

INVENTORY MANAGEMENT SYSTEM

The project aims at developing an inventory management system using the C language that enables an organization to maintain its inventory.

The project demonstrates the creation of a user interface of a system, without the use of C Graphics library. The application uses basic C functions to generate menus, show message boxes and print text on the screen. To display customized text with colors and fonts according to application requirements, functions have been created in the application, which fetch the exact video memory addresses of a target location, to write text at the particular location.

The application also implements the concept of structures to define the inventory items. It also effectively applies the various C concepts, such as file operations, looping and branching constructs and string manipulation functions.

```
*****
```

Application: Inventory Management System
 Compiled on: Borland Turbo C++ 3.0

```
*****
```

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <graphics.h>
#include <string.h>

#define TRUE 1
#define FALSE 0

/* List of Global variables used in the application*/
int mboxbrdrclr,mboxbgclr,mboxfgclr;           /* To set colors for all message boxes in
                                                       the application*/
int menutxtbgclr,menutxtfgclr,appframeclr;      /* To set the frame and color's for menu
                                                       items's*/
int section1_symb,section1_bgclr,section1_fgclr; /* To set color of section 1, the region
                                                       around the menu options*/
int section2_symb,section2_bgclr,section2_fgclr; /* To set color of section 2, the section
                                                       on the right of the menu options*/
int fEdit;
int animcounter;

static struct struct_stock                      /* Main database structure*/
{
  char itemcode[8];
  char itemname[50];
  float itemrate;
  float itemqty;
  int minqty;                                /*Used for Reorder level, which is the
                                               minimum no of stock*/
} inv_stock;

struct struct_bill
{
  char itemcode[8];
  char itemname[50];
```

```

float itemrate;
float itemqty;
float itemtot;
}item_bill[100];

char password[8];

const long int stocksize=sizeof(inv_stock); /*stocksize stores the size of the
                                             struct_stock*/
float tot_investment;
int numItems; /*To count the no of items in the stock*/
int button,column,row; /*To allow mouse operations in the application*/

FILE *dbfp; /*To perform database file operations on
                 "inv_stock.dat"*/

int main(void)
{
    float issued_qty;
    char userchoice,code[8];
    int flag,i,itemsold;
    float getInvestmentInfo(void);
    FILE *ft;
    int result;
    getConfigration();

/* Opens & set 'dbfp' globally so that it is accessible from anywhere in the application*/
    dbfp=fopen("d:\invstoc.dat","r+");
    if(dbfp==NULL)
    {
        clrscr();
        printf("\nDatabase does not exists.\nPress Enter key to create it. To exit, press any
              other key.\n ");
        fflush(stdin);
        if(getch()==13)
        {
            dbfp=fopen("d:\invstoc.dat","w+");
            printf("\nThe database for the application has been created.\nYou must restart the
                  application.\nPress any key to continue.\n");
            fflush(stdin);
            getch();
            exit(0);
        }
        else
    }
}

```

```

    {
        exit(0);
    }
}

/* Application control will reach here only if the database file has been opened successfully*/
if(initmouse() == 0)
    messagebox(10,33,"Mouse could not be loaded.", "Error ",'
',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
showmouseptr();
_setcursortype(_NOCURSOR);

while(1)
{
    clrscr();
    fEdit=FALSE;
    ShowMenu();
    numItems=0;
    rewind(dbfp);

/* To calculate the number of records in the database*/
while(fread(&inv_stock,stocksize,1,dbfp)==1)
    ++numItems;
    textcolor(menutxtfgclr);
    textbackground(menutxtbgclr);
    gotopos(23,1);
    cprintf("Total Items in Stock: %d",numItems);
    textcolor(BLUE);
    textbackground(BROWN);
    fflush(stdin);

/*The application will wait for user response */
userchoice=getUserResponse();
switch(userchoice)
{
    /* To Close the application*/
    case '0':
BackupDatabase(); /*Backup the Database file to secure data*/
flushall();
fclose(dbfp);
fcloseall();
print2screen(12,40,"Thanks for Using the application.",BROWN,BLUE,0);
sleep(1);
}

```

```

setdefaultmode();
exit(0);

/* To Add an item*/
case '1':
if(getdata()==1)
{
fseek(dbfp,0,SEEK_END);
/*Write the item information into the database*/
fwrite(&inv_stock,stocksize,1,dbfp);
print2screen(13,33,"The item has been successfully added. ",BROWN,BLUE,0);
getch();
}
break;

/* To edit the item information*/
case '2':
print2screen(2,33,"Enter Item Code>",BROWN,BLUE,0);gotopos(2,54);fflush(stdin);
scanf("%s",&code);
fEdit=TRUE;
if(CheckId(code)==0)
{
if(messagebox(0,33,"Press Enter key to edit the item.", "Confirm",
',mboxbrdrclr,mboxbgclr,mboxfgclr,0)!=13)
{
messagebox(10,33,"The item information could not be modified. Please try
again.", "Edit ", ' ',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
fEdit=FALSE;
break;
}
fEdit=TRUE;
getdata();
fflush(stdin);
fseek(dbfp,-stocksize,SEEK_CUR);
fwrite(&inv_stock,stocksize,1,dbfp);
}
else
messagebox(10,33,"The item is not available in the database.", "No records found",
',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
fEdit=FALSE;
break;
}

```

```

/* To show information about an Item*/
case '3':
print2screen(2,33,"Enter Item Code: ",BROWN,BLUE,0);gotopos(2,55);fflush(stdin);
scanf("%s",&code);
flag=0;
rewind(dbfp);
while(fread(&inv_stock,stocksize,1,dbfp)==1)
{
    if(strcmp(inv_stock.itemcode,code)==0)
    {
        DisplayItemInfo();
        flag=1;
    }
}
if(flag==0)
    messagebox(10,33,"The item is not available.", "No records found ",'
',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
break;

/* To show information about all items in the database*/
case '4':
if(numItems==0)
    messagebox(10,33,"No items are available. ","Error ",'
',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
textcolor(BLUE);
textbackground(BROWN);
gotopos(3,33);
cprintf("Number of Items Available in Stock: %d",numItems);
gotopos(4,33);
getInvestmentInfo();
cprintf("Total Investment :Rs.%2f",tot_investment);
gotopos(5,33);
cprintf("Press Enter To View. Otherwise Press Any Key...");fflush(stdin);
if(getch()==13)
{
    rewind(dbfp);
    while(fread(&inv_stock,stocksize,1,dbfp)==1); /*List All records*/
        DisplayItemRecord(inv_stock.itemcode);
}
textcolor(BLUE);
break;

```

```

/* To issue Items*/
case '5':
    itemsold=0;
    i=0;
    top:
print2screen(3,33,"Enter Item Code: ",BROWN,BLUE,0);fflush(stdin);gotopos(3,55);
scanf("%s",&code);
if(CheckId(code)==1)
    if(messagebox(10,33,"The item is not available.", "No records found ",'
                  ',mboxbrdrclr,mboxbgclr,mboxfgclr,0)==13)
        goto top;
else
    goto bottom;
rewind(dbfp);
while(fread(&inv_stock,stocksize,1,dbfp)==1)
{
    if(strcmp(inv_stock.itemcode,code)==0) /*To check if the item code is available in
                                               the database*/
    {
        issued_qty=IssueItem();
        if(issued_qty > 0)
        {
            itemsold+=1;
            strcpy(item_bill[i].itemcode,inv_stock.itemcode);
            strcpy(item_bill[i].itemname,inv_stock.itemname);
            item_bill[i].itemqty=issued_qty;
            item_bill[i].itemrate=inv_stock.itemrate;
            item_bill[i].itemtot=inv_stock.itemrate*issued_qty;
            i+=1;
        }
        print2screen(19,33,"Would you like to issue another item(Y/
                           N)?",BROWN,BLUE,0);fflush(stdin);gotopos(19,45);
        if(toupper(getch())=='Y')
            goto top;
        bottom:
        break;
    }
}
break;

