**Made By:Himanshu Mehra**

**Class: XII-A**

**Roll No: 24**

***Particulars of the Supervisor***

***Name: Mrs. Renu Manglik***

***Designation: HOD Computer Science***

***Universal Public School***

***ANUBHAV MOHANTY***

*ROLL NO.:*

*CLASS:* ***XII****, SECTION-****A***

*This is to certify that the project entitled “Cell Database and Management”, which deals with statistics related with cell Phones being submitted by* ***Himanshu Mehra****. This project is a bonafide piece of work carried out by me in the consultation with the supervisor.*

*It is with immense pleasure that I find myself writing down my feelings to express my sincere thanks to all those special people who helped me a lot in order to complete this mammoth project.*

*Through this page I would like to extend my gratitude towards my computer science teacher Mrs. Renu Manglik, without whom this project would just had been only a dream. Her essential guidance made the project successful.*

*I would also like to thank my fellow mates whose constructive criticism and valuable suggestion proved to be an icing on the cake in the program.*

Himanshu Mehra

Roll No : 24

New Delhi

Date: 22.12.2014

CONTENTS

1. *System Description at a Glance*
2. *Inputs and outputs*
3. *Problem Description*

* *Requirement Analysis*
* *Hardware and Software Requirements*

1. *System Specification*

* *System Design*
* *Data Design*
* *Architectural Design*
* *Data Dictionary*

1. *File Design Structures*
2. *Procedure/Function Descriptions*
3. *Program Source Code*

***SYSTEM DESCRIPTION AT A GLANCE***

The market for cell phones has increased very significantly over the last decade. The worldwide smartphone market grew 27.2% year over year in the second quarter of 2014 (2014Q3), just over a third of a billion shipments at 335 million units, according to data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker. 8 At this pace, 2014 promises to close at nearly 1.3 billion shipments, with Android taking the lion's share, spread across over 180 tracked vendors. The bulk of these Android players are squarely focused on the sub-US$200 space, driven by demand at increasingly value-oriented consumers. Third quarter shipments were slightly above IDC's forecast and expects iOS share to recover during the holiday peak as existing users upgrade to iPhone 6.

**Inputs are:**

* Cell serial no
* Cell brand
* Cell name
* Cell price
* Cell OS
* Cell Camera
* Ram
* Display
* Processor
* Screen
* Dimensions
* SIM
* Date of Release
* Cennectivity
* GPU
* Battery Backup
* Quantity

***INPUTS***

**Outputs are:**

* Cell serial no
* Cell brand
* Cell name
* Cell price
* Cell OS
* Cell Camera
* Ram
* Display
* Processor
* Screen
* Dimensions
* SIM
* Date of Release
* Cennectivity
* GPU
* Battery Backup
* Quantity
* Total Price after Purchase

***OUTPUTS***

***PROBLEM DESCRIPTIONS***

**Required output of the system**

The computerized system generates the following output.

* The computer should be able to Display all the information about the phone and ask the user about his choice of purchase
* The Program can manage the whole database by inputting and manipulating data in the maintanence section
* Some of the features include deleting, undeleting, packing etc.
* The computer should be able to calculate the final price of the purchased items after discount.

***HARDWARE***

***AND***

***SOFTWARE REQUIREMENTS***

The computer should be able to process the final prices using a simple predefined formula and at the same time, it should also provide a user friendly interface.

All these suggests that the choice of a computer 486 or equivalent to about 250 MB hard disk and about 8 to 16 MB RAM, a line printer or a dot-matrix printer and preferable a laser printer is optimum.

The ability of sophisticated tool is imperative, thus, a C++ is chosen to program the system. Having done this the next step is to carefully analyze the data, identify processing element and design appropriate database file.

Data design is the first of the three design activities that are conducted during system development. The impact of data structure on program structure and procedural complexity leads to data design to have a profound influence on system quality.

