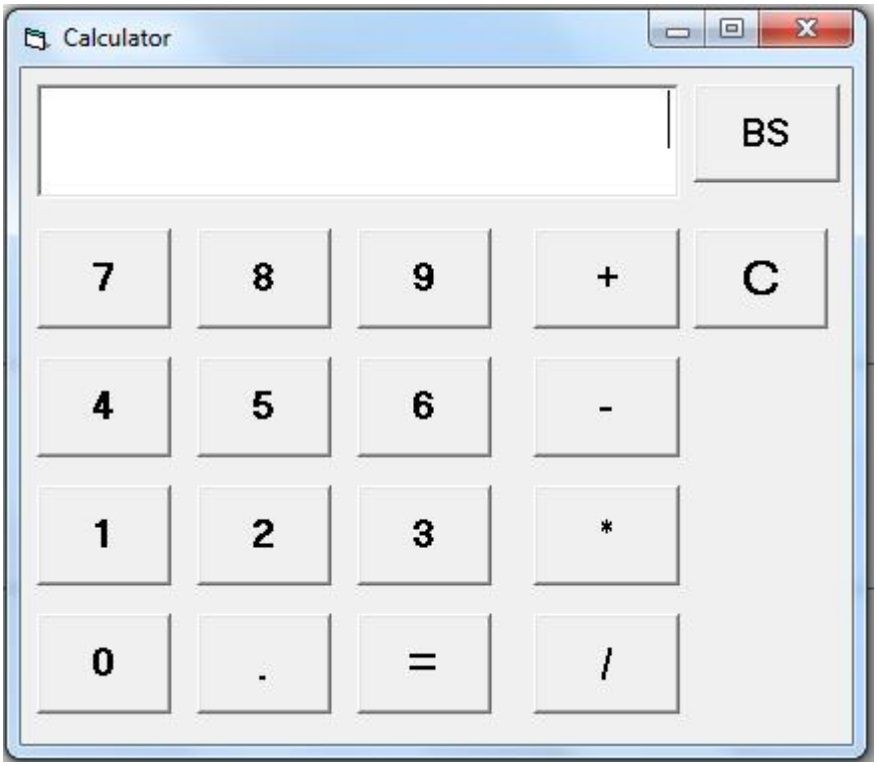
/*** 1. CALCULATOR ***/

Form Design

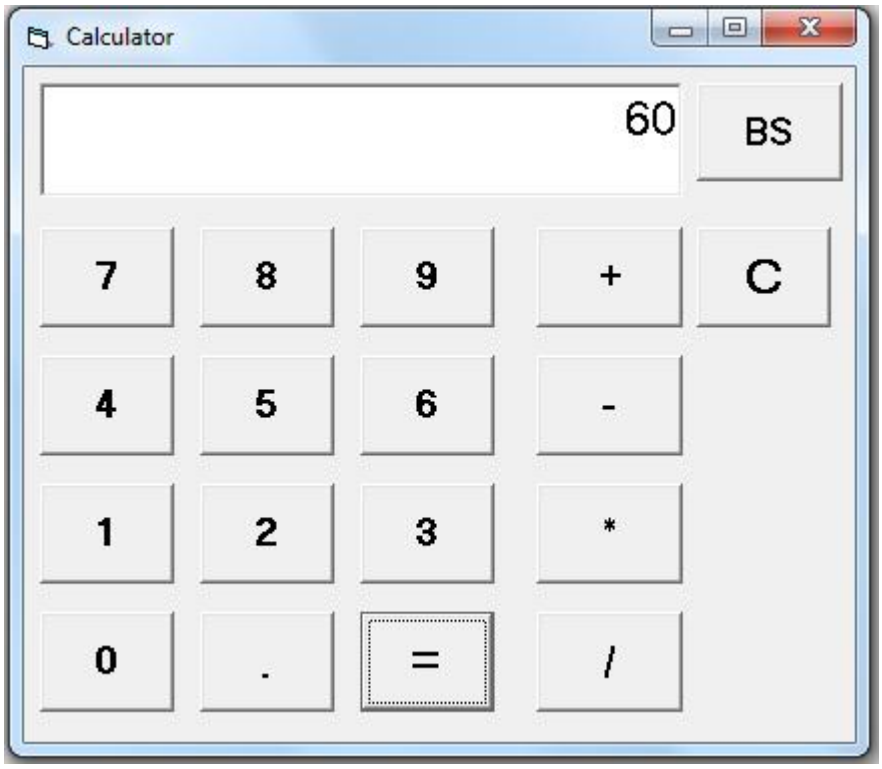


Coding

```
Dim SIGNE As String
Dim old As Double
Private Sub cmd_bs_Click()
If Len(Text1.Text) > 0 Then
Text1.Text = Left(Text1.Text, Len(Text1.Text) - 1)
End If
End Sub
Private Sub cmd_c_Click()
Call Form_Load
End Sub
```

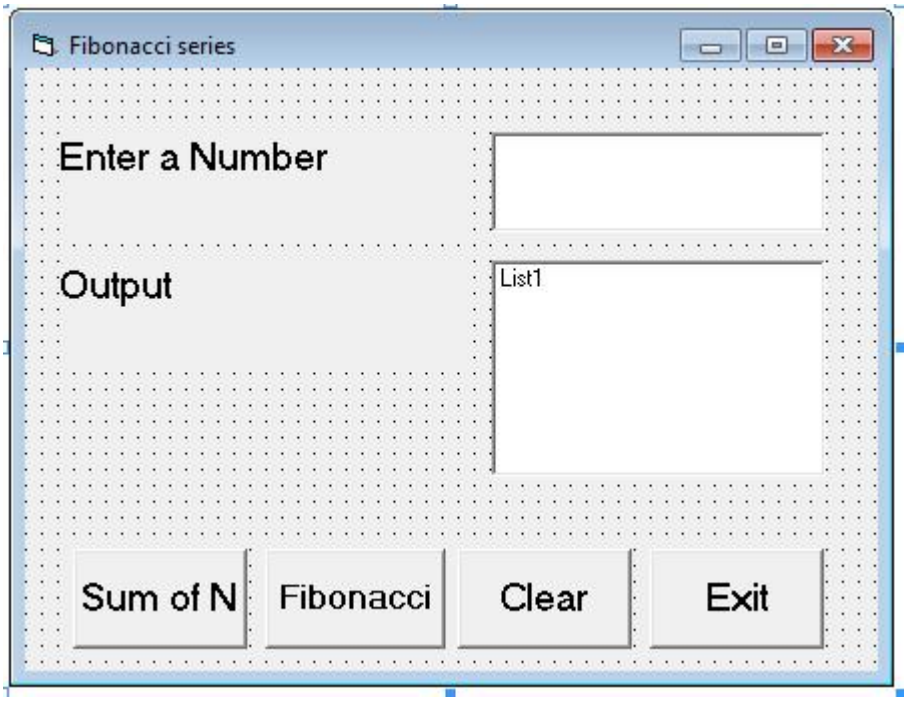
```
Private Sub cmd_equ_Click()
Select Case (SIGNE)
Case Is = "+"
Text1.Text = old + Val(Text1.Text)
Case Is = "-"
Text1.Text = old - Val(Text1.Text)
Case Is = "*"
Text1.Text = old * Val(Text1.Text)
Case Is = "/"
Text1.Text = old / Val(Text1.Text)
End Select
End Sub

Private Sub cmd_no_Click(Index As Integer)
Text1.Text = Text1.Text + cmd_no(Index).Caption
If cmd_no(Index).Caption = "." Then cmd_no(10).Enabled = False
End if
End Sub

Private Sub cmd_op_Click(Index As Integer)
old = Val(Text1.Text)
Text1.Text = Val(" ")
SIGNE = cmd_op(Index).Caption
cmd_no(10).Enabled = True
End Sub

Private Sub Form_Load()
Text1.Text = Val(" ")
old = 0
cmd_no(10).Enabled = True
End Sub
```