/* Items to order*/
case '6':
if(numItems<=0)

```

```

{
    messagebox(10,33,"No items are available. ","Items Not Found ",'
               ',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
    break;
}

print2screen(3,33,"Stock of these items is on the minimum
level:",BROWN,RED,0);fflush(stdin);

flag=0;
fflush(stdin);
rewind(dbfp);
while(fread(&inv_stock,stocksize,1,dbfp)==1)
{
    if(inv_stock.itemqty <= inv_stock.minqty)
    {
        DisplayItemInfo();
        flag=1;
    }
}
if(flag==0)
    messagebox(10,33,"No item is currently at reorder level. ","Reorder Items",'
               ',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
    break;

default:
messagebox(10,33,"The option you have entered is not available. ","Invalid Option ",'
               ',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
break;
}
}
}

/*Display Menu & Skins that the user will see*/
ShowMenu()
{
    if(section1_bgclr != BROWN || section1_symb != ' ')
        fillcolor(2,1,23,39,section1_symb,section1_bgclr,section1_fgclr,0);
    if(section2_bgclr != BROWN || section2_symb != ' ')
        fillcolor(2,40,23,79,section2_symb,section2_bgclr,section2_fgclr,0);
    print2screen(2,2,"1: Add an Item",menutxtbgclr,menutxtfgclr,0);
    print2screen(4,2,"2: Edit Item Information",menutxtbgclr,menutxtfgclr,0);
    print2screen(6,2,"3: Show Item Information",menutxtbgclr,menutxtfgclr,0);
    print2screen(8,2,"4: View Stock Report",menutxtbgclr,menutxtfgclr,0);
    print2screen(10,2,"5: Issue Items from Stock",menutxtbgclr,menutxtfgclr,0);
}

```

```

print2screen(12,2,"6: View Items to be Ordered ",menutxtbgclr,menutxtfgclr,0);
print2screen(14,2,"0: Close the application",menutxtbgclr,menutxtfgclr,0);

htskin(0,0,' ',80,appframeclr,LIGHTGREEN,0);
htskin(1,0,' ',80,appframeclr,LIGHTGREEN,0);
vtskin(0,0,' ',24,appframeclr,LIGHTGREEN,0);
vtskin(0,79,' ',24,appframeclr,LIGHTGREEN,0);
htskin(24,0,' ',80,appframeclr,LIGHTGREEN,0);
vtskin(0,31,' ',24,appframeclr,LIGHTGREEN,0);
return;
}

/*Wait for response from the user & returns choice*/
getUserResponse()
{
    int ch,i;
    animcounter=0;

    while(!kbhit())
    {
        getmousepos(&button,&row,&column);

        /*To show Animation*/
        BlinkText(0,27,"Inventory Management System",1,YELLOW,RED,LIGHTGRAY,0,50);
        animcounter+=1;

        i++;
        if(button==1 && row==144 && column>=16 && column<=72) /*Close*/
            return('0');
        if(button==1 && row==16 && column>=16 && column<=136) /*Add New Item*/
            return('1');
        if(button==1 && row==32 && column>=16 && column<=144) /*Edit Item*/
            return('2');
        if(button==1 && row==48 && column>=16 && column<=160) /*Show an Item*/
            return('3');
        if(button==1 && row==64 && column>=16 && column<=104) /*Stock Report*/
            return('4');
        if(button==1 && row==80 && column>=16 && column<=144) /*Issue an Item*/
            return('5');
        if(button==1 && row==96 && column>=16 && column<=152) /*Items to order*/
            return('6');
    }
}

```

```

ch=getch();
return ch;
}

/*Reads a valid id and its information,returns 0 if id already exists*/
getdata()
{
    char tmp[8];
    float tst;
    _setcursortype(_NORMALCURSOR);
    print2screen(3,33,"Enter Item Code: ",BROWN,BLUE,0);fflush(stdin);gotopos(3,53);
    scanf("%s",&tmp);
    if(CheckId(tmp)==0 && fEdit == FALSE)
    {
        messagebox(10,33,"The id already exists. ","Error ','",
                   ',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
        return 0;
    }
    strcpy(inv_stock.itemcode,tmp); /*Means got a correct item code*/
    print2screen(4,33,"Name of the Item: ",BROWN,BLUE,0);fflush(stdin);gotopos(4,53);
    gets(inv_stock.itemname);
    print2screen(5,33,"Price of Each Unit: ",BROWN,BLUE,0);fflush(stdin);gotopos(5,53);
    scanf("%f",&inv_stock.itemrate);
    print2screen(6,33,"Quantity: ",BROWN,BLUE,0);fflush(stdin);gotopos(6,53);
    scanf("%f",&inv_stock.itemqty);
    print2screen(7,33,"Reorder Level: ",BROWN,BLUE,0);fflush(stdin);gotopos(7,53);
    scanf("%d",&inv_stock.minqty);
    _setcursortype(_NOCURSOR);
    return 1;
}

/*Returns 0 if the id already exists in the database, else returns 1*/
int CheckId(char item[8])
{
    rewind(dbfp);
    while(fread(&inv_stock,stocksize,1,dbfp)==1)
        if(strcmp(inv_stock.itemcode,item)==0)
            return(0);
    return(1);
}

/*Displays an Item*/
DisplayItemRecord(char idno[8])

```

```
{  
    rewind(dbfp);  
    while(fread(&inv_stock, stocksize, 1, dbfp)==1)  
        if(strcmp(idno, inv_stock.itemcode)==0)  
            DisplayItemInfo();  
    return;  
}  
  
/*Displays an Item information*/  
DisplayItemInfo()  
{  
    int r=7;  
    textcolor(menutxtfgclr);  
    textbackground(menutxtbgclr);  
    gotopos(r,33);  
    cprintf("Item Code: %s", "");  
    gotopos(r,33);  
    cprintf("Item Code: %s", inv_stock.itemcode);  
    gotopos(r+1,33);  
    cprintf("Name of the Item: %s", "");  
    gotopos(r+1,33);  
    cprintf("Name of the Item: %s", inv_stock.itemname);  
    gotopos(r+2,33);  
    cprintf("Price of each unit: %.2f", "");  
    gotopos(r+2,33);  
    cprintf("Price of each unit: %.2f", inv_stock.itemrate);  
    gotopos(r+3,33);  
    cprintf("Quantity in Stock: %.4f", "");  
    gotopos(r+3,33);  
    cprintf("Quantity in Stock: %.4f", inv_stock.itemqty);  
    gotopos(r+4,33);  
    cprintf("Reorder Level: %d", "");  
    gotopos(r+4,33);  
    cprintf("Reorder Level: %d", inv_stock.minqty);  
    gotopos(r+5,33);  
    cprintf("\nPress Any Key...");fflush(stdin);getch();  
    textbackground(BROWN);  
    textcolor(BLUE);  
    return;  
}  
  
/*This function will return 0 if an item cannot issued, else issues the item*/  
IssueItem()
```

```

{
    float issueqnty;
    DisplayItemInfo();
    print2screen(15,33,"Enter Quantity: ",BROWN,BLUE,0);fflush(stdin);gotopos(15,49);
    scanf("%f",&issueqnty);

