear end of line |
| dos.h | delay() | delay the output |
| stdio.h | sprint() | sent formatted output to screen |
| rename() | rename a file |
| remove() | remove a file |
| dir.h | findfirst() | search file in directory |
| findnext() | continue search for files |
| setdisk() | sets current drive no. |
| getdisk() | get current drive no. |
| mkdir() | makes a directory |
| string.h | strcpy() | copy 1 string to another |
| strcmp() | compare 2 strings |
| strcat() | combine 2 strings |
| strlen() | get string length |
| strupr() | converts lowercase to uppercase |
| graphics.h | initgraph() | initialize graphics system |
| graphresult() | return error code |
| line() | draw a line |
| getmaxx() | returns max. ordinate |
| getmaxy() | returns max. abscissa |
| closegraph() | close graphics system |
| stdlib.h | exit() | exit from program |
| random() | generate random no. |

**User Defined Classes**

1. **class filei**

{ char name[13],attrib;

char time[9],date[9];

unsigned long size;

public:

void copy(ffblk &f)

{ strcpy(name,f.ff\_name);

attrib=f.ff\_attrib;

size=f.ff\_fsize;

}

void print();

void setname(char name1[13])

{ strcpy(name,name1); }

char retatt()

{ return attrib; }

int chkname()

{

if(!strcmp(name,".")||!strcmp(name,".."))

return 1;

else return 0;

}

void getname(char a[13])

{ strcpy(a,name); }

};

void filei::print()

{ int x=wherex(),y=wherey();

window(1,25,80,25);

textbackground(LIGHTBLUE);

textcolor(WHITE);

clrscr();

char str[80];

sprintf(str,"Name : %s ",name);

if(attrib!=16)

{ strcat(str,"Size : ");

char a[255];

printbytes(size,a);

strcat(str,a);

}

else strcat(str,"Directory");

statusbar2(str);

window(1,2,80,24);

textbackground(WHITE);

textcolor(BLACK);

gotoxy(x,y);

}

1. **class filep**

{ filei f;

char path[255];

public:

void getpath(char a[255])

{ strcpy(a,path); }

filei getfilei()

{ return f; }

void setfilei(filei fi)

{ f=fi; }

void setpath(char a[255])

{ strcpy(path,a); }

};

III.

**class copylist**

{ char dir[50];

public:

copylist(char a[50])

{ strcpy(dir,a); }

void addtolist(filep);

void execlist(char direc[255]);

void execmovelist(char direc[255]);

void clearlist();

};

void copylist::execmovelist(char direc[255])

{ ifstream ifile(dir,ios::binary);

filep f;

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

{ move(f,direc); }

ifile.close();

clearlist();

}

void copylist::addtolist(filep f)

{ ofstream ofile(dir,ios::app|ios::binary);

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

ofile.close();

}

void stat(char []);

void copylist::execlist(char direc[255])

{ ifstream ifile(dir,ios::binary);

filep f;

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

{ copy(f,direc); }

ifile.close();

clearlist();

}

void copylist::clearlist()

{ remove(dir); }

***Source Code***

#include<fstream.h>

#include<process.h>

#include<conio.h>

#include<dos.h>

#include<stdio.h>

#include<dir.h>

#include<string.h>

#include<graphics.h>

#include<stdlib.h>

void getdrives(char []);

void newln();

void openrename(char []);

void progressbar(int);

void statusbar2(char []);

void openfolder(char []);

void statusbar(char []);

void WELCOME();

class filei

{

char name[13],attrib;

char time[9],date[9];

unsigned long size;

public:

void copy(ffblk &f)

{

strcpy(name,f.ff\_name);

attrib=f.ff\_attrib;

size=f.ff\_fsize;

}

void print();

void setname(char name1[13])

{

strcpy(name,name1);

}

char retatt()

{

return attrib;

}

int chkname()

{

if(!strcmp(name,".")||!strcmp(name,".."))

return 1;

else return 0;

}

void getname(char a[13])

{

strcpy(a,name);

}

};

void printbytes(unsigned long size,char str [])

{

if(size>1024)

{

if((size/1024)>1024)

{

float a;

a=size/1024;

a/=1024;

sprintf(str,"%f MB",a);

return;

}

else

{

float a=size/1024;

sprintf(str,"%f KB",a);

return;

}

}

sprintf(str,"%d BYTES",size);

}

void filei::print()

{

int x=wherex(),y=wherey();

window(1,25,80,25);

textbackground(LIGHTBLUE);

textcolor(WHITE);

clrscr();

char str[80];

sprintf(str,"Name : %s ",name);

if(attrib!=16)

{

strcat(str,"Size : ");

char a[255];

printbytes(size,a);

strcat(str,a);

}

else strcat(str,"Directory");

statusbar2(str);

window(1,2,80,24);

textbackground(WHITE);

textcolor(BLACK);

gotoxy(x,y);

}

class filep

{

filei f;

char path[255];

public:

void getpath(char a[255])

{

strcpy(a,path);

}

filei getfilei()

{

return f;

}

void setfilei(filei fi)

{

f=fi;

}

void setpath(char a[255])

{

strcpy(path,a);

}

};

void move(filep f,char direc[255])

{

char original[255];

char final[255];

f.getpath(original);

char name[13];

f.getfilei().getname(name);

strcpy(final,direc);

strcat(final,"\\");

strcat(final,name);

rename(original,final);

}

void copy(filep f,char direc[255])

{

char original[255];

char final[255];

f.getpath(original);

char name[13];

f.getfilei().getname(name);

strcpy(final,direc);

strcat(final,"\\");

strcat(final,name);

fstream ofile(final,ios::binary|ios::out);

fstream ifile(original,ios::binary|ios::in);

char byte;

while(ifile.read(&byte,1))

ofile.write(&byte,1);

ofile.close();

ifile.close();

}

class copylist

{

char dir[50];

public:

copylist(char a[50])

{

strcpy(dir,a);

}

void addtolist(filep);

void execlist(char direc[255]);

void execmovelist(char direc[255]);

void clearlist();

};

void copylist::execmovelist(char direc[255])

{

ifstream ifile(dir,ios::binary);

filep f;

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

{

move(f,direc);

}

ifile.close();

clearlist();

}

void copylist::addtolist(filep f)

{

ofstream ofile(dir,ios::app|ios::binary);

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

ofile.close();

}

void stat(char []);

void copylist::execlist(char direc[255])

{

ifstream ifile(dir,ios::binary);

filep f;

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

{

copy(f,direc);

}

ifile.close();

clearlist();

}

void copylist::clearlist()

{

remove(dir);

}

void mycomp()

{

char myc[12]="My Computer";

statusbar(myc);

statusbar2("Details about your computer");

window(1,2,80,24);

textbackground(WHITE);

textcolor(BLACK);

clrscr();

char drives[50]="CDE";

getdrives(drives);

int n=strlen(drives);

int pos=0;

gotoxy(1,2);

textcolor(WHITE);

textbackground(LIGHTBLUE);

clreol();

cprintf(" %c:",drives[0]);

textbackground(WHITE);

textcolor(BLACK);

gotoxy(1,wherey()+1);

for(int i=1;i<n;i++)