/* OUTPUT-CALCULATOR ***/**



/*** 2. FIBONACCI SERIES & SUM OF N NUMBER ***/

Form Design

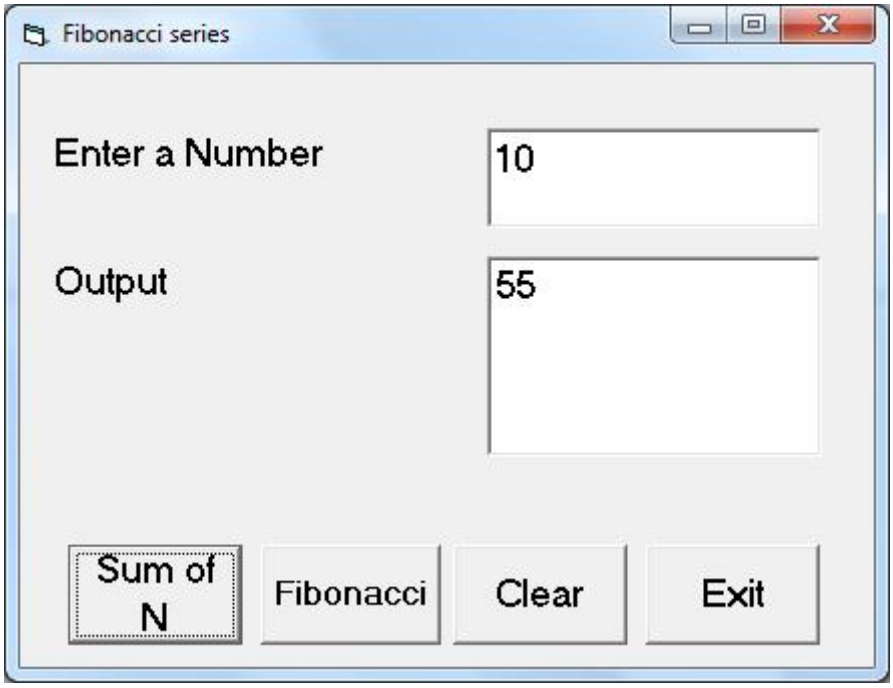


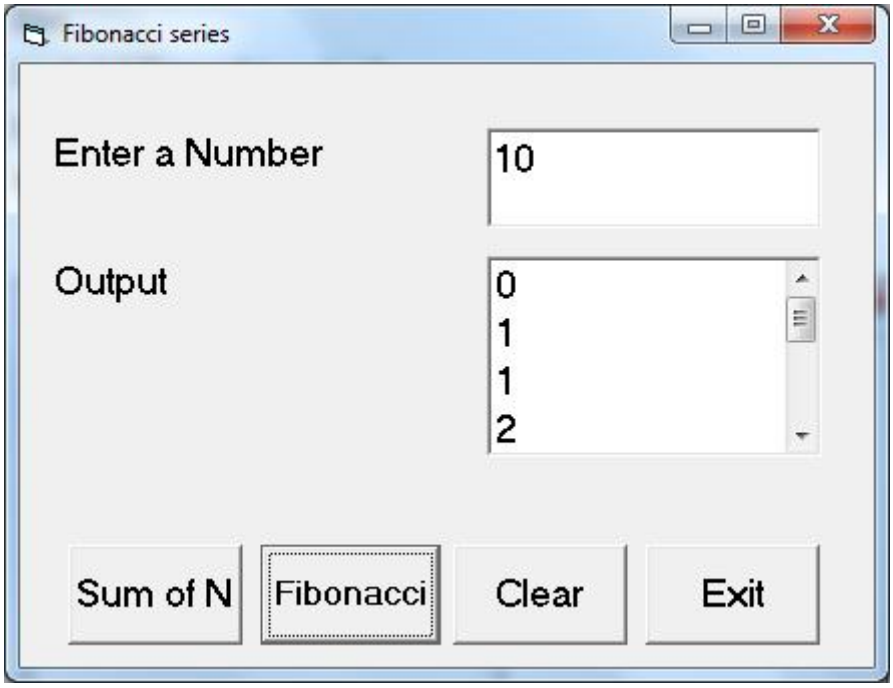
Coding

```
Dim n, i As Integer
Dim f1, f2, f3 As Single
Private Sub cmdclear_Click()
Text1 = ""
List1.Clear
End Sub

Private Sub cmdexit_Click()
End
End Sub

Private Sub cmdfab_Click()
n = Val(Text1)
f1 = -1
f2 = 1
For i = 1 To n
f3 = f1 + f2
List1.AddItem f3
f1 = f2
```

```
f2 = f3
Next i
End Sub

Private Sub cmdsum_Click()
n = Val(Text1)
If n = 1 Then
List1.AddItem 1
Else
List1.AddItem n * (n + 1) / 2
End If
End Sub
```

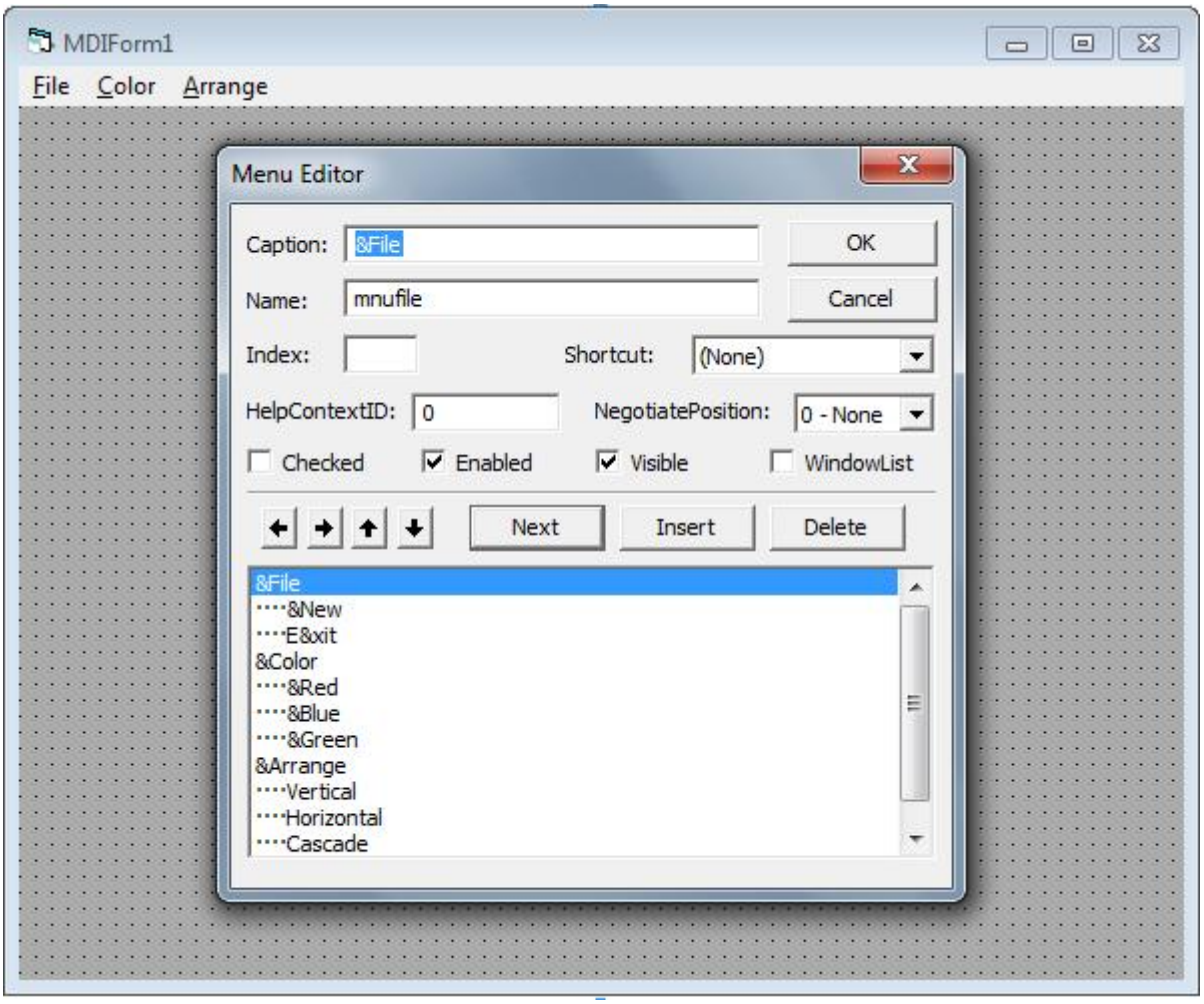
/* OUTPUT- FIBONACCI SERIES & SUM OF N NUMBER ***/**





/*** 3. MENU DRIVEN USING MDI ***/

Form Design



Coding

Dim frmd As Form1

Private Sub mnublue_Click()