    /*If the stock of the item is greater than minimum stock*/
    if((inv_stock.itemqty - issueqnty) >= inv_stock.minqty)
    {
        textColor(BLUE);
        textbackground(BROWN);
        gotopos(18,33);
        cprintf("%.4f Item(s) issued.",issueqnty);
        gotopos(19,33);
        cprintf("You should pay RS. %.2f",issueqnty*inv_stock.itemrate);getch();
        textColor(BLUE);
        inv_stock.itemqty-=issueqnty;           /*Updating quantity for the item in stock*/
        fseek(dbfp,-stocksize,SEEK_CUR);
        fwrite(&inv_stock,stocksize,1,dbfp);
        return issueqnty;
    }
    /* If the stock of the item is less than minimum stock.ie Reorder level*/
    else
    {
        messagebox(10,33,"Insufficient quantity in stock.","Insufficient Stock",
                   ' ',mboxbrdrclr,mboxbgclr,mboxfgclr,0);
        gotopos(17,33);
        textColor(BLUE);
        textbackground(BROWN);
        cprintf("ONLY %.4f pieces of the Item can be issued.",inv_stock.itemqty-inv_stock.minqty);
        gotopos(18,33);
        cprintf("Press Any Key...");getch();
        textColor(BLUE);
        textbackground(BROWN);
        return 0;
    }
}

/* Calculates the total investment amount for the stock available*/
float getInvestmentInfo(void)
{
    tot_investment=0;
}

```

```
rewind(dbfp);
while(fread(&inv_stock,stocksize,1,dbfp)==1)
    tot_investment+=(inv_stock.itemrate*inv_stock.itemqty);
return tot_investment;
}

/* Creates a backup file "Bakckup" of "inv_stock.dat"*/
BackupDatabase(void)
{
FILE *fback;
fback=fopen("d:/Backup.dat","w");
rewind(dbfp);
while(fread(&inv_stock,stocksize,1,dbfp)==1)
    fwrite(&inv_stock,stocksize,1,fback);
fclose(fback);
return;
}

/*This structure is used color settings for the application*/
struct colors
{
char cfg_name[10];

int mboxbrdrclr;
int mboxbgclr;
int mboxfgclr;

int menutxtbgclr;
int menutxtfgclr;
int appframeclr;

int section1_symb;
int section1_bgclr;
int section1_fgclr;

int section2_symb;
int section2_bgclr;
int section2_fgclr;
}clr;
const long int clrsize=sizeof(clr);

/* Gets the display configuration for the application*/
getConfiguration()
```

```

{
FILE *flast;
flast=fopen("lastcfg","r+");
if(flast==NULL)
{
    SetDefaultColor();
    return 0;
}
rewind(flast);

/*Reads the first record.*/
fread(&clr,clrsiz,1,flast);
#endif OKAY
if(strcmp(clr.cfg_name,"lastclr")!=0)
{
    SetDefaultColor();
    fclose(flast);
    return 0;
}
#endif
mboxbrdrclr=clr.mboxbrdrclr;mboxbgclr=clr.mboxbgclr;mboxfgclr=clr.mboxfgclr;
menutxtbgclr=clr.menutxtbgclr;menutxtfgclr=clr.menutxtfgclr;appframeclr=clr.appframeclr;
section1_symb=clr.section1_symb;section1_bgclr=clr.section1_bgclr;section1_fgclr=clr.section1_fgclr;
section2_symb=clr.section2_symb;section2_bgclr=clr.section2_bgclr;section2_fgclr=clr.section2_fgclr;
fclose(flast);
return 1;
}

/* Sets the default color settings for the application*/
SetDefaultColor()
{
mboxbrdrclr=BLUE,mboxbgclr=GREEN,mboxfgclr=WHITE;
menutxtbgclr=BROWN,menutxtfgclr=BLUE,appframeclr=CYAN;
section1_symb=' ',section1_bgclr=BROWN,section1_fgclr=BLUE;
section2_symb=' ',section2_bgclr=BROWN,section2_fgclr=BLUE;
return 1;
}

/* Adds animation to a text */
BlinkText(const int r,const int c,char txt[],int bgclr,int fgclr,int BGCLR2,int FGCLR2,int
blink,const int dly)

```

```
{  
    int len=strlen(txt);  
  
    BGCLR2=bgclr;FGCLR2=BLUE;  
    htskin(r,c,' ',len,bgclr,bgclr,0);  
    print2screen(r,c,txt,bgclr,fgclr,blink);  
  
    write2screen(r,c+animcounter+1,txt[animcounter],BGCLR2,FGCLR2,0);  
    write2screen(r,c+animcounter+2,txt[animcounter+1],BGCLR2,FGCLR2,0);  
    write2screen(r,c+animcounter+3,txt[animcounter+2],BGCLR2,FGCLR2,0);  
    write2screen(r,c+animcounter+4,txt[animcounter+3],BGCLR2,FGCLR2,0);  
    write2screen(r,c+animcounter+5,txt[animcounter+4],BGCLR2,FGCLR2,0);  
    write2screen(r,c+animcounter+6,txt[animcounter+5],BGCLR2,FGCLR2,0);  
    delay(dly*2);  
    write2screen(r,c+animcounter+1,txt[animcounter],bgclr,fgclr,0);  
    write2screen(r,c+animcounter+2,txt[animcounter+1],bgclr,fgclr,0);  
    write2screen(r,c+animcounter+3,txt[animcounter+2],bgclr,fgclr,0);  
    write2screen(r,c+animcounter+4,txt[animcounter+3],bgclr,fgclr,0);  
    write2screen(r,c+animcounter+5,txt[animcounter+4],bgclr,fgclr,0);  
    write2screen(r,c+animcounter+6,txt[animcounter+5],bgclr,fgclr,0);  
  
    animcounter+=1;  
    if(animcounter+5 >= len) animcounter=0;  
  
    return;  
}  
  
/* Displays a single character with its attrribute*/  
write2screen(int row,int col,char ch,int bg_color,int fg_color,int blink)  
{  
    int attr;  
    char far *v;  
    char far *ptr=(char far*)0xB8000000;  
    if(blink!=0)  
        blink=128;  
  
    attr=bg_color+blink;  
  
    attr=attr<<4;  
    attr+=fg_color;  
    attr=attr|blink;
```

```

v=ptr+row*160+col*2; /*Calculates the video memory address corresponding to row & column*/
*v=ch;
v++;
*v=attr;
return 0;
}

/* Prints text with color attribute direct to the screen*/
print2screen(int row,int col,char string[],int bg_color,int fg_color,int blink)
{
    int i=row,j=col,strno=0,len;
    len=strlen(string);
    while(j<80)
    {
        j++;
        if(j==79)
        {
            j=0;
            i+=1;
        }
        write2screen(i,j,string[strno],bg_color,fg_color,blink); /*See below function*/
        strno+=1;
        if(strno > len-1)
        break;
    }
    return;
}

/*Prints text horizontally*/
htskin(int row,int column,char symb,int no,int bg_color,int fg_color,int blink)
{
    int i;
    for(i=0;i<no;i++)
        write2screen(row,column++,symb,bg_color,fg_color,blink); /*Print one symbol*/
    return;
}

/*Print text vertically*/
vtskin(int row,int column,char symb,int no,int bg_color,int fg_color,int blink)
{
    int i;
    for(i=0;i<no;i++)

```

```

write2screen(row++,column,symb,bg_color,fg_color,blink); /*Print one symbol*/
return;
}

/* Shows a message box*/
messagebox(int row,int column,char message[50],char heading[10],char symb,int borderclr,int bg_
color,int fg_color,int blink)
{
int len;
char key,image[1000];
len=strlen(message);
capture_image(row,column,row+3,column+len+7,&image);
draw_mbox(row,column,row+3,column+len+7,symb,symb,borderclr,YELLOW,blink,borderclr,YELLOW,blink);
fillcolor(row+1,column+1,row+2,column+len+6,' ',bg_color,bg_color,0);
print2screen(row+1,column+2,message,bg_color,fg_color,blink);
print2screen(row+2,column+2,"Press Any Key...",bg_color,fg_color,blink);
print2screen(row,column+1,heading,borderclr,fg_color,blink);
sound(400);
delay(200);
nosound();
fflush(stdin);
key=getch();
put_image(row,column,row+3,column+len+7,&image);
return key;
}