All the data objects required by this system are listed below:

* ***Cell.dat***

The project.dat database is to be created on the basis of information collected from various sources. It should contain information related to:

1. Name of the cell phone
2. All the specifications
3. The power to modify,delete,search.
4. Transactions

* ***Graphics.dat***

1. To provide the Homepage

.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Name** | **Where Used** | **Description** |
| 1 | sno | Attributes used for information about cell phones | Integer |
| 2 | brand | 20 character |
| 3 | name | 20 character |
| 4 | Price | integer |
| 5 | OS | 20 character |
| 6 | Camera | Array of 2 integers |
| 7 | RAM | Integer |
| 8 | Display | 20 Characters |
| 9 | Processor | 20 Characters |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Name** | **Where Used** | **Description** |
| 10 | Clock Speed | Attributes used for information about cell phones. | Float |
| 11 | Screen Size | float |
| 12 | Dimesions | Array of float[3] |
| 13 | SIM | 20 character |
| 14 | Date | 20 character |
| 15 | Battery Backup | 30 Characters |

* ***Class Cell:***

|  |  |  |
| --- | --- | --- |
| S.No | Procedure/Function name | Description |
| 1 | void dispall() | Function to display all information |
| 2 | void input() | Function to input cell related data |
| 3 | void dispundel() | Function to display undeleted content |
| 4 | Char\* retcname() | Returns Cell Name |
| 5 | char\* retcbrand() | Function to return Cell Brand |
| 6 | Void del(char) | Function to assign Delmark |
| 7 | Int retsno() | Function to return serial no |
| 8 | Char retdelmark() | Function to return delmark |
| 9 | Char\* retbrand() | Function to return cellbrand |
| 10 | Int retquantity() | Function to return Quantity |
| 11 | Void getquantity(int) | Function to Get Quantity |
| 12 | Float calculateprice() | Function to Calculate Price |

* ***Class u\_id:***

|  |  |  |
| --- | --- | --- |
| S.No | Procedure/Function name | Description |
| 1 | void id() | Compares the user id and password with file |
| 2 | Char\* retuser() | Function to return user id |
| 3 | char\* retpoass() | Function to return password |
| 4 | Void input() | Function to input ID and Password |

* ***class buy :***

|  |  |  |
| --- | --- | --- |
| S.No | Procedure/Function name | Description |
| 1 | void getinvdt() | Function to get invoice no |
| 2 | void getinvno(int) | Function to produce an invoice no |
| 3 | int retqty() | Returns quantity |
| 4 | Void getprice(int) | Gets the price |

* ***The data-members of the classes can be found on the data dictionary page.***

#include<conio.h>

#include<fstream.h>

#include<ctype.h>

#include<stdio.h>

#include<string.h>

extern lol();

void dispdel();

void disp();

void menu();

void signup();

void transactions();

void maintenance();

void purchase();

class u\_id

{

char userid[20];

char pass[20];

public:

char ding;

void id();

char\* retuser();

char\* retpass();

void input();

}og;

char\* u\_id::retuser()

{

return userid;

}

char\* u\_id::retpass()

{

return pass;

}

class cell

{

char delmark;

int sno;

char brand[20];

char name[20];

int price;

char os[20];

int mpixel[2];

int ram;

char display[20];

char pname[20];

float pghz;

float screen;

float finprice;

float dimensions[3];

char sim[20];

char date[10];

char connectivity[20];

char gpu [20];

char battery[30];

int quantity;

public:

void dispall();

void input();

void dispundel();

char\* retcname();

char\* retcbrand();

void del(char);

int retsno();

char retdelmark();

char\* retbrand();

int retquantity();

void getquantity(int);

float calculateprice();

};

void writedata()

{ ofstream ofile("cell.dat", ios::binary| ios::app);

if(!ofile) { cout<<"Error";getch();}

cell ob;

char ans;

do { ob.input();

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

cout<<"Enter more ?";

cin>>ans;

ans=toupper(ans);

} while(ans!='N');

ofile.close();

}

void readdata()

{ ifstream ifile("cell.dat", ios::binary);

if(!ifile) { cout<<"Error"; }

cell ob;

clrscr();

ifile.read((char\*)&ob, sizeof(ob));

while(!ifile.eof())

{ if (ob.retdelmark()!='\*')

ob.dispall();

getch();

ifile.read((char\*)&ob, sizeof(ob));

}

ifile.close();

}

char\* cell::retcname()

{ return name;

}

char\* cell::retbrand()

{

return brand;

}

void cell::getquantity(int a)