Me.ActiveForm.BackColor = vbBlue

End Sub

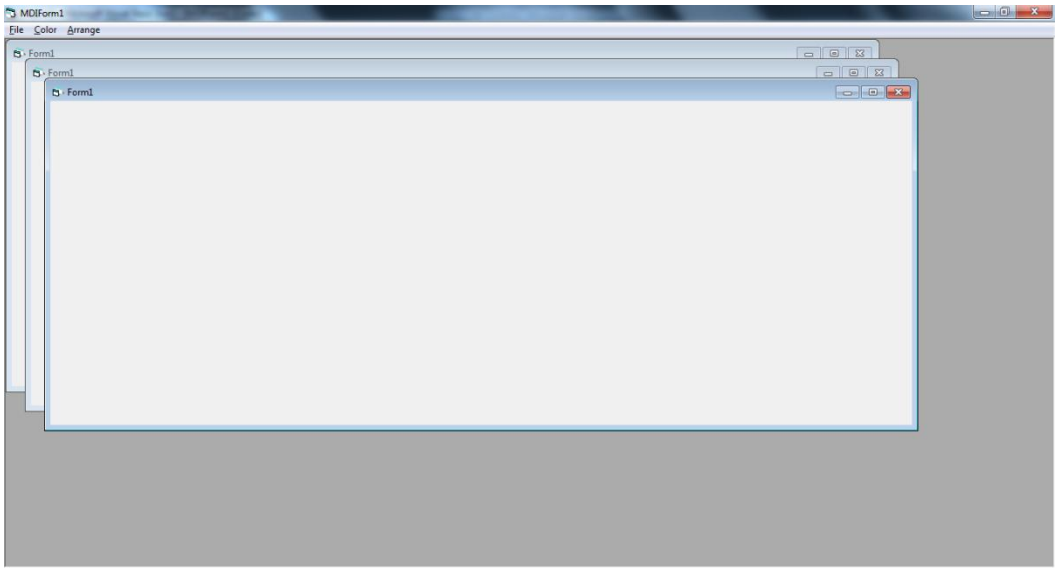
Private Sub mnucascade_Click()

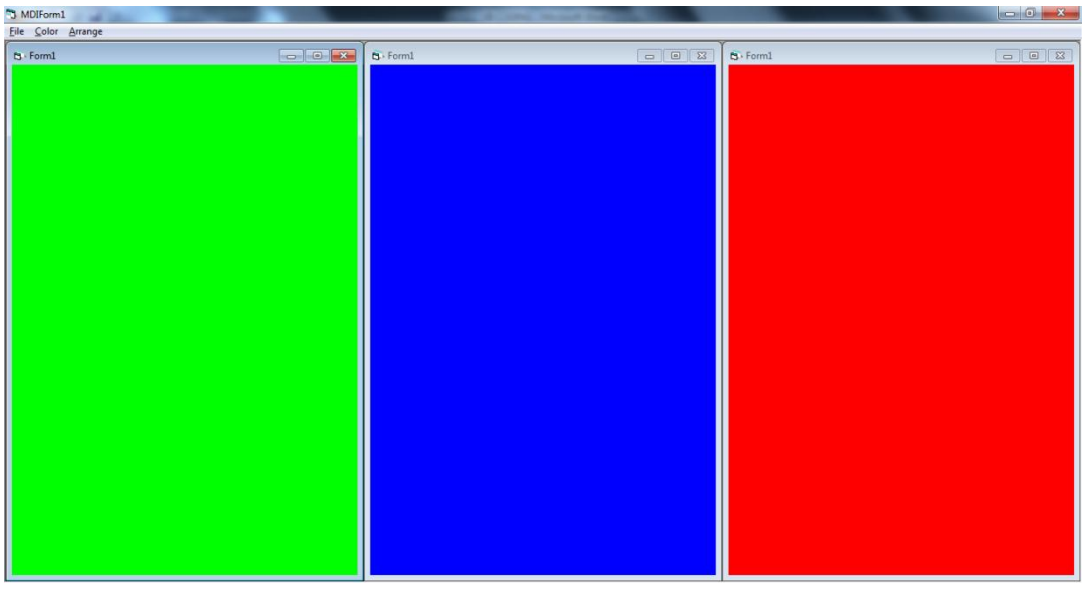
Me.Arrange vbCascade

End Sub

```
Private Sub mnugreen_Click()
Me.ActiveForm.BackColor = vbGreen
End Sub
Private Sub mnuhorizontal_Click()
Me.Arrange vbHorizontal
End Sub
Private Sub MDIForm_Load()
mnucolor.Enabled = True
End Sub
Private Sub mnunew_Click()
Set frmd = New Form1
frmd.Show
End Sub
Private Sub mnured_Click()
Me.ActiveForm.BackColor = vbRed
End Sub
Private Sub mnuvertical_Click()
Me.Arrange vbVertical
End Sub
```

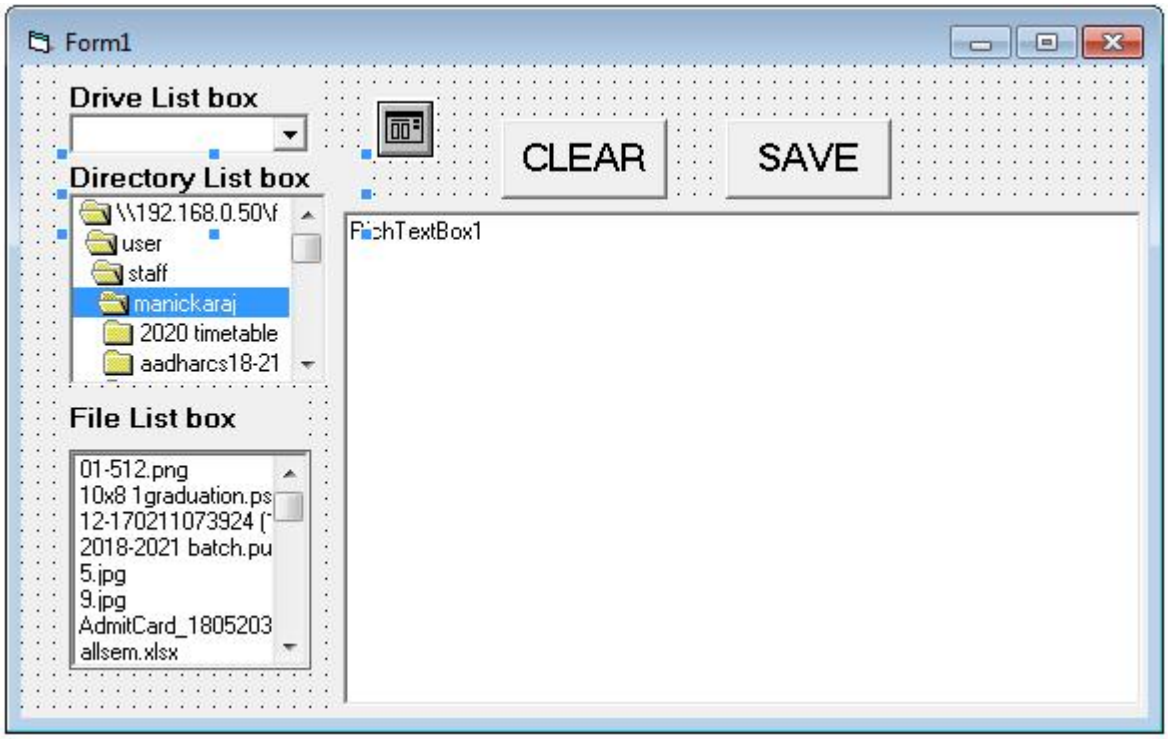
/*** OUTPUT - MENU DRIVEN USING MDI ***/