/* Fills color in a region*/
fillcolor(int top_row,int left_column,int bottom_row,int right_column,char symb,int bg_color,int
fg_color,int blink)
{
int i,j;
for(i=top_row;i<=bottom_row;i++)
htskin(i,left_column,symb,right_column-left_column+1,bg_color,fg_color,blink);
return;
}

/* Prints a message box with an appropriate message*/
draw_mbox(int trow,int tcolumn,int brow,int bcolumn,char hsymb,char vsymb,int hbg_color,int hfg_
color,int hblink,int vbg_color,int vfg_color,int vblink)
{
htskin(trow,tcolumn,hsymb,bcolumn-tcolumn,hbg_color,hfg_color,hblink);
htskin(brow,tcolumn,hsymb,bcolumn-tcolumn,hbg_color,hfg_color,hblink);
}

```

```

vtskin(trow,tcolumn,vsymb,brow-trow+1,vbg_color,vfg_color,vblink);
vtskin(trow,bcolumn,vsymb,brow-trow+1,vbg_color,vfg_color,vblink);
return;
}

/* Copies the txt mode image below the messagebox*/
capture_image(int toprow,int leftcolumn,int bottomrow,int rightcolumn,int *image)
{
    char far *vidmem;
    int i,j,count;
    count=0;
    for(i=toprow;i<=bottomrow;i++)
        for(j=leftcolumn;j<=rightcolumn;j++)
    {
        vidmem=(char far*)0xB8000000+(i*160)+(j*2); /*Calculates the video memory address
corresponding to row & column*/
        image[count]=*vidmem;
        image[count+1]=*(vidmem+1);
        count+=2;
    }
    return;
}

/* Places an image on the screen*/
put_image(int toprow,int leftcolumn,int bottomrow,int rightcolumn,int image[])
{
    char far *ptr=(char far*)0xB8000000;
    char far *vid;
    int i,j,count;
    count=0;
    for(i=toprow;i<=bottomrow;i++)
        for(j=leftcolumn;j<=rightcolumn;j++)
    {
        vid=ptr+(i*160)+(j*2); /*Calculates the video memory address corresponding to row &
column*/
        *vid=image[count];
        *(vid+1)=image[count+1];
        count+=2;
    }
    return;
}

```

```
/* To move the cursor position to desired position*/
gotopos(int r,int c)
{
    union REGS i,o;
    i.h.ah=2;
    i.h.bh=0;
    i.h.dh=r;
    i.h.dl=c;
    int86(16,&i,&o);
    return 0;
}
```