{

cout<<drives[i]<<":";

gotoxy(1,wherey()+1);

}

gotoxy(1,2);

while(1)

{

char c=getch();

if(c==27)

exit(0);

if(c=='\b')

continue;

else if(!c)

{

switch(getch())

{

case 80:

if(pos+1!=n)

{

clrscr();

pos=pos+1;

gotoxy(1,2);

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

{

if(wherey()!=pos+2)

{

cout<<drives[i]<<":";

gotoxy(1,wherey()+1);

}

else

{

textbackground(LIGHTBLUE);

textcolor(WHITE);

clreol();

cprintf(" %c:",drives[i]);

gotoxy(1,wherey()+1);

textcolor(BLACK);

textbackground(WHITE);

}

}

gotoxy(1,pos+2);

}

break;

case 72:

if(pos-1>-1)

{

clrscr();

pos=pos-1;

gotoxy(1,2);

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

{

if(wherey()!=pos+2)

{

cout<<drives[i]<<":";

gotoxy(1,wherey()+1);

}

else

{

textbackground(LIGHTBLUE);

textcolor(WHITE);

clreol();

cprintf(" %c:",drives[i]);

gotoxy(1,wherey()+1);

textbackground(WHITE);

textcolor(BLACK);

}

}

gotoxy(1,pos+2);

}

break;

}

}

else if(c==13)

{

char direc[7];

direc[0]=drives[pos];

direc[1]=':';

direc[2]='\0';

statusbar(direc);

openfolder(direc);

clrscr();

statusbar(myc);

textcolor(WHITE);

textbackground(LIGHTBLUE);

newln();

clreol();

cprintf(" %c:",drives[0]);

textbackground(WHITE);

textcolor(BLACK);

gotoxy(1,wherey()+1);

n = strlen(drives);

for(int i=1;i<n;i++)

{

cout<<drives[i]<<":";

gotoxy(1,wherey()+1);

}

pos=0;

gotoxy(1,1);

}

}

}

int print(filei a[255],char direc[255])

{

ffblk dir;

int done;

clrscr();

int n=0;

char NEW[255];

strcpy(NEW,direc);

strcat(NEW,"\\\*.\*");

done = findfirst(NEW,&dir,FA\_DIREC);

while (!done)

{

a[n].copy(dir);

if(a[n].chkname()==1)

n--;

n++;

if(n==255)break;

done = findnext(&dir);

}

return n;

}

void getdrives(char d[50])

{

int dn=0;

for (int disk = 0;disk < 26;++disk)

{

setdisk(disk);

if (disk == getdisk())

d[dn++]=disk+'A';

}

d[dn]='\0';

setdisk(2);

}

void newln()

{

gotoxy(1,wherey()+1);

}

int filemenu()

{

char menu[7][20]={"Move","Delete","Rename","Copy","Paste","New Folder","Exit"};

window(40,2,80,24);

textbackground(LIGHTBLUE);

textcolor(WHITE);

clrscr();

gotoxy(1,2);

textcolor(BLACK);

textbackground(YELLOW);

clreol();

gotoxy(1,2);

cprintf(" %s",menu[0]);

textcolor(WHITE);

textbackground(LIGHTBLUE);

int n=7;

for(int i=1;i<n;i++)

{

gotoxy(1,wherey()+1);

cprintf(" %s",menu[i]);

}

int pos=0;

gotoxy(1,2);

while(1)

{

char c=getch();

if(c==27)

exit(0);

if(c=='\b')

{

window(40,2,80,24);

textbackground(WHITE);

textcolor(BLACK);

clrscr();

window(1,2,80,24);

break;

}

else if(!c)

{

switch(getch())

{

case 80:

if(pos+1!=n)

{

clreol();

cprintf(" %s",menu[pos]);

pos=pos+1;

textbackground(YELLOW);

textcolor(BLACK);

newln();

clreol();

cprintf(" %s",menu[pos]);

textcolor(WHITE);

textbackground(LIGHTBLUE);

gotoxy(1,pos+2);

}

break;

case 72:

if(pos-1!=-1)

{

clreol();

cprintf(" %s",menu[pos]);

pos=pos-1;

textbackground(YELLOW);

textcolor(BLACK);

gotoxy(1,wherey()-1);

clreol();

cprintf(" %s",menu[pos]);

textcolor(WHITE);

textbackground(LIGHTBLUE);

gotoxy(1,pos+2);

}

break;

}

}

else if(c==13)

{

window(40,2,80,24);

textbackground(WHITE);

textcolor(BLACK);

clrscr();

window(1,2,80,24);

if(pos==6)

exit(0);

return pos;

}

}

return -1;

}

void openfolder(char direc[255])

{

copylist copy("C:\\copy.dat"),move("C:\\move.dat");

textbackground(WHITE);

textcolor(BLACK);

clrscr();

filei a[255];

int pos=0;

int n = print(a,direc);

char name[13];

if(!n)

{

cout<<"No files found. Press any key to return to My Computer";

getch();

return;

}

for(int i=0;i<n&&i<20;i++)

{

a[i].getname(name);

if(i!=0)

{

cprintf("%s",name);

gotoxy(1,wherey()+1);

}

else

{

gotoxy(1,2);

a[i].print();

textbackground(LIGHTBLUE);

textcolor(WHITE);

clreol();

cprintf(" %s",name);

gotoxy(1,wherey()+1);

textbackground(WHITE);

textcolor(BLACK);

}

}

gotoxy(1,2);

int down=0,up=0;

textcolor(BLACK);

while(1)

{

char c=getch();

if(c==27) //Exit if escape

exit(0);

if(c=='\b') //Go back if backspace is pressed

{

for(int i=strlen(direc)-1;direc[i]!='\\'&&i>=0;i--);

if(i==-1)

return;

direc[i]='\0';

for(i=strlen(direc)-1;direc[i]!='\\'&&i>=0;i--);

char name[30];

i++;

for(int k=0;direc[i]!='\0';i++,k++)

name[k]=direc[i];

name[k]='\0';

statusbar(name);

textbackground(WHITE);

textcolor(BLACK);

clrscr();

up=down=0;

n=print(a,direc);

for(i=0;i<n&&i<20;i++)

{

a[i].getname(name);

if(i!=0)

{

cprintf("%s",name);

gotoxy(1,wherey()+1);

}

else

{

gotoxy(1,2);

a[i].print();

textbackground(LIGHTBLUE);

textcolor(WHITE);

clreol();

cprintf(" %s",name);

gotoxy(1,wherey()+1);

textbackground(WHITE);

textcolor(BLACK);

}

}

gotoxy(1,2);

pos=0;

}

else if(!c)

{

switch(getch())

{

case 80:

if(pos+1!=n)