{

quantity=a;

}

int cell::retquantity()

{

return quantity;

}

float cell :: calculateprice()

{

finprice= (0.12\*price)+ price;

return finprice;

}

void cell::del(char a)

{

delmark=a;

}

char cell::retdelmark()

{ return delmark;

}

int cell :: retsno()

{

return sno;

}

void cell::input()

{

clrscr();

delmark = ' ';

cout<<"\nEnter Serial Number : ";

cin>>sno;

cout<<"\nEnter Brand : ";

gets(brand);

cout<<"\nEnter Name : ";

gets(name);

cout<<"\nEnter Price : ";

cin>>price;

cout<<"\nEnter qty : ";

cin>>quantity;

cout<<"\nEnter OS : ";

gets(os);

cout<<"\nEnter Megapixel (Back camera) : ";

cin>>mpixel[0];

cout<<"\nEnter Megapixel (Front camera) : ";

cin>>mpixel[1];

cout<<"\nEnter RAM : ";

cin>>ram;

cout<<"\nEnter Display : ";

gets(display);

cout<<"\nEnter Processor : ";

gets(pname);

cout<<"\nEnter Clock Speed : ";

cin>>pghz;

cout<<"\nEnter Screen Size (inch) : ";

cin>>screen;

cout<<"\nEnter length : ";

cin>>dimensions[0];

cout<<"\nEnter breadth : ";

cin>>dimensions[1];

cout<<"\nEnter Width : ";

cin>>dimensions[2];

cout<<"\nEnter Sim Info : ";

gets(sim);

cout<<"\nEnter Date of Release : ";

gets(date);

cout<<"\nEnter connectivity : ";

gets(connectivity);

cout<<"\nEnter Graphic Processor : ";

gets(gpu);

cout<<"\nEnter Battery Life : ";

gets (battery);

}

void cell::dispall()

{

clrscr();

cout<<"Serial Number : "<<sno<<endl;

cout<<"Brand : "<<brand<<endl;

cout<<"Name : "<<name<<endl;

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

cout<<"qty : "<<quantity<<endl;

cout<<"OS : "<<os<<endl;

cout<<"Megapixel (Back camera) : "<<mpixel[0]<<endl;

cout<<"Megapixel (Front camera) : "<<mpixel[1]<<endl;

cout<<"RAM : "<<ram<<endl;

cout<<"Display : "<<display<<endl;

cout<<"Processor : "<<pname<<endl;

cout<<"Clock Speed : "<<pghz<<endl;

cout<<"Screen Size (inch) : "<<screen<<endl;

cout<<"length : "<<dimensions[0]<<endl;

cout<<"breadth : "<<dimensions[1]<<endl;

cout<<"Width : "<<dimensions[2]<<endl;

cout<<"Sim Info : "<<sim<<endl;

cout<<"Date of Release : "<<date<<endl;

cout<<"connectivity : "<<connectivity<<endl;

cout<<"Graphic Processor : "<<gpu<<endl;

cout<<"Battery Life : "<<battery<<endl;

getch();

}

void dispdel()

{ ifstream ifile("cell.dat", ios::binary);

if(!ifile) { cout<<"Error"; }

cell ob;

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

{

if(ob.retdelmark()=='\*')

ob.dispall();

}

ifile.close();

}

void dispundel()

{ ifstream ifile("cell.dat", ios::binary);

if(!ifile) { cout<<"Error"; }

cell ob;

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

{

if(ob.retdelmark()!='\*')

ob.dispall();

}

ifile.close();

}

void disp()

{ int p;

do { clrscr();

cout<<"1. Display All Contents\n";

cout<<"2. Display Deleted\n";

cout<<"3. Display Undeleted\n";

cout<<"4. Exit\n\n";

cout<<"Enter your Choice : ";

cin>>p;

switch(p)

{ case 1:{ readdata();

break;

}

case 2:{ dispdel();

break;

}

case 3:{ dispundel();

break;

}

case 4:{ break;

}

default:{

cout<<"Wrong Choice";

getch();

}

}

} while(p!=4);

}

void delt()