```
union REGS i,o;
```

```
/* Initialize the mouse*/
```

```
initmouse()
```

```
{
    i.x.ax=0;
    int86(0x33,&i,&o);
    return(o.x.ax);
}
```

```
/* Shows the mouse pointer*/
```

```
showmouseptr()
{
    i.x.ax=1;
    int86(0x33,&i,&o);
    return;
}
```

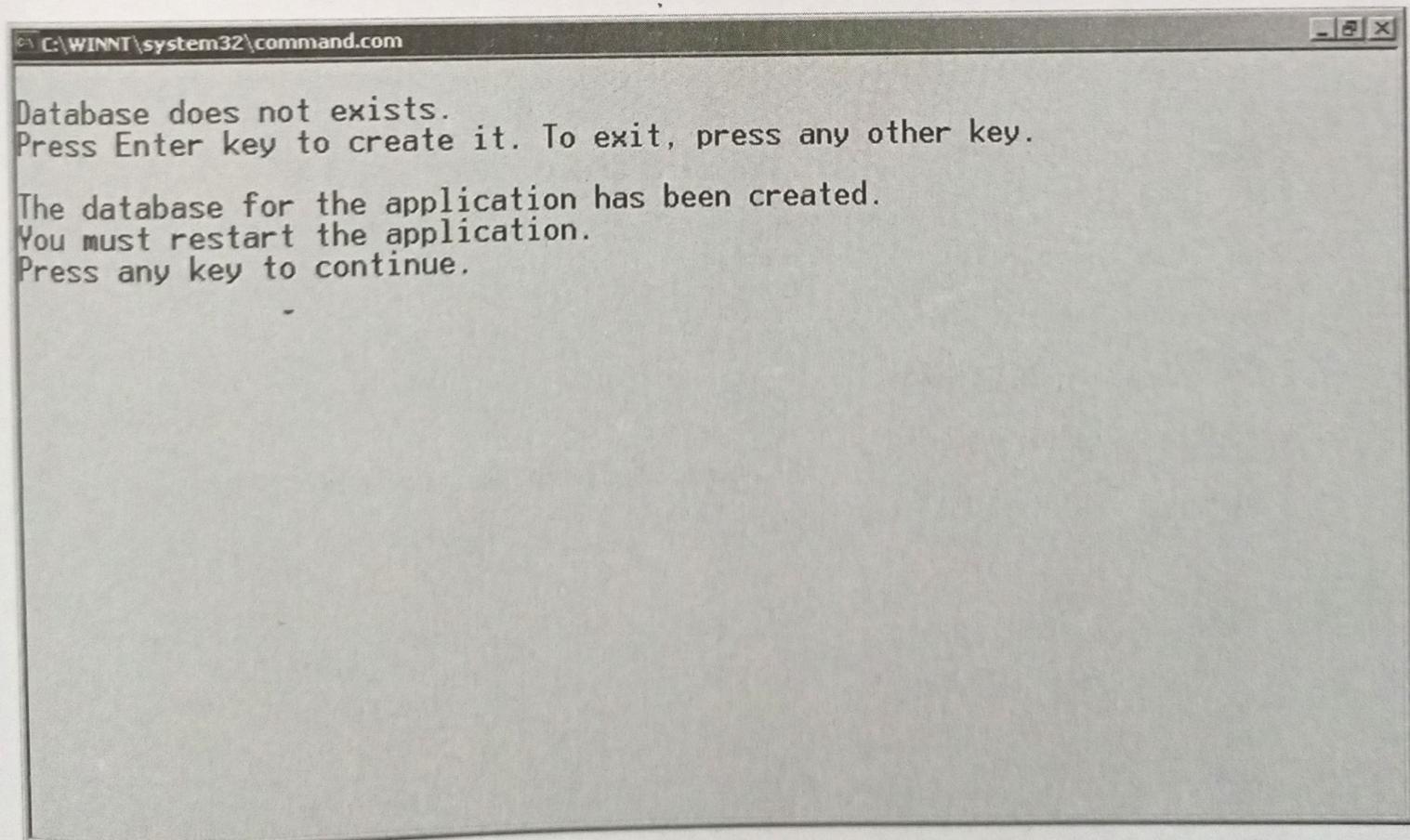
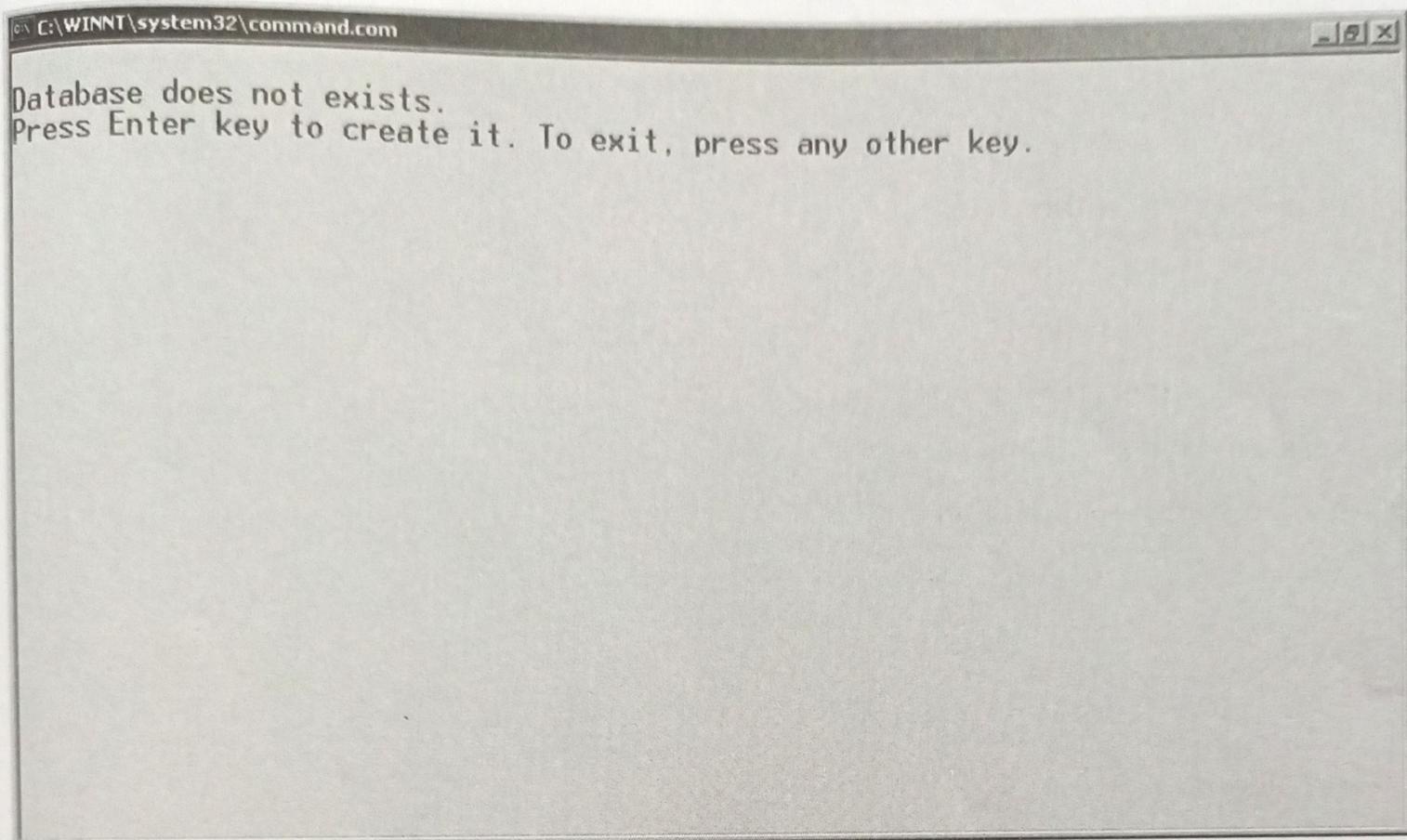
```
/* Get the mouse position*/
```

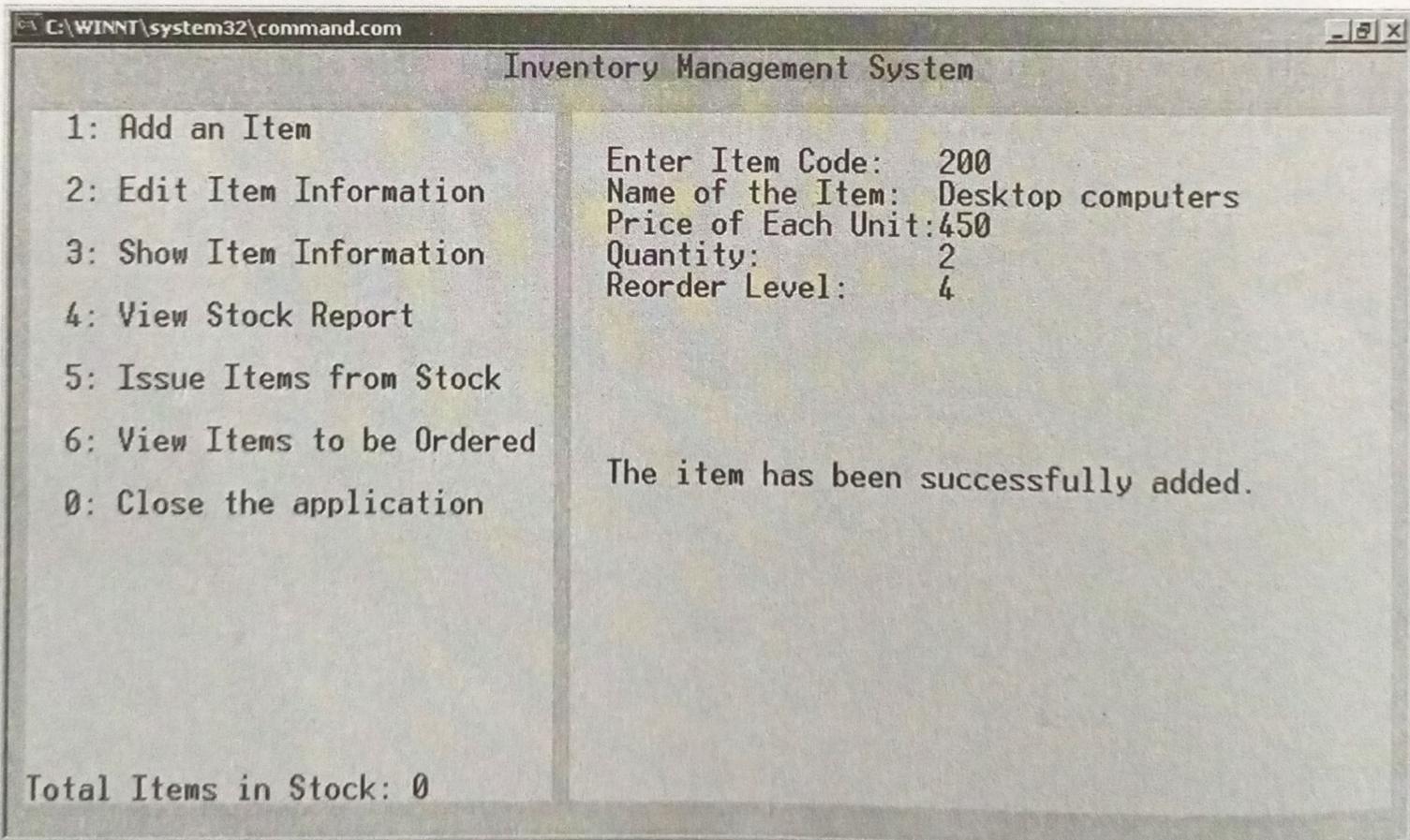
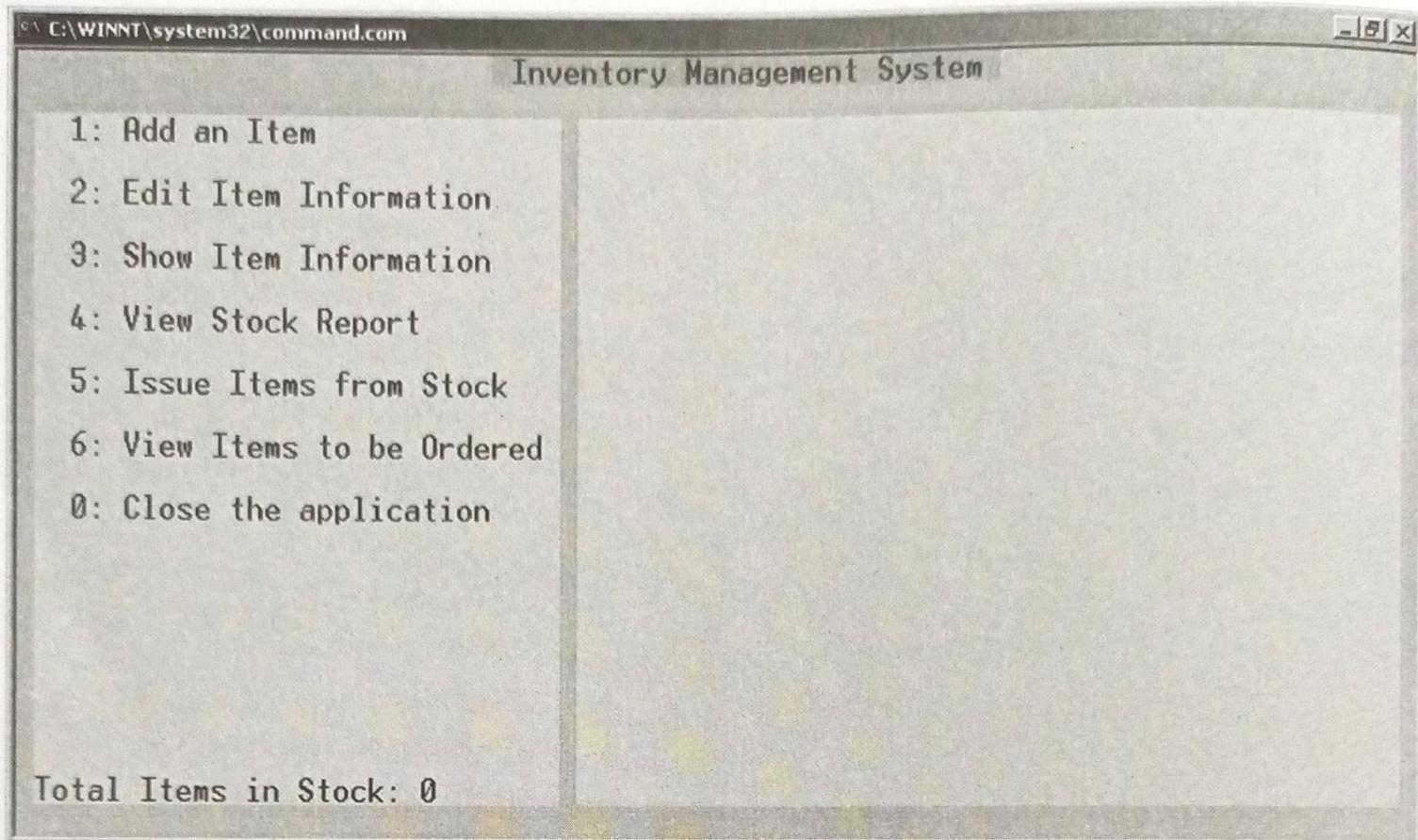
```
getmousepos(int *button,int *x,int *y)
{
    i.x.ax=3;
    int86(0x33,&i,&o);