{

clreol();

a[pos].getname(name);

cprintf("%s",name);

pos=pos+1;

if(pos+2-down==22)

{

down++;

up--;

gotoxy(1,2);

delline();

gotoxy(1,21);

}

else

{

gotoxy(1,pos+2-down);

}

a[pos].getname(name);

textbackground(LIGHTBLUE);

textcolor(WHITE);

clreol();

cprintf(" %s",name);

a[pos].print();

textcolor(BLACK);

textbackground(WHITE);

gotoxy(1,pos+2-down);

}

break;

case 72: if(pos-1>-1)

{

clreol();

a[pos].getname(name);

cprintf("%s",name);

pos=pos-1;

if(pos+2+up==1)

{

down--;

up++;

gotoxy(1,21);

clreol();

gotoxy(1,2);

insline();

}

else

{

gotoxy(1,pos+2+up);

}

textbackground(LIGHTBLUE);

textcolor(WHITE);

clreol();

a[pos].getname(name);

cprintf(" %s",name);

a[pos].print();

textcolor(BLACK);

textbackground(WHITE);

gotoxy(1,pos+2+up);

}

break;

}

}

else if(c==13)

{

int ch=-1;

if(a[pos].retatt()!=16)

ch=filemenu();

else

ch=-2;

if(ch==-1)

{

gotoxy(1,pos+2+up);

textbackground(LIGHTBLUE);

textcolor(WHITE);

clreol();

a[pos].getname(name);

cprintf(" %s",name);

a[pos].print();

textcolor(BLACK);

textbackground(WHITE);

gotoxy(1,pos+2+up);

continue;

}

window(1,2,80,24);

if(ch==-2)

{

up=down=0;

char name[13];

a[pos].getname(name);

statusbar(name);

if(a[pos].retatt()==16)

{

strcat(direc,"\\"); //make the new directory path

strcat(direc,name);

n=print(a,direc);

if(!n)

{

cout<<"No files found.Press any key to return";

for(int i=strlen(direc)-1;direc[i]!='\\'&&i>=0;i--);

if(i==-1)

return;

direc[i]='\0';

for(i=strlen(direc)-1;direc[i]!='\\'&&i>=0;i--);

char name[30];

i++;

for(int k=0;direc[i]!='\0';i++,k++)

name[k]=direc[i];

name[k]='\0';

statusbar(name);

textbackground(WHITE);

textcolor(BLACK);

clrscr();

up=down=0;

n=print(a,direc);

getch();

}

}

pos=0;

}

else if(ch==0)

{

filep movefile;

char temp[255];

strcpy(temp,direc);

strcat(temp,"\\");

strcat(temp,name);

movefile.setfilei(a[pos]);

movefile.setpath(temp);

move.addtolist(movefile);

}

else if(ch==1)

{

char temp[255];

strcpy(temp,direc);

strcat(temp,"\\");

strcat(temp,name);

remove(temp);

n=print(a,direc);

}

else if(ch==2)

{

char newname[255];

openrename(newname);

strupr(newname);

char temp[255],temp1[255];

strcpy(temp,direc);

strcat(temp,"\\");

strcpy(temp1,temp);

strcat(temp1,newname);

strcat(temp,name);

rename(temp,temp1);

a[pos].setname(newname);

}

else if(ch==3)

{

filep copyfile;

char temp[255];

strcpy(temp,direc);

strcat(temp,"\\");

strcat(temp,name);

copyfile.setfilei(a[pos]);

copyfile.setpath(temp);

copy.addtolist(copyfile);

}

else if(ch==4)

{

char temp[255];

strcpy(temp,direc);

copy.execlist(temp);

strcpy(temp,direc);

move.execmovelist(temp);

n=print(a,direc);

}

else if(ch==5)

{

char temp[255];

openrename(temp);

char temp1[255];

strcpy(temp1,direc);

strcat(temp1,"\\");

strcat(temp1,temp);

mkdir(temp1);

remove(temp1);

n=print(a,direc);

}

textbackground(WHITE);

textcolor(BLACK);

clrscr();

gotoxy(1,2);

for(int i=down;i<n&&i<(20+down);i++)

{

a[i].getname(name);

if(i!=pos)

{

cprintf("%s",name);

gotoxy(1,wherey()+1);

}

else

{

a[i].print();

textbackground(LIGHTBLUE);

textcolor(WHITE);

clreol();

cprintf(" %s",name);

gotoxy(1,wherey()+1);

textbackground(WHITE);

textcolor(BLACK);

}

}

gotoxy(1,pos+2-down);

}

}

}

void border(int c1,int r1,int c2,int r2,int d)

{

window(1,1,80,25);

for(int i=c1;i<=c2;i++)

{ gotoxy(i,r1);

cprintf("²");

delay(d);

}

for(i=r1;i<=r2;i++)

{

gotoxy(c2,i);

cprintf("²");

delay(d);

}

for(i=c2;i>=c1;i--)

{

gotoxy(i,r2);

cprintf("²");

delay(d);

}

for(i=r2;i>=r1;i--)

{

gotoxy(c1,i);

cprintf("²");

delay(d);

}

}

void openrename(char a[255])

{

window(20,10,62,18);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

textcolor(GREEN);

border(20,10,62,18,0);

window(21,11,62,18);

newln();

cout<<" ENTER NAME (NO SPACES) :";

newln();

newln();

for(int i=0;i<=40;i++)

cprintf("²");

textcolor(WHITE);

newln();

newln();

char c[255];

cin>>c;

strcpy(a,c);

window(1,2,80,24);

textcolor(BLACK);

textbackground(WHITE);

clrscr();

}

void statusbar(char up[50])

{

window(1,1,80,1);

textbackground(RED);

textcolor(WHITE);

clrscr();

int x=(80-strlen(up))/2;

gotoxy(x,1);

cout<<up;

x=(40-strlen("Details"))/2;

gotoxy(40+x,1);

cout<<"Details";

window(1,2,80,24);

}

void statusbar2(char up[50])

{

window(1,25,80,25);

textbackground(LIGHTBLUE);

textcolor(WHITE);

clrscr();

int x=(80-strlen(up))/2;

gotoxy(x,1);

cout<<up;

x=(40-strlen("Details"))/2;

gotoxy(40+x,1);

window(1,2,80,24);

}

void sand()

{

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "..\\BGI");

int error=graphresult();

if(error)

cout<<"ERROR";

line(0, 0, getmaxx(), getmaxy());

closegraph();

}

void main()

{

WELCOME();

sand();

\_setcursortype(\_NOCURSOR);

mycomp();

getch();

}

void printname()

{

while(!kbhit())

{

delay(100);

textbackground(BLACK);

textcolor(random(16));

// F

for(int i=0;i<10;i++)

{

gotoxy(3,i+5);

cprintf("²²");

}

gotoxy(4,5);

cprintf("²²²²²");

gotoxy(4,6);

cprintf("²²²²²");

gotoxy(4,10);

cprintf("²²²");

gotoxy(4,9);

cprintf("²²²");