/*** 4. DRIVE, DIRECTORY AND FILE BOX ***/

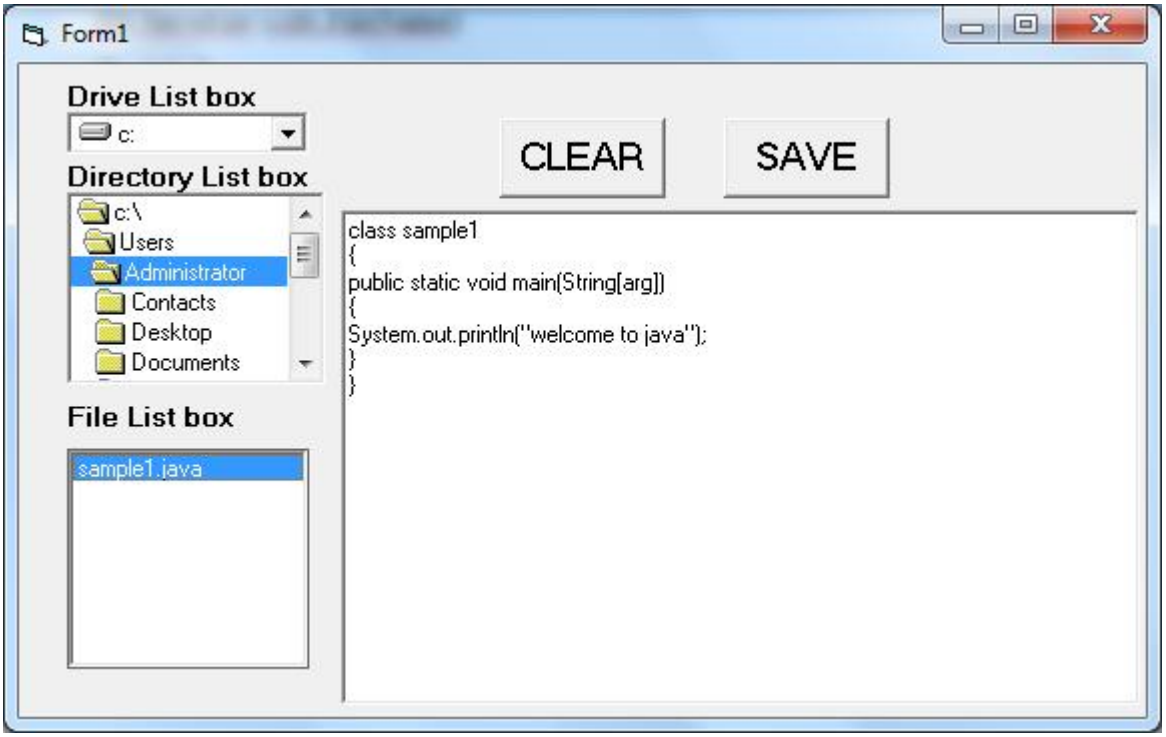
Form Design

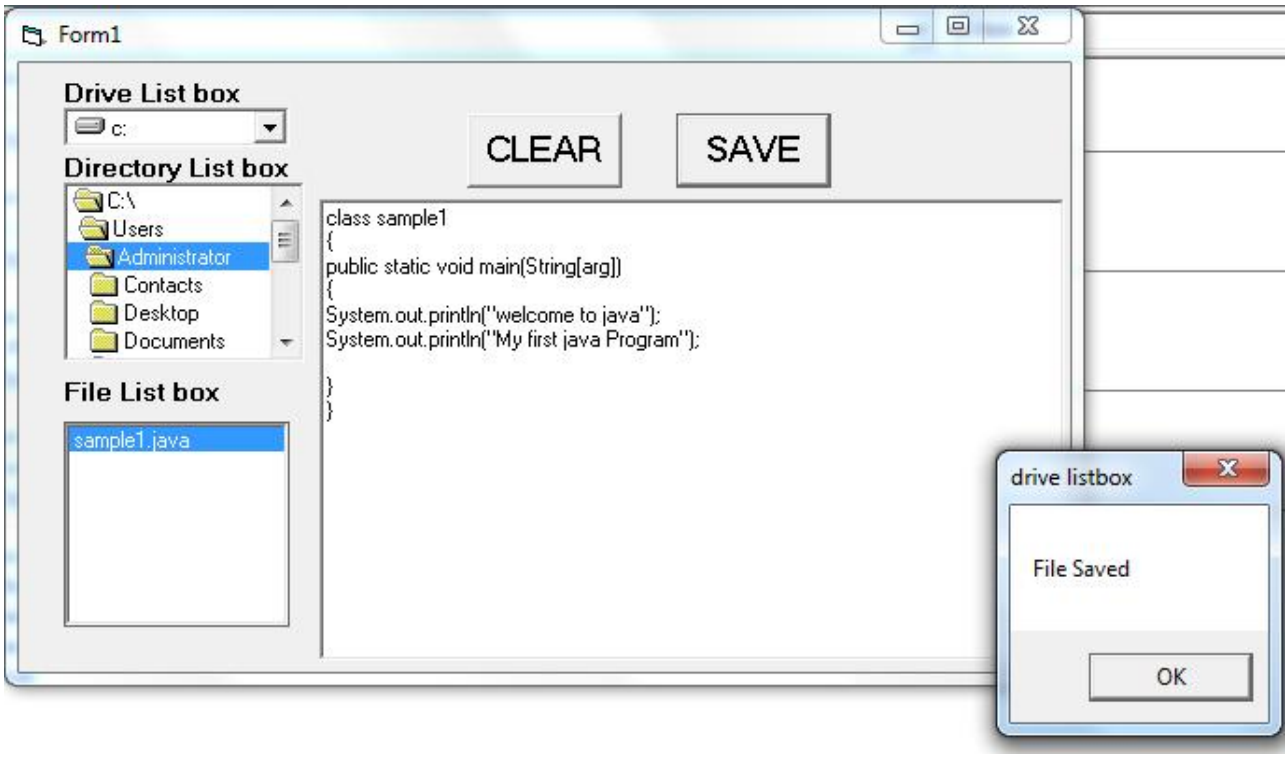


Coding

```
Private Sub clear_Click()
rtb = ""
End Sub
Private Sub Dir1_Change()
File1.Path = Dir1.Path
End Sub
Private Sub Drive1_Change()
Dir1.Path = Drive1.Drive
End Sub
Private Sub File1_Click()
rtb.LoadFile (File1.Path & "/" & File1.FileName)
End Sub
Private Sub save_Click()
cdb.ShowSave
rtb.SaveFile (cdb.FileName)
MsgBox "File Saved"
End Sub
```

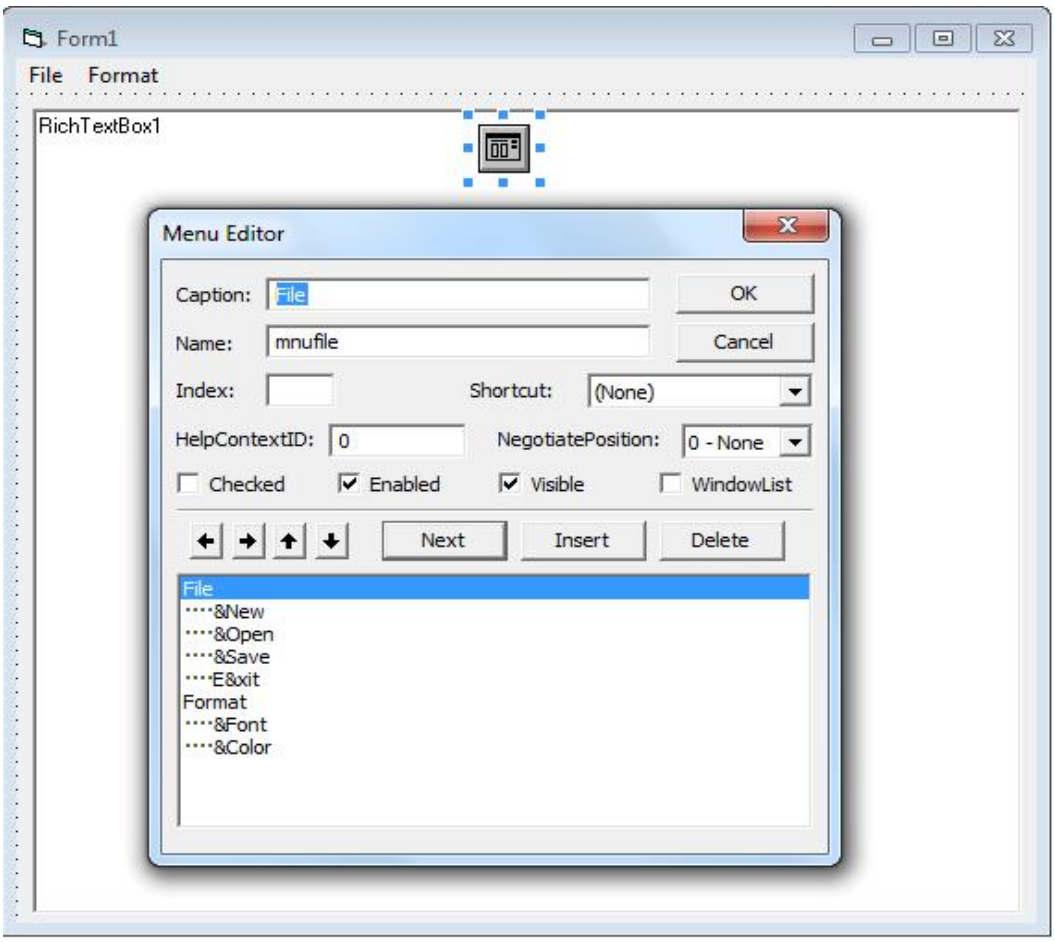
/*** OUTPUT - DRIVE, DIRECTORY AND FILE BOX ***/