    *button=o.x.bx;
    *x=o.x.dx;
    *y=o.x.cx;
    return 0;
}
```

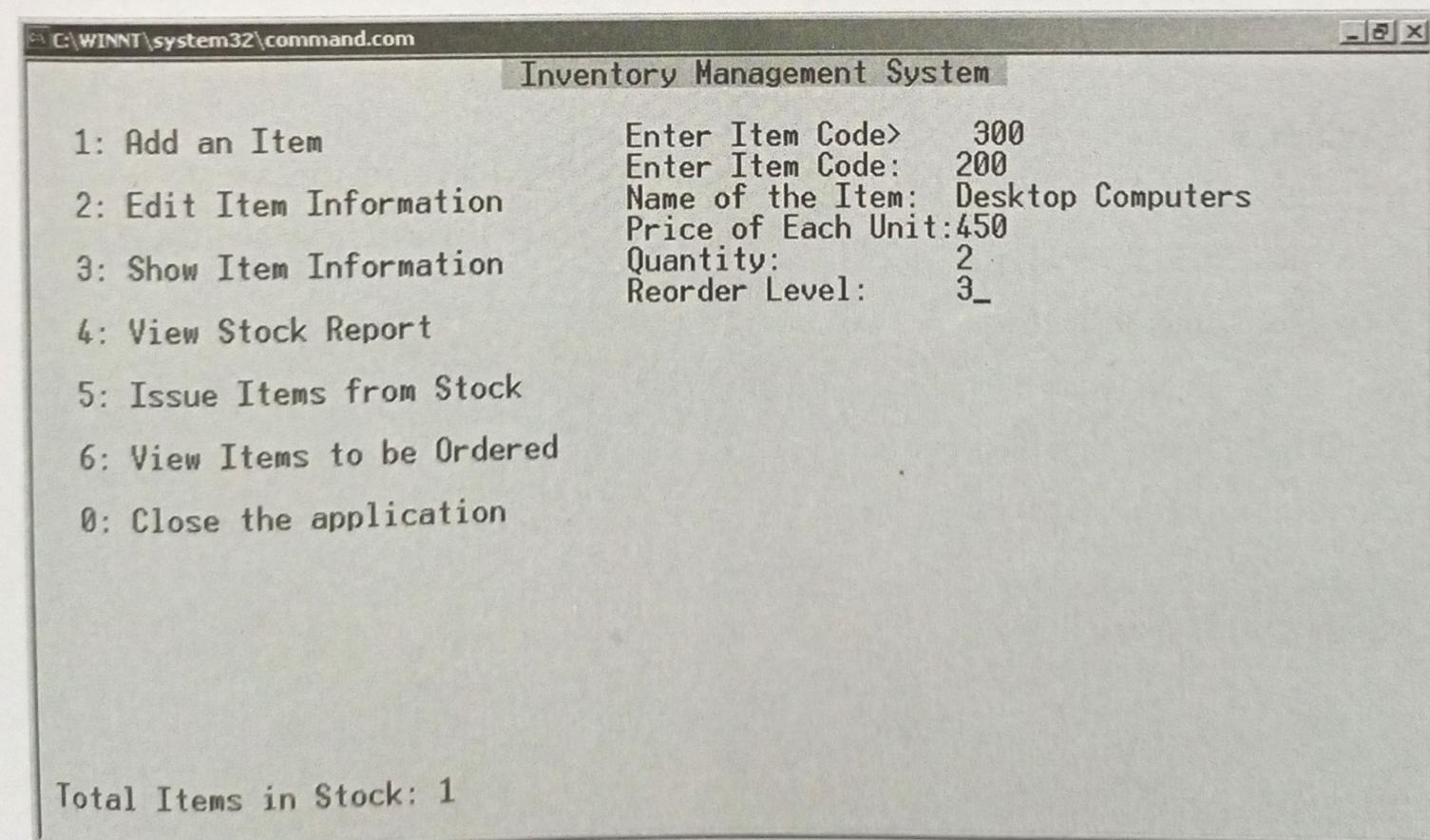
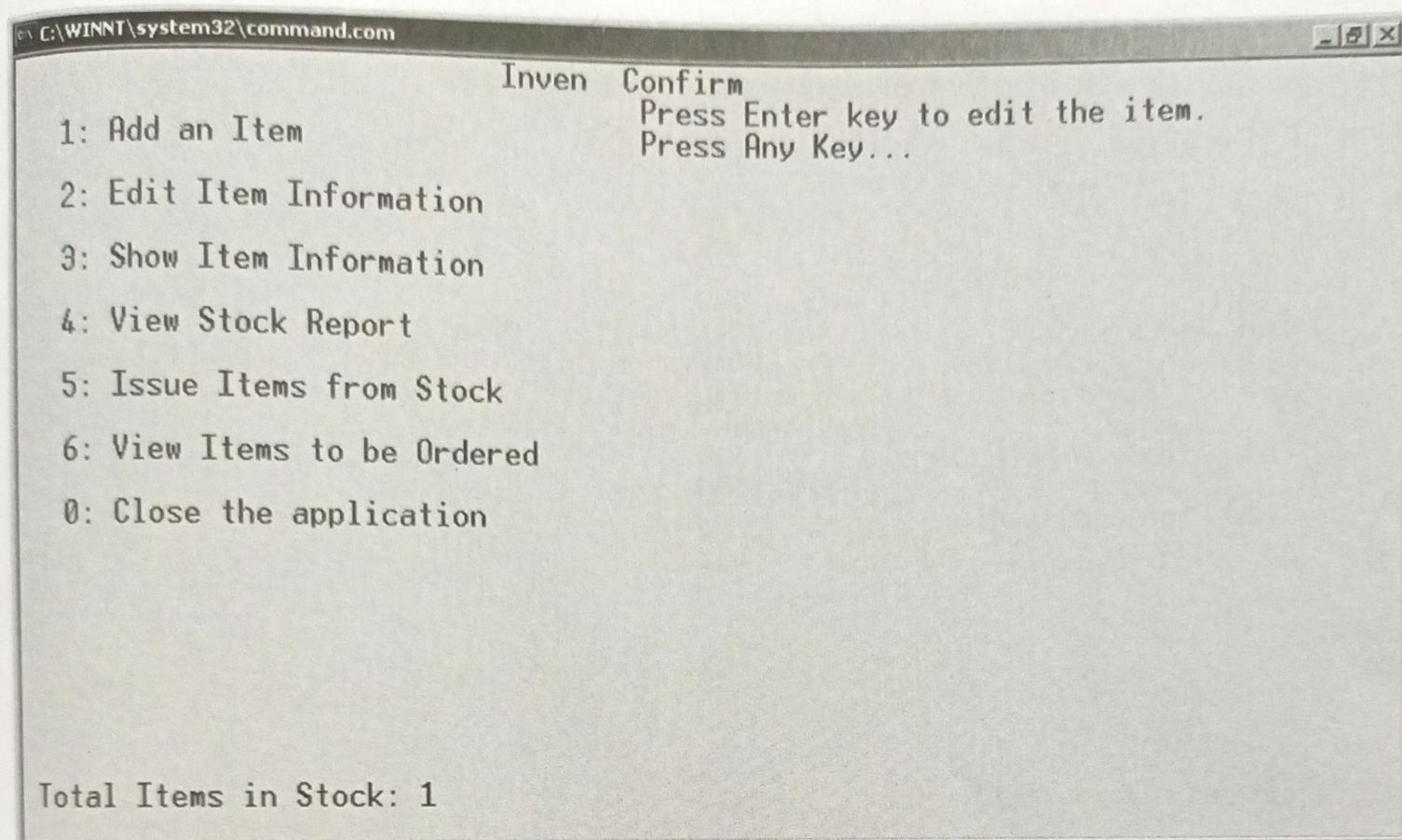
```
/* Restores the default text mode*/
setdefaultmode()
{
    set25x80();
    setdefaultcolor();
    return;
}