// I

textcolor(random(16));

for(i=0;i<10;i++)

{

gotoxy(10,i+5);

cprintf("²²");

}

//L

textcolor(random(16));

for(i=0;i<10;i++)

{

gotoxy(13,i+5);

cprintf("²²");

}

gotoxy(13,14);

cprintf("²²²²²");

gotoxy(13,13);

cprintf("²²²²²");

//E

textcolor(random(16));

for(i=0;i<10;i++)

{

gotoxy(19,i+5);

cprintf("²²");

}

gotoxy(19,13);

cprintf("²²²²²");

gotoxy(19,14);

cprintf("²²²²²");

gotoxy(19,5);

cprintf("²²²²²");

gotoxy(19,6);

cprintf("²²²²²");

gotoxy(19,9);

cprintf("²²²²²");

gotoxy(19,10);

cprintf("²²²²²");

// -

textcolor(random(16));

gotoxy(25,9);

cprintf("²²²²");

gotoxy(25,10);

cprintf("²²²²");

// M

textcolor(random(16));

for(i=0;i<10;i++)

{

gotoxy(30,i+5);

cprintf("²²");

}

gotoxy(32,5);

cprintf("² ²");

gotoxy(32,6);

cprintf("² ²");

gotoxy(32,7);

cprintf(" ² ²");

gotoxy(32,8);

cprintf(" ² ²");

gotoxy(32,9);

cprintf(" ²");

gotoxy(32,10);

cprintf(" ²");

for(i=0;i<10;i++)

{

gotoxy(37,i+5);

cprintf("²²");

}

//A

textcolor(random(16));

for(i=0;i<10;i++)

{

gotoxy(40,i+5);

cprintf("²²");

}

gotoxy(40,5);

cprintf("²²²²");

gotoxy(40,6);

cprintf("²²²²");

gotoxy(40,10);

cprintf("²²²");

gotoxy(40,9);

cprintf("²²²");

for(i=0;i<10;i++)

{

gotoxy(43,i+5);

cprintf("²²");

}

// N

textcolor(random(16));

for(i=0;i<10;i++)

{

gotoxy(46,i+5);

cprintf("²²");

}

gotoxy(48,13);

cprintf(" ²");

gotoxy(48,14);

cprintf(" ²");

gotoxy(48,11);

cprintf(" ²");

gotoxy(48,12);

cprintf(" ²");

gotoxy(48,5);

cprintf("² ");

gotoxy(48,6);

cprintf("² ");

gotoxy(48,7);

cprintf(" ² ");

gotoxy(48,8);

cprintf(" ² ");

gotoxy(48,9);

cprintf(" ²");

gotoxy(48,10);

cprintf(" ²");

for(i=0;i<10;i++)

{

gotoxy(53,i+5);

cprintf("²²");

}

//A

textcolor(random(16));

for(i=0;i<10;i++)

{

gotoxy(56,i+5);

cprintf("²²");

}

gotoxy(56,5);

cprintf("²²²²");

gotoxy(56,6);

cprintf("²²²²");

gotoxy(56,10);

cprintf("²²²");

gotoxy(56,9);

cprintf("²²²");

for(i=0;i<10;i++)

{ gotoxy(59,i+5);

cprintf("²²");

}

//G

textcolor(random(16));

for(i=0;i<10;i++)

{ gotoxy(62,i+5);

cprintf("²²");

}

gotoxy(62,13);

cprintf("²²²²²");

gotoxy(62,14);

cprintf("²²²²²");

gotoxy(62,5);

cprintf("²²²²²");

gotoxy(62,6);

cprintf("²²²²²");

gotoxy(62,9);

cprintf("²² ²²");

gotoxy(62,10);

cprintf("²² ²²");

gotoxy(62,11);

cprintf("²² ²²");

gotoxy(62,12);

cprintf("²² ²²");

//E

textcolor(random(16));

for(i=0;i<10;i++)

{ gotoxy(68,i+5);

cprintf("²²");

}

gotoxy(68,13);

cprintf("²²²²²");

gotoxy(68,14);

cprintf("²²²²²");

gotoxy(68,5);

cprintf("²²²²²");

gotoxy(68,6);

cprintf("²²²²²");

gotoxy(68,9);

cprintf("²²²²²");

gotoxy(68,10);

cprintf("²²²²²");

// R

textcolor(random(16));

gotoxy(74,13);

cprintf(" ²");

gotoxy(74,14);

cprintf(" ²");

gotoxy(74,11);

cprintf(" ²");

gotoxy(74,12);

cprintf(" ²");

gotoxy(74,5);

cprintf("²²²²²");

gotoxy(74,6);

cprintf("²²²²²");

gotoxy(74,7);

cprintf(" ² ");

gotoxy(74,8);

cprintf(" ² ");

gotoxy(74,9);

cprintf(" ²");

gotoxy(74,10);

cprintf(" ²");

for(i=0;i<10;i++)

{

gotoxy(74,i+5);

cprintf("²²");

}

textcolor(MAGENTA);

gotoxy(25,18);

cprintf("Class 12 Computer Science Project ");

gotoxy(1,1);

cout<<"Running File-Manager version 1.0 by Ruchir Jain\n";

gotoxy(2,23);

textcolor(LIGHTGREEN);

cprintf(" E-MAIL: ");

textcolor(YELLOW);

cprintf("ruchirjain24@gmail.com");

gotoxy(20,25);

textcolor(128);

textbackground(RED);

cprintf("Press any key to continue. Press esc to Exit");

}

if(getch()==27)

exit(1);

}

void progressbar(int i=100)

{

textbackground(BLACK);

clrscr();

window(19,7,71,9);

textbackground(RED);

textcolor(BLACK);

clrscr();

cout<<" Progress...... "<<i<<"%";

window(20,8,70,8);

textbackground(BLACK);

clrscr();

window(20,8,20+i/2,8);

textbackground(GREEN);

clrscr();

textcolor(GREEN);

}

void WELCOME()

{

clrscr();

textbackground(BLACK);

clrscr();

gotoxy(1,1);

cout<<"Loading File-Manager version 1.0 by Ruchir Jain\n";

textcolor(CYAN);

cout<<" ";

for(int i=1;i<79;i++)

cprintf("\_");

for(i=1;i<23;i++)

{

gotoxy(1,i+2);

cprintf("|");

}

for( i=1;i<79;i++)

cprintf("\_");

for(i=1;i<23;i++)

{

gotoxy(80,i+2);

cprintf("|");