{ fstream ff("cell.dat", ios::in| ios::out| ios::ate| ios::binary);

if (!ff) { cout<<"Error"; getch(); return;}

cell ob;

int a;

char ans;

cout<<"\nEnter sno : ";

cin>>a;

clrscr();

ff.seekp(0,ios::beg);

while(ff.read((char\*)&ob, sizeof(ob)))

{ if(a==ob.retsno() && ob.retdelmark()!='\*')

{ ob.dispall();

cout<<"\nDelete this ?";

cin>>ans;

ans=toupper(ans);

if(ans=='Y')

{ ob.del('\*');

int c=-1\*sizeof(ob);

ff.seekp(c,ios::cur);

ff.write((char\*)&ob, sizeof(ob));

}

}

break;

}

ff.close();

}

void undel()

{

char c[20], d; int f=0;

cell ob;

cout<<"\nEnter model to be undeleted : ";

gets(c);

fstream ff ("cell.dat", ios :: binary|ios::in|ios::out|ios::ate);

if (!ff)

{

clrscr();

cout<<"\nError"; getch();

}

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

while (ff.read((char\*)&ob, sizeof(ob)))

{

if (strcmpi(c,ob.retcname())==0 && ob.retdelmark()!='\*')

{

f=1;

ob.dispall();

gotoxy(32,25);

cout<<"\nDo you Want to undelete? ";

cin>>d;

if (d=='n')

{

break;

}

else

{

ob.del(' ');

int c=-sizeof(ob);

ff.seekp(c,ios::cur);

ff.write((char\*)&ob, sizeof(ob));

}

break;

}

}

ff.close();

if (f==0)

{

clrscr();

cout<<"\nNot Found";

}

getch();

}

void modify()

{clrscr();

char c[20], d; int f=0;

cell ob;

cout<<"\nEnter model to be searched : ";

gets(c);

fstream ff ("cell.dat", ios :: binary|ios::in|ios::out|ios::ate);

if (!ff)

{

clrscr();

cout<<"\nError"; getch();

}

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

while (ff.read((char\*)&ob, sizeof(ob)))

{

if (strcmpi(c,ob.retcname())==0 && ob.retdelmark()!='\*')

{

f=1;

ob.dispall();

cout<<"\nDo you Want to Modify? ";

cin>>d;

if (d=='n')

{

break;

}

else

{

ob.input();

int c=-1\*sizeof (ob);

ff.seekg(c, ios::cur);

ff.write ((char\*)&ob, sizeof (ob));

}

break;

}

}

ff.close();

if (f==0)

{

clrscr();

cout<<"\nNot Found";

getch();

}

getch();

}

void pack()

{

ifstream ifile("cell.dat", ios::binary|ios::in);

if(!ifile)

{

clrscr();

cout<<"\nError Opening File";

getch();

return;

}

cell ob;

ofstream ofile("temp.dat",ios::binary|ios::out);

if(!ofile)

{ clrscr();

cout<<"\nError Opening"; getch(); return; }

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

{ if(ob.retdelmark()!='\*')

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

}

ifile.close();

ofile.close();

remove("cell.dat");

rename("temp.dat","cell.dat");

cout<<"----Success-----";

getch();

}

//..........................SEARCHING.................................

void sno()

{

ifstream ifile("cell.dat", ios::binary);

cell ob;

if(!ifile)

{

cout<<"Error";

getch();

}

int sno,f=0;

cout<<"\nEnter serial no : ";

cin>>sno;

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

{ if(ob.retsno()==sno)

{

ob.dispall();

f=1;

break;

}

}

if(f==0)

cout<<"\nSearch Unsuccessful";

getch();

}

void snm()

{ clrscr();

ifstream ifile("cell.dat", ios::binary);

cell ob;

if(!ifile)

{clrscr();

cout<<"Error"; getch();

}

char nm[9]; int f=0;

cout<<"\nEnter Phone Name : ";

gets(nm);

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

{ if(strcmpi(ob.retcname(),nm)==0)

{ ob.dispall();

f=1;

break;

}

}

if(f==0)

cout<<"\nSearch unsuccessful";

getch();

}

void sgrp()

{

clrscr();

ifstream ifile("cell.dat", ios::binary);

cell ob;

if(!ifile) { cout<<"Error"; }

char bp[4]; int f=0;

cout<<"\nEnter Cell Brand : ";

gets(bp);