/*** 5.TEXT FILES ***/

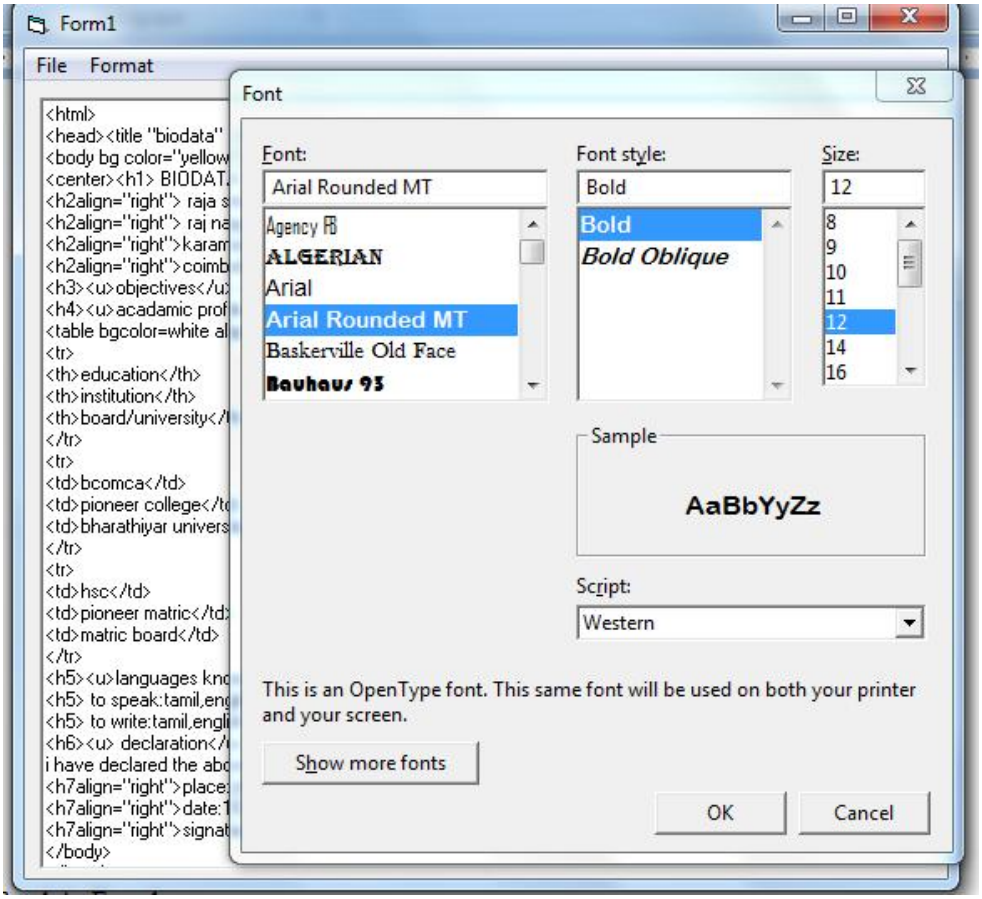
Form Design

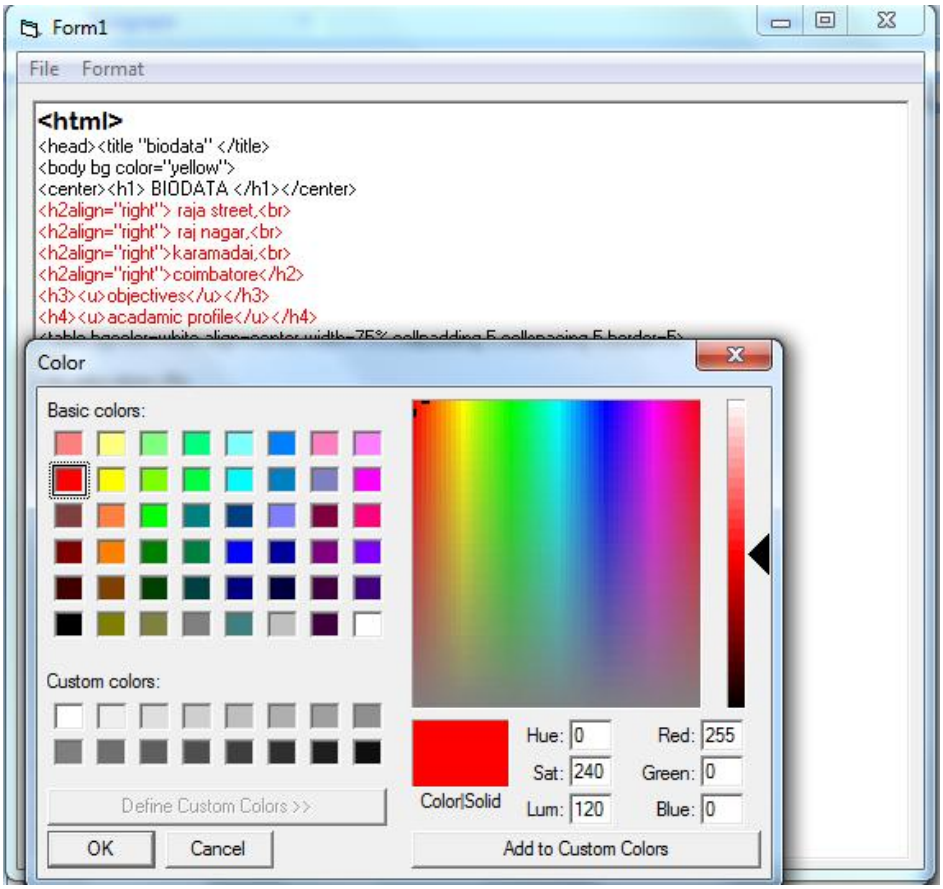


Coding

```
Private Sub mnucolor_Click()
cdb.ShowColor
rtb.SelColor = cdb.Color
End Sub
Private Sub mnuexit_Click()
End
End Sub
Private Sub mnufont_Click()
cdb.Flags = cdlCFBoth Or cdlCFLeffects
cdb.ShowFont
rtb.SelFontName = cdb.FontName
rtb.SelFontSize = cdb.FontSize
rtb.SelBold = cdb.FontBold
rtb.SelItalic = cdb.FontItalic
rtb.SelUnderline = cdb.FontStrikethru
End Sub

Private Sub mnuopen_Click()
rtb.Visible = True
cdb.ShowOpen
rtb.LoadFile (cdb.FileName)
End Sub
Private Sub mnusave_Click()
cdb.ShowSave
rtb.SaveFile (cdb.FileName)
MsgBox ("FILE SAVED" & cdb.FileName)
End Sub

Private Sub mnunew_Click()
rtb.Visible = True
rtb.Text = " "
End Sub
```