}

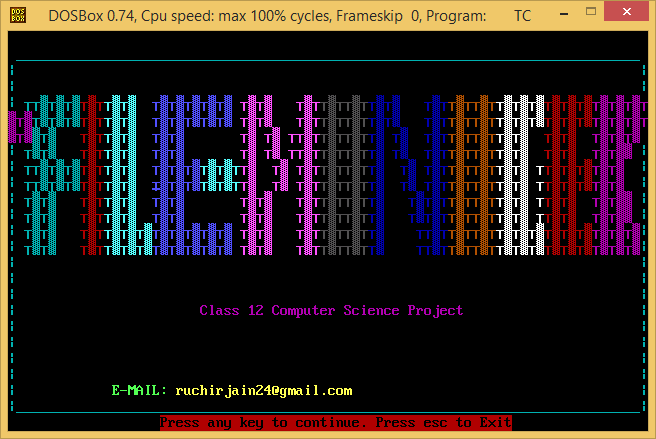
textcolor(BLUE);

printname();

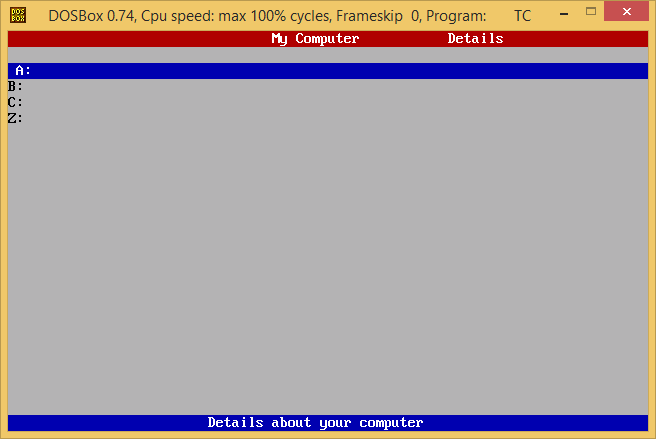
}

*Output Screens*

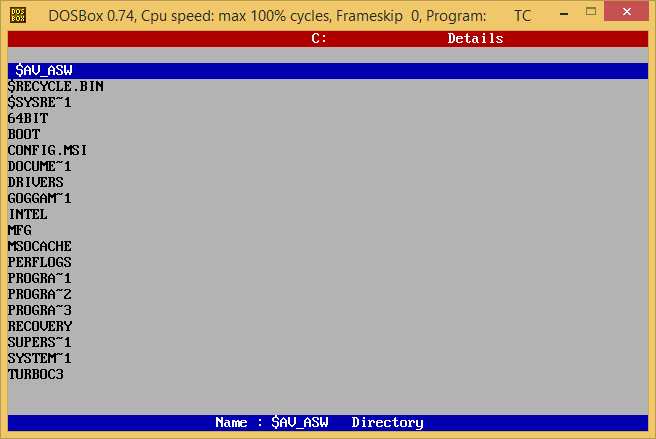
***Welcome Screen***

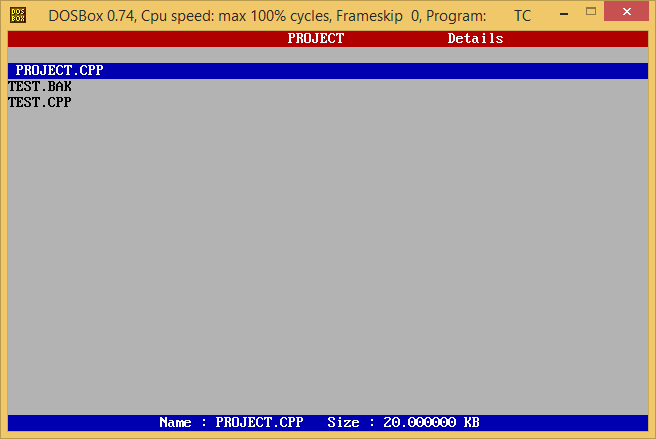
******

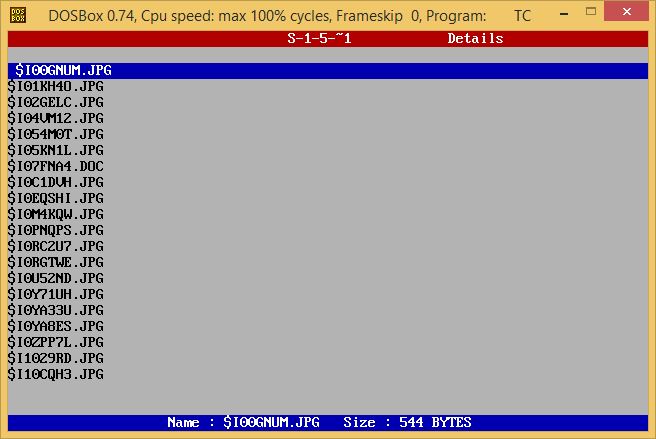
***Main Screen***



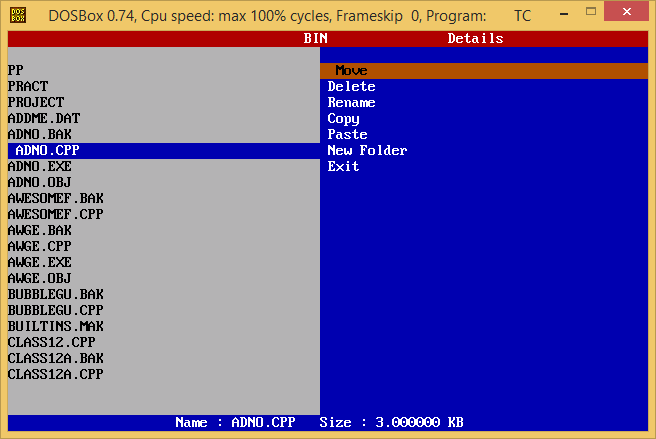
***Opening C: Drive***



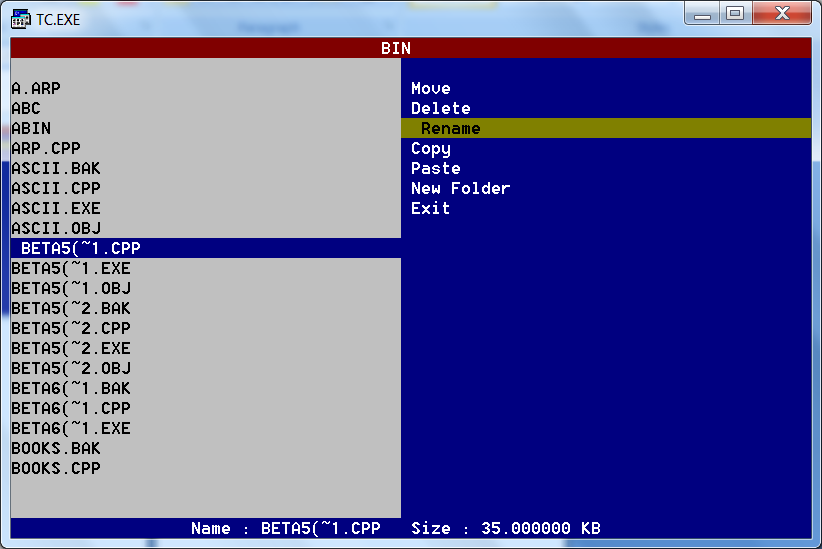
***Displaying Details of files***

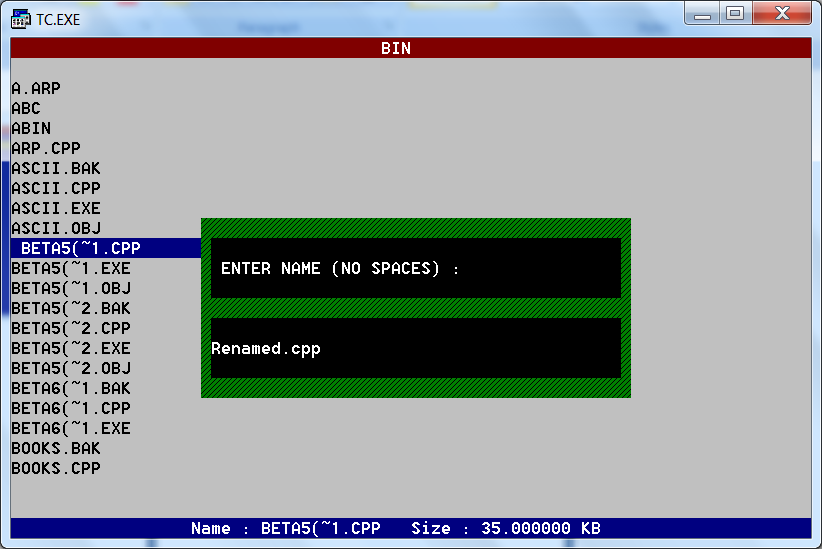