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

{ if(strcmpi(ob.retbrand(),bp)==0)

{ ob.dispall();

f=1;

break;

}

}

if(f==0)

cout<<"\nSearch unsuccessful";

getch();

}

void search()

{ int x;

do { clrscr();

cout<<"1. Search by No\n";

cout<<"2. Search by Name\n";

cout<<"3. Exit\n\n";

cout<<"Your Choice : ";

cin>>x;

switch(x)

{ case 1:{ sno();

getch();

break;

}

case 2:{ snm();

getch();

break;

}

case 3:{ break;

}

default:{

cout<<"Wrong Choice";

}

}

} while(x!=3);

}

//....................................................................

void menu()

{

int a;

do { clrscr();

cout<<"1.Input\n";

cout<<"2.Display\n";

cout<<"3.Maintenance\n";

cout<<"4.Buy\n";

cout<<"5.Exit\n\n";

cout<<"Enter your Choice : ";

cin>>a;

switch(a)

{ case 1:{ writedata();

break;

}

case 2:{ disp();

getch();

break;

}

case 3:{

maintenance();

getch();

break;

}

case 4:{

purchase();

getch();

break;

}

case 5:{

break;

}

default:{

cout<<"Wrong Choice";

break;;

}

}

}while(a!=5);

}

void maintenance()

{

int ch;

cell ob;

do

{clrscr();

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

cout<<"2.Modify"<<endl;

cout<<"3.Delete"<<endl;

cout<<"4.Undelete"<<endl;

cout<<"5.Pack"<<endl;

cout<<"6.Exit"<<endl;

cout<<"\n\nEnter Your Choice : ";

cin>>ch;

switch(ch)

{

case 1:{ search();

getch();

break;

}

case 2:{ modify();

getch();

break;

}

case 3:{

delt();

break;

}

case 4:{

undel();

break;

}

case 5:{ pack();

break;

}

case 6:{ return;

}

default:{

cout<<"Wrong Choice"; getch();

}

}

} while (ch!=6);

}

void u\_id:: id()

{ clrscr();

char iid[20];

char passs[5];

int f=0;

ifstream ifile("id.dat", ios::binary | ios::in);

if(!ifile) { clrscr();

gotoxy(32,17);

cout<<"Error"; getch(); return;}

gotoxy(32,6);

cout<<"Enter ID : ";

gets(iid);

cout<<endl;

cout<<endl;

gotoxy(32,9);

cout<<"Enter Password : ";

int i=0,r=0;

do

{

passs[i]=getch();

gotoxy(50+r,9);

if (passs[i]==13)

break;

cout<<"\*";

i++;

r++;

}while(i<20);

passs[i]='\0';

ifile.read((char\*)&og, sizeof (og));

while(!ifile.eof())

{

if(strcmpi(og.retuser(),iid)==0 && strcmpi(og.retpass(), passs)==0)

{

menu();

f=1;

break;

}

if(f==0)

{

cout<<endl;

gotoxy(32,17);

cout<<"Wrong Id or Password\n";

}

ifile.read((char\*)&og, sizeof (og));

}

ifile.close();

getch();

}

void signup()

{

ofstream ofile("id.dat", ios::binary| ios::app);

if(!ofile)

{

clrscr();

clrscr();

cout<<"Error"; }

char ans;

og.input();

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

cout<<"Signup Successful";

getch();

ofile.close();

}

void u\_id :: input()

{

int k;

gotoxy(32,6);

cout<<"\nEnter User Name : ";

gets(userid);

gotoxy(32,8);

cout<<"\nPassword : ";

gets(pass);

getch();

}

void main()

{

lol();

int a;

do

{

clrscr();

gotoxy(33,7);

cout<<"MENU"<<endl<<endl;

gotoxy(33,10);

cout<<"1. Login"<<endl;

gotoxy(33,11);

cout<<"2. Sign Up"<<endl;

gotoxy(33,12);

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

gotoxy(33,15);

cout<<"Enter Your Choice : ";

cin>>a;

switch (a)

{

case 1 :

{

og.id();

break;

}

case 2 :

{

signup();

break;

}

case 3:

{

break;

}

default :

{

gotoxy(33,17);

cout<<"Wrong Input";

getch();

}

}

}while(a!=3);

}

/\*-----------------Transaction---------------------\*/

class buy

{

int invno;

char invdt[25];

char qty;

int price;

public:

void getinvdt();

void getinvno(int);

int retqty();

void getprice(int);

};

int buy::retqty()

{

return qty;

}

void buy::getprice(int x)

{

price=x;

}

void buy::getinvno(int s)

{

invno=s;

}

void buy::getinvdt()

{

gets(invdt) ;

}

void transactions()

{ buy os;

cell ob;

ofstream ofile("buy.dat",ios::binary|ios::app);

if(!ofile)

{

cout<<"File not found";

getch();

return;

}

int inv =ofile.tellp()/sizeof(os);

inv++;

fstream ff("cell.dat",ios::binary|ios::ate|ios::in|ios::out);

if(!ff)

{

cout<<"File not found";

getch();

return;

}

char ans;

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

int b,q;

clrscr();

cout<<"\nMobile Serial Number you want to buy:";

cin>>b;

int f =0;

while(ff.read( (char\*)&ob,sizeof(ob)))

{

if(b==ob.retsno())

{

ob.dispall();

f=1;

float a;

clrscr();

cout<<"\nEnter Quantity : ";

cin>>q;

if(ob.retquantity()< q)

{

cout<<"\nQuantity Left :"<<ob.retquantity();

cout<<"\nPlease Enter A Lesser Number Or Try Again Later";

getch();

ff.close();

ofile.close();

return;

}

float ab=q \* ob.calculateprice();

os.getprice(ab);

cout<<"\nTotal Price : "<<ab;

cout<<"\nCONFIRM ORDER (Y/N) : ";

cin>>ans;

ans=toupper(ans);

if(ans=='Y')

{

int e= ob.retquantity()-q;

ob.getquantity(e);

os.getinvno(inv);

cout<<"\nInvoice Number is : "<<inv<<endl;

cout<<"\nEnter date : ";

os.getinvdt();

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

int c=-1\*sizeof(ob);

ff.seekp(c,ios::cur);

ff.write((char\*) & ob,sizeof(ob));

cout<<"\nHappy Shopping :-)";

}

else

{

cout<<"\nYou cancelled your order :(";

return;

}

}

}

if(f == 0 )

{

cout<<"\nYour Mobile code did not match.";

getch();

return;

}

ofile.close();

ff.close();

}

//\_\_\_\_\_\_\_\_\_\_\_MAIN FUNCTION\_\_\_\_\_\_\_\_\_\_\_\_\_

void purchase()

{

int a;

buy os;

do {

clrscr();

cout<<"1. Buy a Mobile Phone "<<endl;

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

cout<<"ENTER YOUR CHOICE : ";

cin>>a;

switch(a)

{

case 1 :{ transactions();

getch();

break;

}

case 2 :{

break;

}

default:{

cout<<"Wrong choice...!!";

getch();

break;

}

}

}while(a!=2);

}

#include <graphics.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include<dos.h>

int lol(void)

{

/\* request auto detection \*/

int gdriver = DETECT, gmode, errorcode;

/\* initialize graphics mode \*/

initgraph(&gdriver, &gmode, "c:\\tc\\bgi");

/\* read result of initialization \*/

errorcode = graphresult();

if (errorcode != grOk) /\* an error occurred \*/

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1); /\* return with error code \*/

}

rectangle(100,50, 240, 400);

rectangle(140,60,200,70);

rectangle(110,80, 230,370);

circle(170,385,10);

setcolor(WHITE);

outtextxy(460,420,"LOADING");

settextstyle(1,0,2);

outtextxy(270,50, "WELCOME TO UNIVERSAL MOBILES");

circle(430,200, 40);

ellipse(430,200,30,390,60,25);

ellipse(430,200,30,390,25,70);

settextstyle(3,0,1);

outtextxy (330,300, "Press Enter to continue...");

for(int i=0; i<150; i++)

{setcolor(RED);

rectangle(460,440,460+i,460);

delay(20);

}

/\* clean up \*/

getch();

closegraph();

return 0;

}