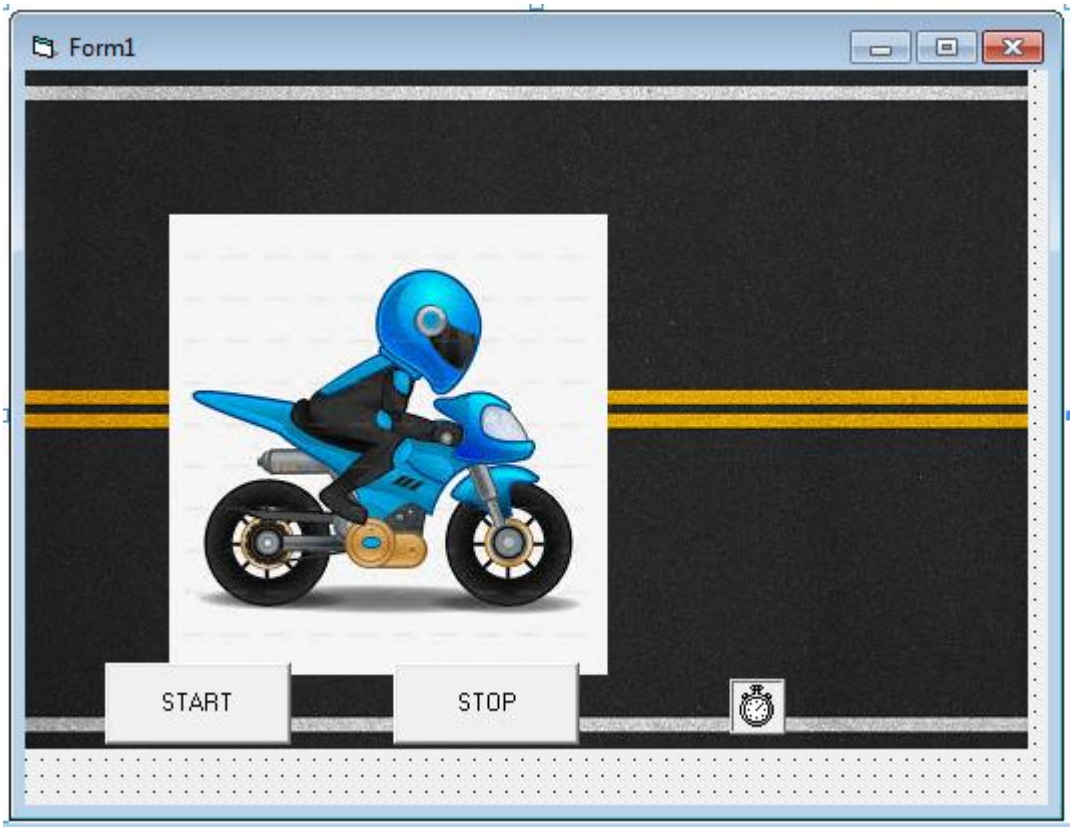
/*** OUTPUT - TEXT FILES ***/





/*** 6. ANIMATION ***/

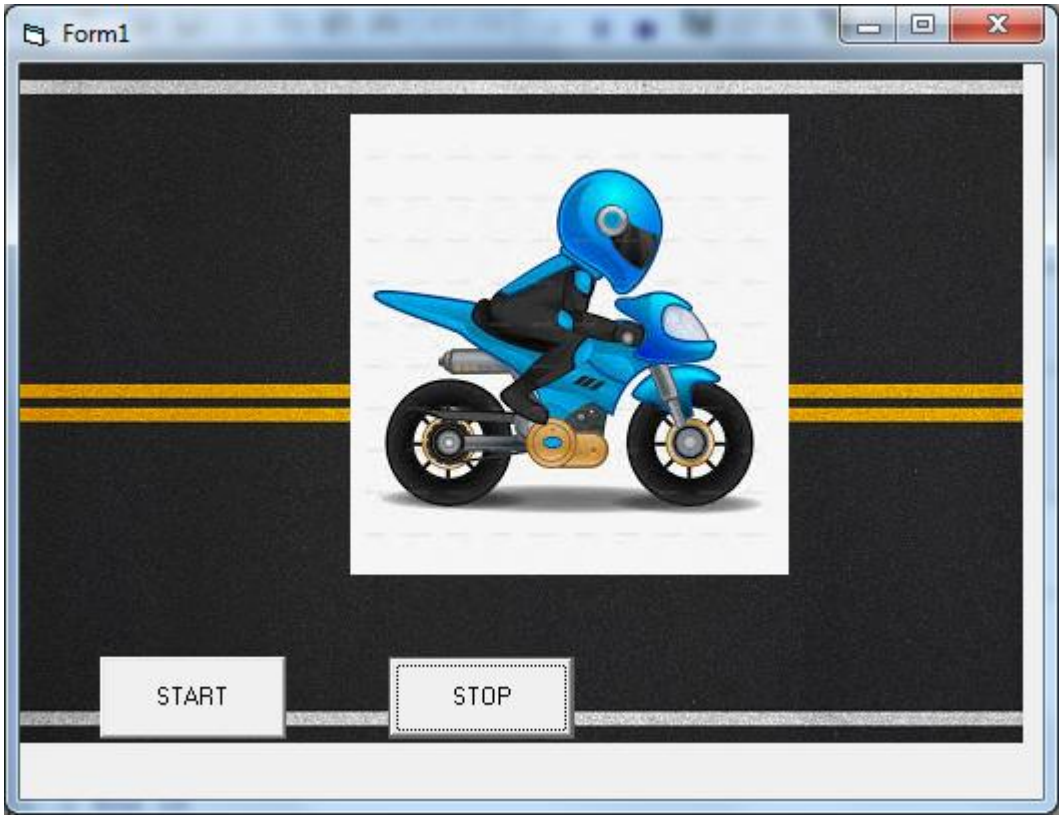
Form Design



Coding

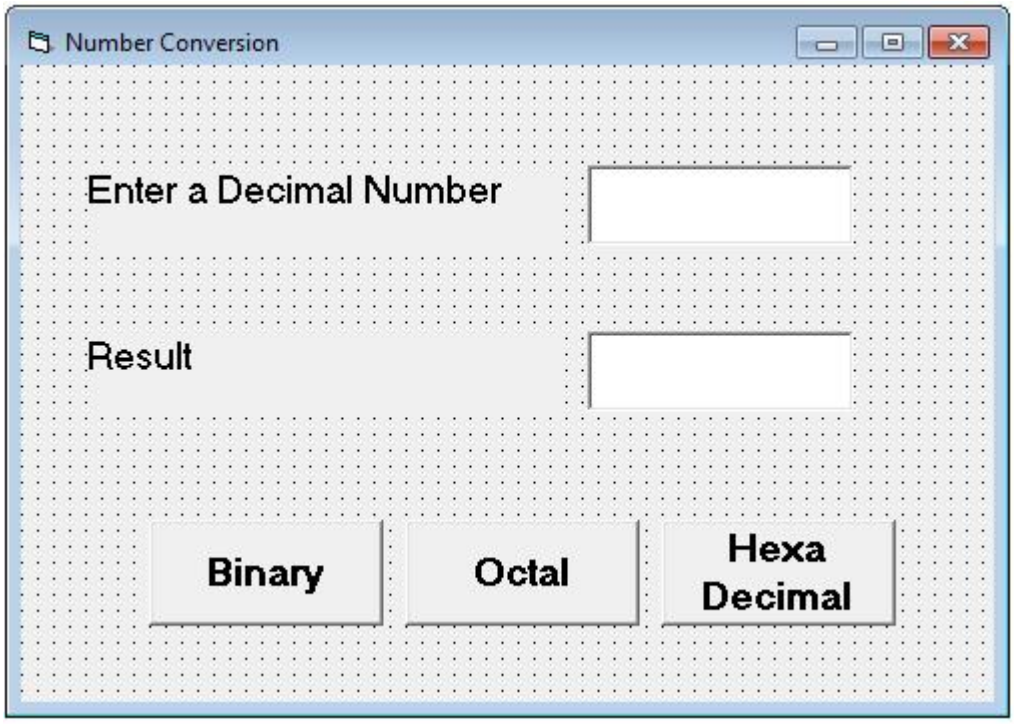
```
Private Sub Form_Load()
Timer1.Enabled = False
End Sub

Private Sub start_Click()
Timer1.Enabled = True
End Sub

Private Sub stop_Click()
Timer1.Enabled = False
End Sub

Private Sub Timer1_Timer()
Image1.Left = Image1.Left + 10
If Image1.Left >= 1000 Then
Image1.Left = Image1.Left + 10
Image1.Top = Image1.Top - 10
End If
End Sub
```