/* Sets the default color and cursor of screen*/
setdefaultcolor()
{
    int i;
    char far *vidmem=(char far*)0xB8000000;
    window(1,1,80,25);
    clrscr();
    for (i=1;i<4000;i+=2)
        *(vidmem+i)=7;
    _setcursortype(_NORMALCURSOR);
    return;
}

/* Sets 25x80 Text mode*/
set25x80()
{
    asm mov ax,0x0003;
    asm int 0x10;
    return;
}
```







```
C:\WINNT\system32\command.com
Inventory Management System

1: Add an Item
2: Edit Item Information
3: Show Item Information
4: View Stock Report
5: Issue Items from Stock
6: View Items to be Ordered
0: Close the application

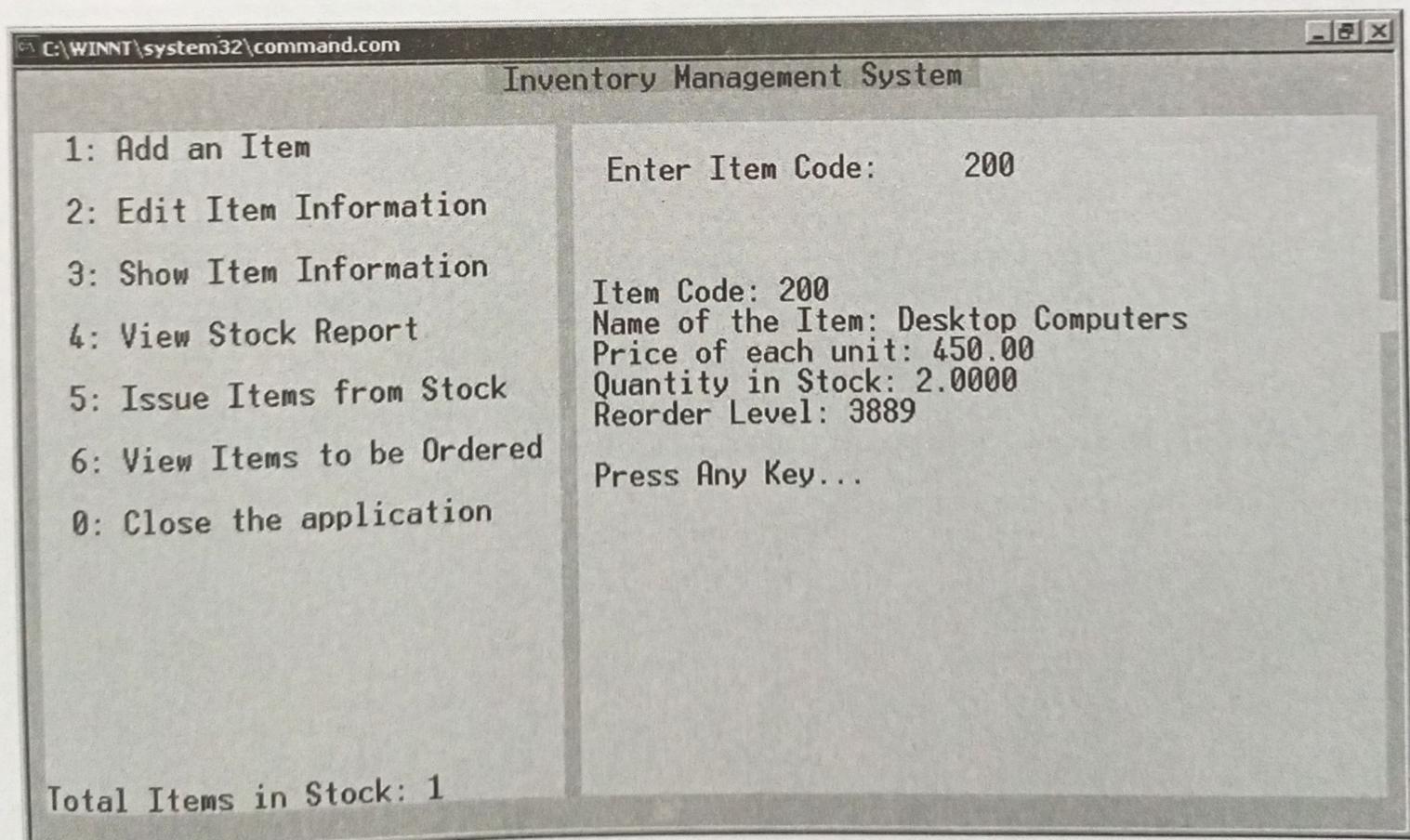
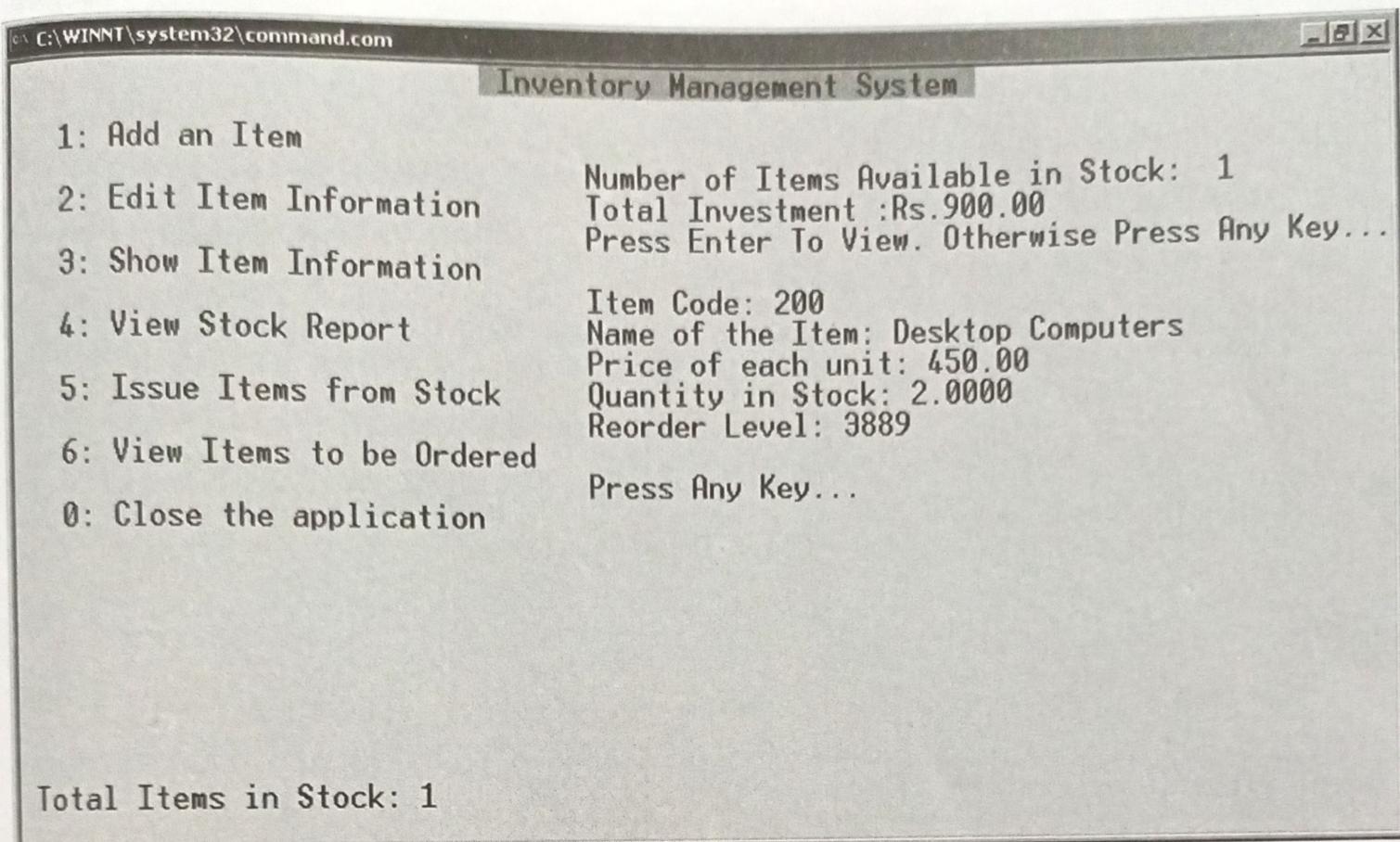
Enter Item Code: 200
Item Code: 200
Name of the Item: Desktop Computers
Price of each unit: 450.00
Quantity in Stock: 2.0000
Reorder Level: 3889

Press Any Key...
Total Items in Stock: 1
```

```
C:\WINNT\system32\command.com
Inventory Management System

1: Add an Item
2: Edit Item Information
3: Show Item Information
4: View Stock Report
5: Issue Items from Stock
6: View Items to be Ordered
0: Close the application

Number of Items Available in Stock: 1
Total Investment :Rs.900.00
Press Enter To View. Otherwise Press Any Key...
Total Items in Stock: 1
```



```
C:\WINNT\system32\command.com Inventory Management System

1: Add an Item
2: Edit Item Information
3: Show Item Information
4: View Stock Report
5: Issue Items from Stock
6: View Items to be Ordered
0: Close the application

Enter Item Code: 200

Item Code: 200
Name of the Item: Desktop Computers
Price of each unit: 450.00
Quantity in Stock: 2.0000
Reorder Level: 3889

Press Any Key...

Enter Quantity:1

Total Items in Stock: 1
```

```
C:\WINNT\system32\command.com Inventory Management System

1: Add an Item
2: Edit Item Information
3: Show Item Information
4: View Stock Report
5: Issue Items from Stock
6: View Items to be Ordered
0: Close the application

Enter Item Code: 200

Item Code: 200
Name of the Item: Desktop Computers
Price of each unit: 450.00
Insufficient Stock
Insufficient quantity in stock.
Press Any Key...

Enter Quantity:1

Total Items in Stock: 1
```

```
C:\WINNT\system32\command.com
Inventory Management System

1: Add an Item
2: Edit Item Information
3: Show Item Information
4: View Stock Report
5: Issue Items from Stock
6: View Items to be Ordered
0: Close the application

Enter Item Code: 200

Item Code: 200
Name of the Item: Desktop Computers
Price of each unit: 450.00
Quantity in Stock: 6.0000
Reorder Level: 2889

Press Any Key...

Total Items in Stock: 1
```

```
C:\WINNT\system32\command.com
Inventory Management System

1: Add an Item
2: Edit Item Information
3: Show Item Information
4: View Stock Report
5: Issue Items from Stock
6: View Items to be Ordered
0: Close the application

Enter Item Code: 200

Item Code: 200
Name of the Item: Desktop Computers
Price of each unit: 450.00
Quantity in Stock: 6.0000
Reorder Level: 2889

Press Any Key...

Enter Quantity:2

Total Items in Stock: 1
```

```
C:\WINNT\system32\command.com
Inventory Management System

1: Add an Item
2: Edit Item Information
3: Show Item Information
4: View Stock Report
5: Issue Items from Stock
6: View Items to be Ordered
0: Close the application

Enter Item Code: 200

Item Code: 200
Name of the Item: Desktop Computers
Price of each unit: 450.00
Quantity in Stock: 6.0000
Reorder Level: 2889

Press Any Key...

Enter Quantity:2

2.0000 Item(s) issued.
You should pay RS. 900.00

Total Items in Stock: 1
```

```
C:\WINNT\system32\command.com
Inventory Management System

1: Add an Item
2: Edit Item Information
3: Show Item Information
4: View Stock Report
5: Issue Items from Stock
6: View Items to be Ordered
0: Close the application

Enter Item Code: 200

Item Code: 200
Name of the Item: Desktop Computers
Price of each unit: 450.00
Quantity in Stock: 6.0000
Reorder Level: 2889

Press Any Key...

Enter Quantity:2

2.0000 Item(s) issued.
Would you like to issue another item(Y/N)?
```

Total Items in Stock: 1