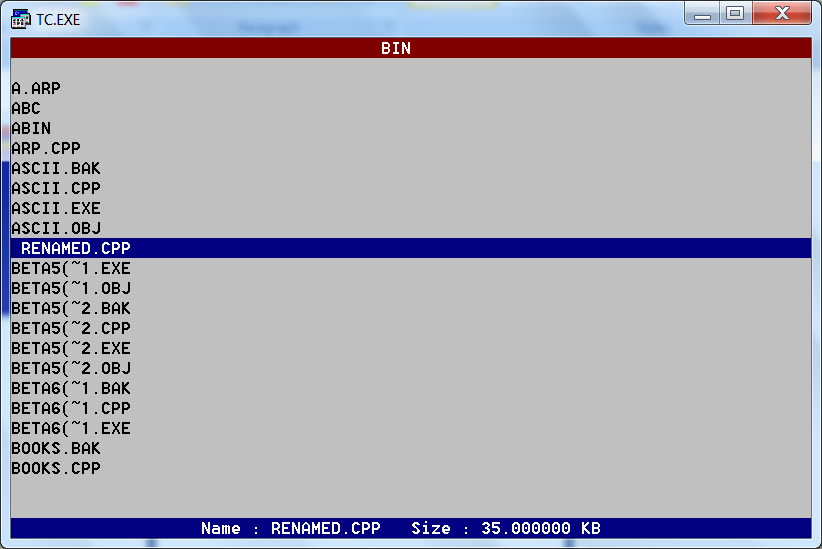
***Options Menu***

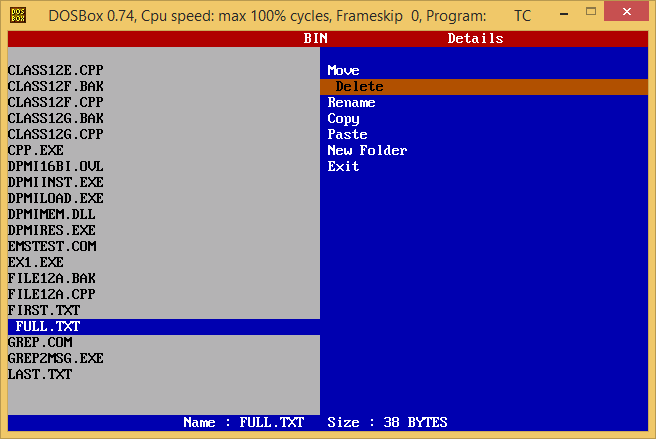


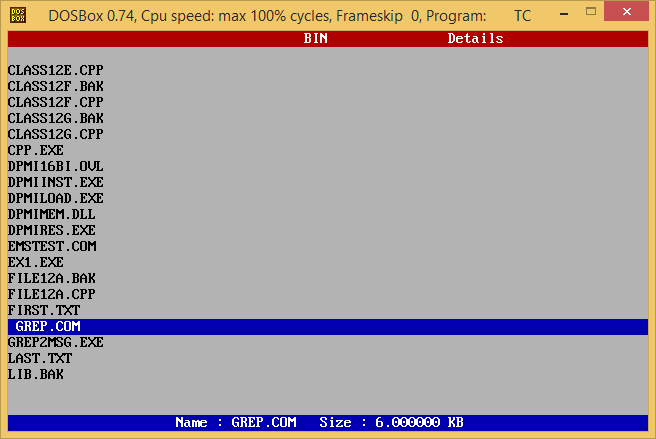
***Renaming File***

******

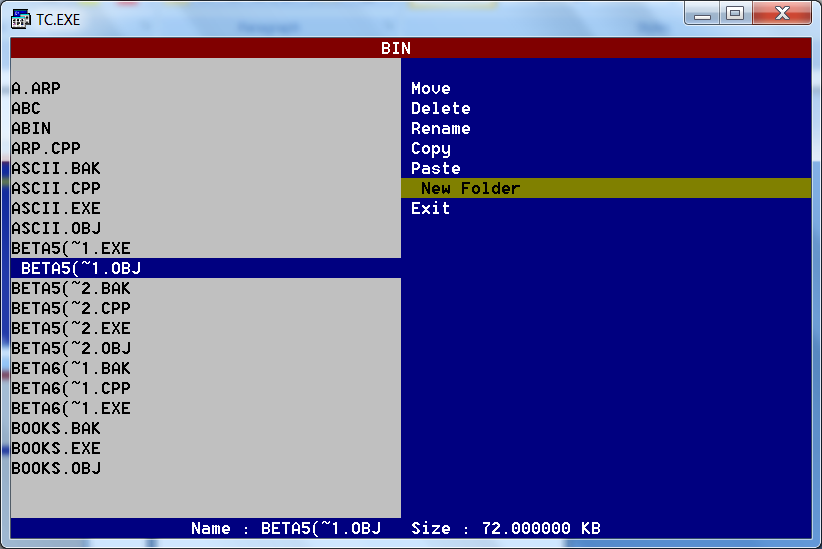


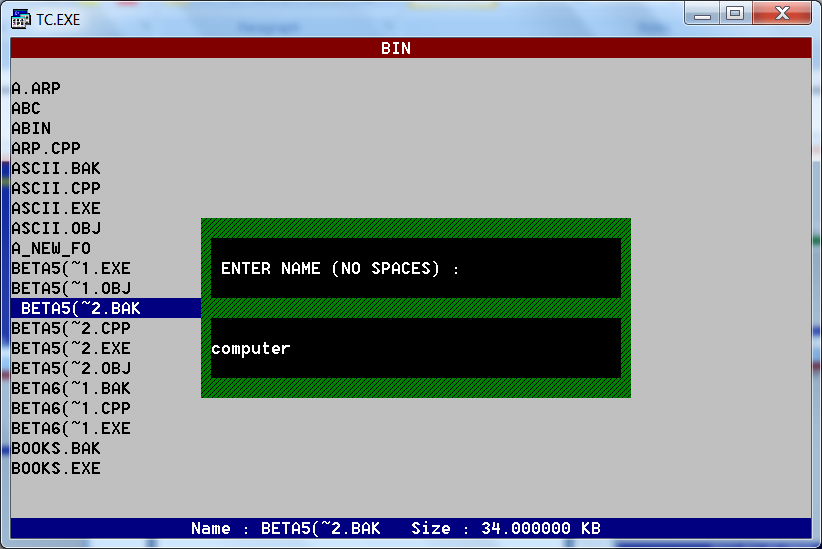


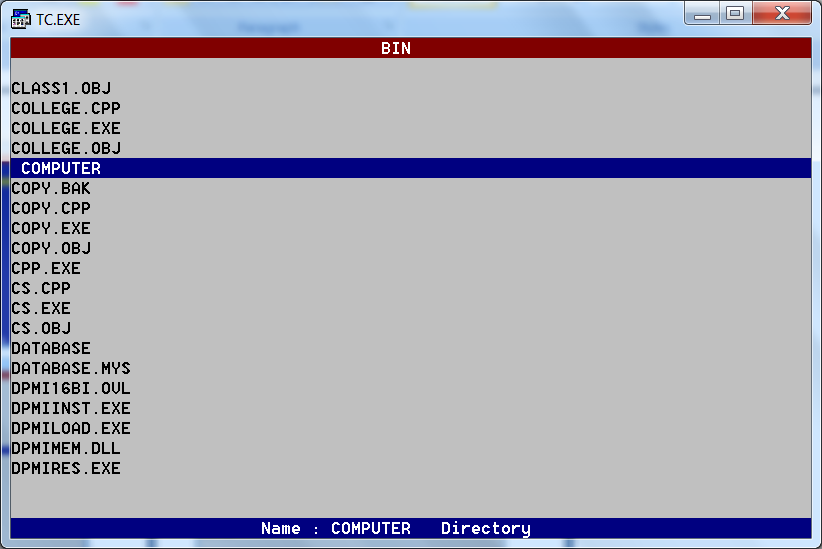
***Deleting file***

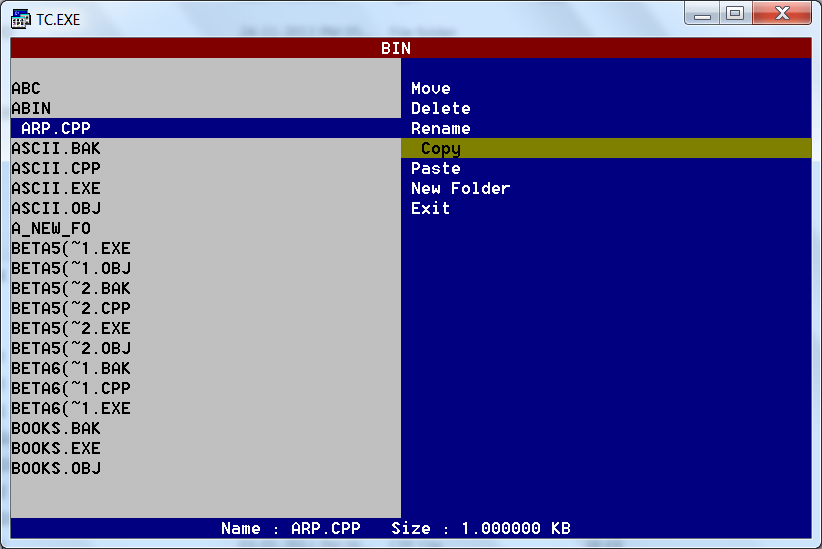


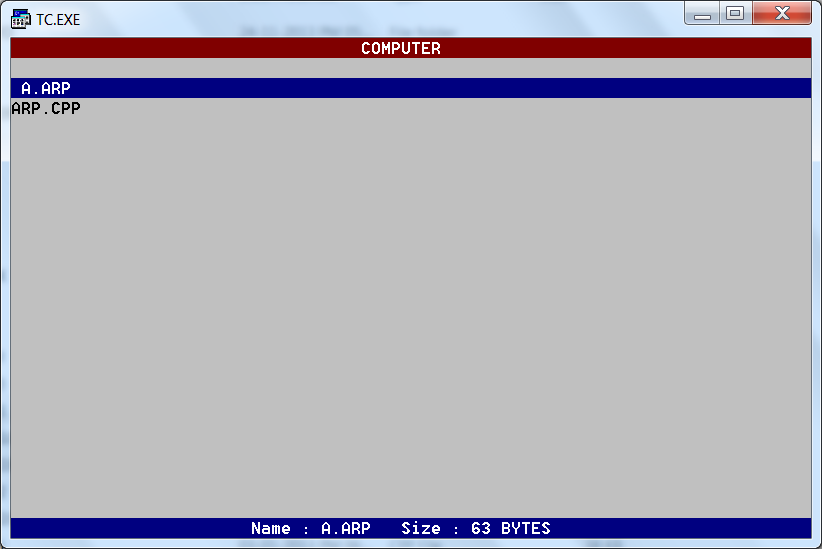
***Creating New Folder***

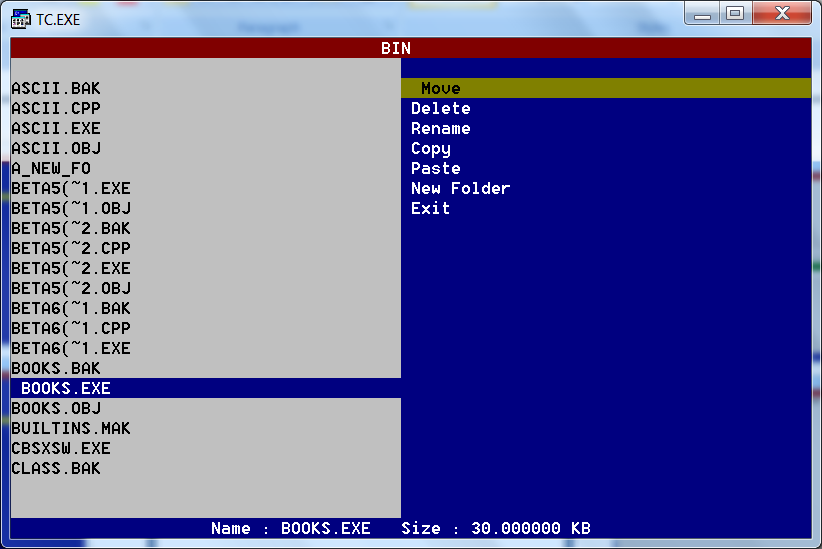