/ OUTPUT- ANIMATION **/**



/*** 7. NUMBER CONVERSION ***/

Form Design

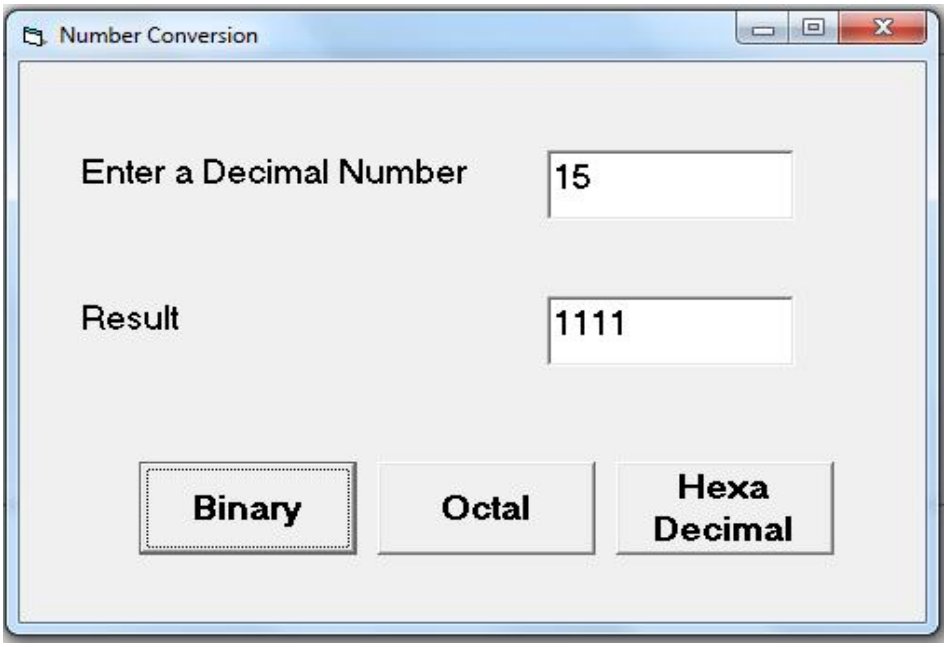


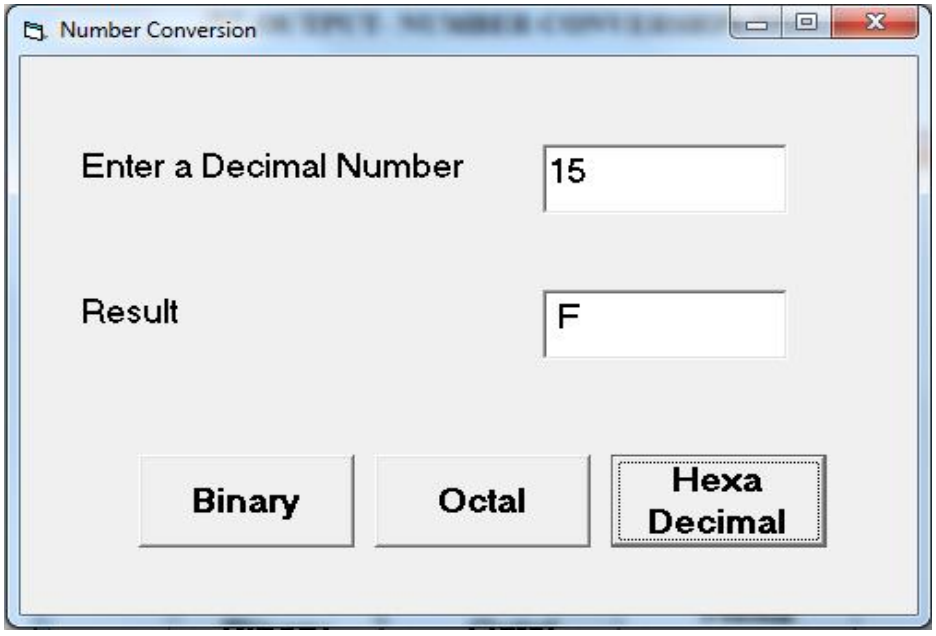
Coding

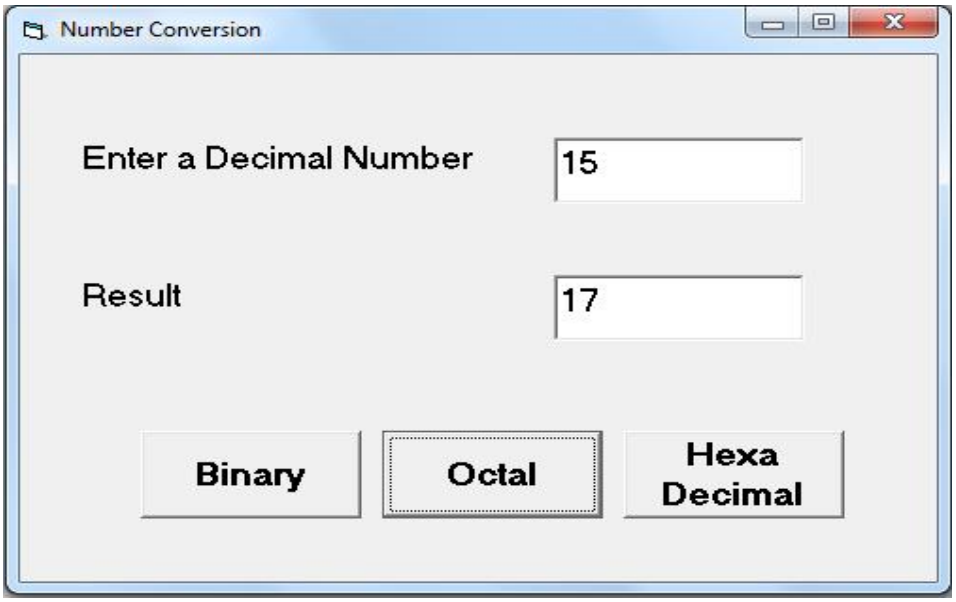
```
Dim result, no, r As Integer
Private Sub Cmdbin_Click()
convert (2)
End Sub
Private Sub Cmdoct_Click()
convert (8)
End Sub
Private Sub Cmdhex_Click()
convert (16)
End Sub
```

```
Public Sub convert(value As Integer)
result = " "
no = 0
no = Val(Text1.Text)
While no > 0
  r = no Mod value
  no = no \ value
  If  r > 9 Then
     result = result & Chr(r + 55)
  Else
     result = r & result
 End If
Wend
Text2.Text = result
End Sub
```

/** OUTPUT- NUMBER CONVERSION **/







/*** 8.EMPLOYEE DETAILS USING DML ***/

CREATION OF EMPLOYEE TABLE

SQL> create table employee(emp_no number(3) primary key, name varchar2(30), designation varchar2(30), gender varchar2(7), age number(3), date_of_joining date, salary number(6));

Table created.

TO VIEW TABLE STRUCTURE

SQL> desc employee;

Name	Null?	Type
EMP_NO	NOT NULL	NUMBER(3)
NAME		VARCHAR2(30)
DESIGNATION		VARCHAR2(30)
GENDER		VARCHAR2(7)
AGE		NUMBER(3)
DATE_OF_JOINING		DATE
SALARY		NUMBER(6)

INSERTING RECORDS INTO TABLE

SQL> insert into employee values(&emp_no , '&name' , '&designation' , '&gender' , &age , '&date_of_joining' , &salary);

Enter value for emp_no: 111
Enter value for name: smith
Enter value for designation: manager
Enter value for gender: male
Enter value for age: 50
Enter value for date_of_joining: 04-jun-60
Enter value for salary: 265000

old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(111,'smith','manager','male',50,'04-jun-60',265000)

1 row created.

/*** 8.EMPLOYEE DETAILS USING DML ***/

SQL> /
Enter value for emp_no: 112
Enter value for name: larry
Enter value for designation: worker
Enter value for gender: male
Enter value for age: 30
Enter value for date_of_joining: 05-dec-97
Enter value for salary: 12000
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(112,'larry','worker','male',30,'05-dec-97',12000)

1 row created.

SQL> /
Enter value for emp_no: 113
Enter value for name: roberts
Enter value for designation: worker
Enter value for gender: male
Enter value for age: 35
Enter value for date_of_joining: 12-feb-90
Enter value for salary: 18000
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(113,'roberts','worker','male',35,'12-feb-90',18000)

1 row created.

SQL> /
Enter value for emp_no: 114
Enter value for name: rose
Enter value for designation: supervisor
Enter value for gender: female
Enter value for age: 40
Enter value for date_of_joining: 08-jan-85
Enter value for salary: 24500
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(114,'rose','supervisor','female',40,'08-jan-85',24500)

1 row created.

/*** 8.EMPLOYEE DETAILS USING DML ***/

SQL> /
Enter value for emp_no: 115
Enter value for name: aasha
Enter value for designation: worker
Enter value for gender: female
Enter value for age: 26
Enter value for date_of_joining: 04-mar-00
Enter value for salary: 10000
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(115,'aasha','worker','female',26,'04-mar-00',10000)

1 row created.

SQL> /
Enter value for emp_no: 116
Enter value for name: alen
Enter value for designation: worker
Enter value for gender: female
Enter value for age: 29
Enter value for date_of_joining: 05-nov-97
Enter value for salary: 12000
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(116,'alen','worker','female',29,'05-nov-97',12000)

1 row created.

SQL> /
Enter value for emp_no: 117
Enter value for name: alex
Enter value for designation: supervisor
Enter value for gender: male
Enter value for age: 42
Enter value for date_of_joining: 18-aug-82
Enter value for salary: 29000
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(117,'alex','supervisor','male',42,'18-aug-82',29000)

1 row created.

/*** 8.EMPLOYEE DETAILS USING DML ***/

SQL> /
Enter value for emp_no: 118
Enter value for name: shaw
Enter value for designation: worker
Enter value for gender: male
Enter value for age: 36
Enter value for date_of_joining: 02-jan-95
Enter value for salary: 18000
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(118,'shaw','worker','male',36,'02-jan-95',18000)

1 row created.

SQL> /
Enter value for emp_no: 119
Enter value for name: dev
Enter value for designation: worker
Enter value for gender: male
Enter value for age: 45
Enter value for date_of_joining: 04-jan-82
Enter value for salary: 18000
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(119,'dev','worker','male',45,'04-jan-82',18000)

1 row created.

SQL> /
Enter value for emp_no: 120
Enter value for name: sheela
Enter value for designation: asst.manager
Enter value for gender: female
Enter value for age: 35
Enter value for date_of_joining: 12-dec-99
Enter value for salary: 28000
old 1: insert into employee
values(&emp_no,'&name','&designation','&gender',&age,'&date_of_joining',&salary)
new 1: insert into employee values(120,'sheela','asst.manager','female',35,'12-dec-99',28000)

1 row created.