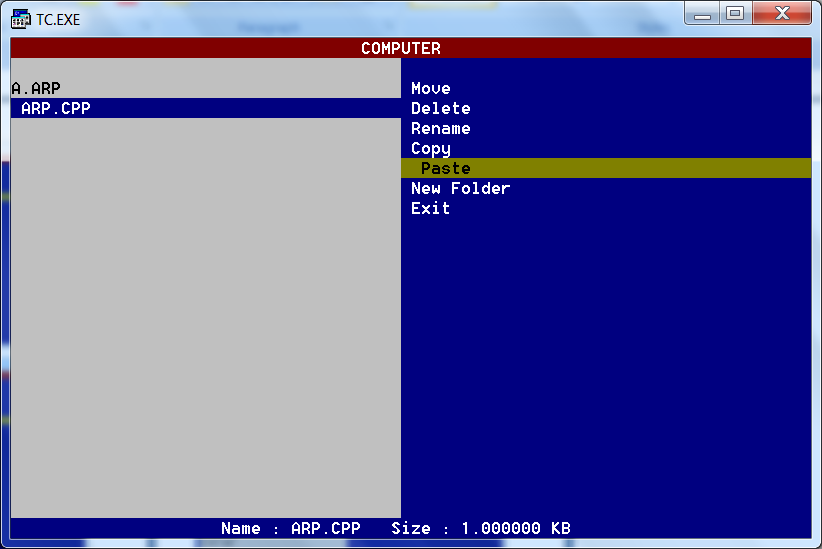


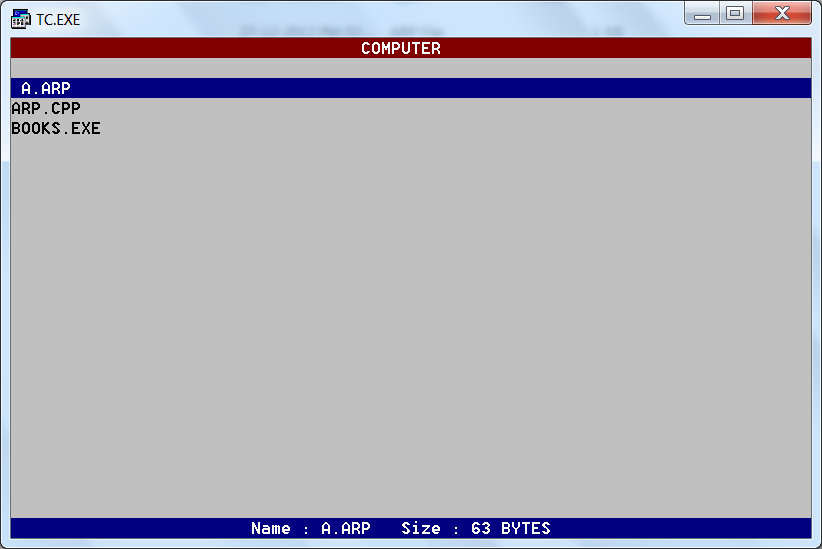


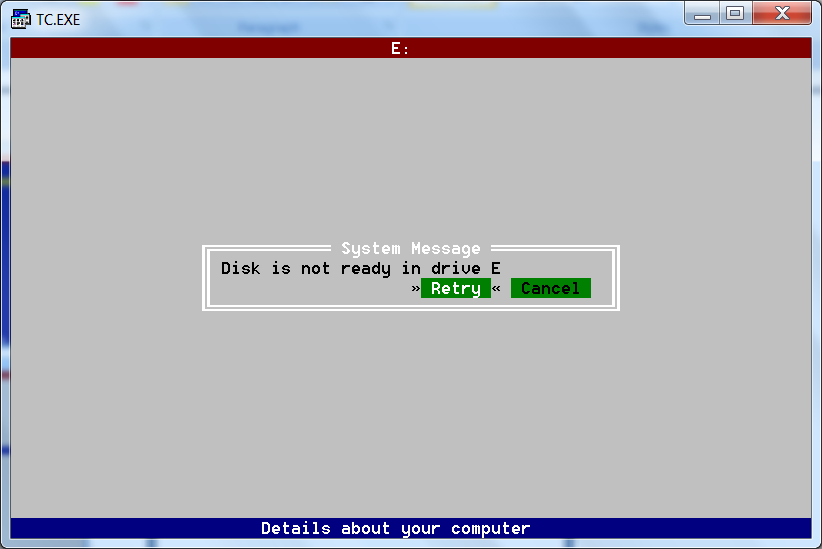
***Copying a file*** 

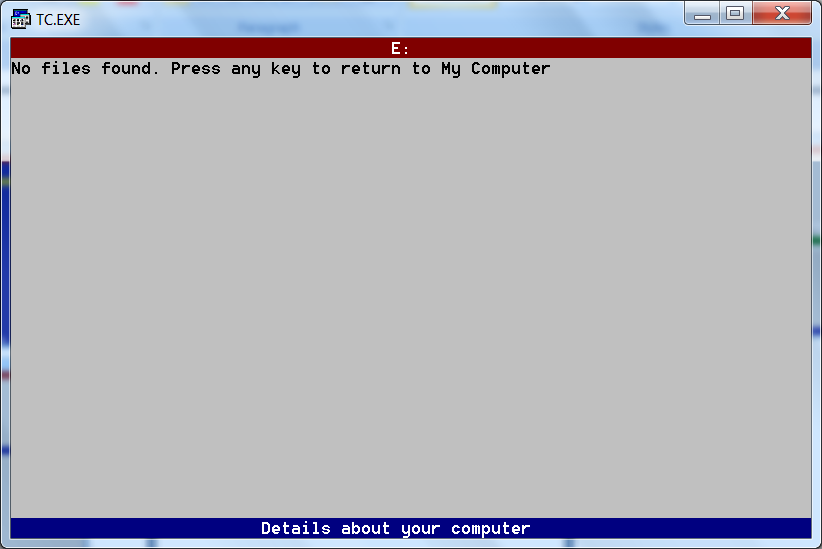


***Moving A File***





***When Drive is Empty/Not Accessible ( eg: When no CD in DVD RW Drive)***



*Acknowledgement*

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely fortunate to have got this all along the completion of my project work. Whatever I have done is only due to such guidance and assistance and I would not forget to thank them.

I respect and thank Mrs. Ritu Nagpal, for giving me an opportunity to do this project and providing me all support and guidance which made me complete the project on time. I am extremely grateful to her for providing such a nice support and guidance.

I owe my profound gratitude to Mr. Mervin Fernandes, who took keen interest on my project work and guided me all along, till the completion of my project work by providing all the necessary information.