/*** 8.EMPLOYEE DETAILS USING DML ***/

DISPLAYING TABLE RECORDS

SQL> set linesize 200;

SQL> select * from employee;

EMP_NO	NAME	DESIGNATION	GENDER	AGE	DATE_OF_J	SALARY
111	smith	manager	male	50	04-JUN-60	265000
112	larry	worker	male	30	05-DEC-97	12000
113	roberts	worker	male	35	12-FEB-90	18000
114	rose	supervisor	female	40	08-JAN-85	24500
115	aasha	worker	female	26	04-MAR-00	10000
116	alen	worker	female	29	05-NOV-97	12000
117	alex	supervisor	male	42	18-AUG-82	29000
118	shaw	worker	male	36	02-JAN-95	18000
119	dev	worker	male	45	04-JAN-82	18000
120	sheela	asst.manager	female	35	12-DEC-99	28000

10 rows selected.

RESTRICTING TABLE RECORDS USING WHERE CLAUSE WITH RELATIONAL AND LOGICAL OPERATOR

SQL> select * from employee where gender = 'male';

EMP_NO	NAME	DESIGNATION	GENDER	AGE	DATE_OF_J	SALARY
111	smith	manager	male	50	04-JUN-60	265000
112	larry	worker	male	30	05-DEC-97	12000
113	roberts	worker	male	35	12-FEB-90	18000
117	alex	supervisor	male	42	18-AUG-82	29000
118	shaw	worker	male	36	02-JAN-95	18000
119	dev	worker	male	45	04-JAN-82	18000

6 rows selected.

/*** 8.EMPLOYEE DETAILS USING DML ***/

SQL> select * from employee where gender='female' and salary < 20000;

EMP_NO	NAME	DESIGNATION	GENDER	AGE	DATE_OF_J	SALARY
115	aasha	worker	female	26	04-MAR-00	10000
116	alen	worker	female	29	05-NOV-97	12000

SORTING

SQL> select emp_no, name, designation from employee order by name desc;

EMP_NO	NAME	DESIGNATION
111	smith	manager
120	sheela	asst.manager
118	shaw	worker
114	rose	supervisor
113	roberts	worker
112	larry	worker
119	dev	worker
117	alex	supervisor
116	alen	worker
115	aasha	worker

10 rows selected.

SQL> select emp_no,name,designation from employee order by name;

EMP_NO	NAME	DESIGNATION
115	aasha	worker
116	alen	worker
117	alex	supervisor
119	dev	worker
112	larry	worker
113	roberts	worker
114	rose	supervisor
118	shaw	worker
120	sheela	asst.manager
111	smith	manager

10 ows selected.

/*** 8.EMPLOYEE DETAILS USING DML ***/

SET OPERATION

```
SQL>     select * from employee where gender = 'male' and salary > 20000
      2  union
      3  select * from employee where gender = 'female' and salary < 20000;
```

EMP_NO	NAME	DESIGNATION	GENDER	AGE	DATE_OF_J	SALARY
111	smith	manager	male	50	04-JUN-60	265000
115	aasha	worker	female	26	04-MAR-00	10000
117	alex	supervisor	male	42	18-AUG-82	29000

GROUPING FUNCTION

```
SQL>     select min(salary) MINIMUM_SALARY ,   max(salary) MAXIMUM_SALARY,
      2   avg(salary) AVERAGE_SALARY  from employee;
```

MINIMUM_SALARY	MAXIMUM_SALARY	AVERAGE_SALARY
10000	265000	43450

```
SQL> commit;
```

Commit complete.

```
SQL> exit
```

/*** 9. INVENTORY TABLE UPDATION USING PL/SQL ***/

CREATION OF INVENTORY TABLE

SQL> create table inventory(prono number(3) primary key,proname varchar2(25),rate number(8,2));

Table created.

TO VIEW TABLE STRUCTURE

SQL> desc inventory;

Name	Null?	Type
PRONO	NOT NULL	NUMBER(3)
PRONAME		VARCHAR2(25)
RATE		NUMBER(8,2)

INSERTING RECORDS INTO TABLE

SQL> insert into inventory values(&Prono,'&Proname',&Rate);
Enter value for prono: 1
Enter value for proname: keyboard
Enter value for rate: 500
old 1: insert into inventory values(&Prono,'&Proname',&Rate)
new 1: insert into inventory values(1,'Keyboard',500)

1 row created.

SQL> /
Enter value for prono: 2
Enter value for proname: monitor
Enter value for rate: 6000
old 1: insert into inventory values(&Prono,'&Proname',&Rate)
new 1: insert into inventory values(2,'monitor',6000)

1 row created.

SQL> /
Enter value for prono: 3
Enter value for proname: mouse
Enter value for rate: 200
old 1: insert into inventory values(&Prono,'&Proname',&Rate)
new 1: insert into inventory values(3,'mouse',200)

1 row created.

/*** 9. INVENTORY TABLE UPDATION USING PL/SQL ***/

```
SQL> /
Enter value for prono: 4
Enter value for proname: pendrive
Enter value for rate: 800
old   1: insert into inventory values(&Prono,'&Proname',&Rate)
new   1: insert into inventory values(4,'pendrice',800)

1 row created.

SQL> /
Enter value for prono: 5
Enter value for proname: cpu
Enter value for rate: 15000
old   1: insert into inventory values(&Prono,'&Proname',&Rate)
new   1: insert into inventory values(5,'cpu',15000)

1 row created.
```

DISPLAYING TABLE RECORDS

SQL> select * from inventory;

PRONO	PRONAME	RATE
1	keyboard	500
2	monitor	6000
3	mouse	200
4	pendrive	800
5	cpu	15000

SQL> ed up_rate;

```
begin
update inventory
set rate = rate + rate * 20 / 100;
end;
/
```

SQL> @up_rate;

PL/SQL procedure successfully completed.

/*** 9. INVENTORY TABLE UPDATION USING PL/SQL ***/

DISPLAYING TABLE RECORDS AFTER UPDATEING RATE

SQL> select * from inventory;

PRONO	PRONAME	RATE
1	Keyboard	600
2	monitor	7200
3	mouse	240
4	pendrive	960
5	cpu	18000

ADDING NEW COLUMN INTO A TABLE

SQL> alter table inventory add no_of_item number(3);

Table altered.

SQL> desc inventory;

Name	Null?	Type
PRONO	NOT NULL	NUMBER(3)
PRONAME		VARCHAR2(25)
RATE		NUMBER(8,2)
NO_OF_ITEM		NUMBER(3)

UPDATEING THE VALUE FOR NEW COLUMN

SQL> **update inventory set no_of_item =&no_of_item where prono=&prono;**

Enter value for no_of_item: 2
old 1: update inventory set no_of_item =&no_of_item
new 1: update inventory set no_of_item =2
Enter value for prono: 1
old 2: where prono=&prono
new 2: where prono=1

1 row updated.

/*** 9. INVENTORY TABLE UPDATION USING PL/SQL ***/

```
SQL> /
Enter value for no_of_item: 3
old   1: update inventory set no_of_item =&no_of_item
new   1: update inventory set no_of_item =3
Enter value for prono: 2
old   2: where prono=&prono
new   2: where prono=2

1 row updated.

SQL> /
Enter value for no_of_item: 10
old   1: update inventory set no_of_item =&no_of_item
new   1: update inventory set no_of_item =10
Enter value for prono: 3
old   2: where prono=&prono
new   2: where prono=3

1 row updated.

SQL> /
Enter value for no_of_item: 6
old   1: update inventory set no_of_item =&no_of_item
new   1: update inventory set no_of_item =6
Enter value for prono: 4
old   2: where prono=&prono
new   2: where prono=4

1 row updated.

SQL> /
Enter value for no_of_item: 15
old   1: update inventory set no_of_item =&no_of_item
new   1: update inventory set no_of_item =15
Enter value for prono: 5
old   2: where prono=&prono
new   2: where prono=5

1 row updated.
```

/*** 9. INVENTORY TABLE UPDATION USING PL/SQL ***/

DISPLAYING TABLE RECORDS AFTER UPDATION

SQL> select * from inventory;

PRONO	PRONAME	RATE	NO_OF_ITEM
1	Keyboard	600	2
2	monitor	7200	10
3	mouse	240	3
4	pendrice	960	6
5	cpu	18000	15

SQL> commit;

Commit complete.

SQL>

/*** 10. IMPLEMENTATION OF TRIGGERS ***/

CREATION OF INVENTORY MASTER TABLE

```
SQL>create table inventory_master(prono number(4) primary key, proname
  2 varchar2(30),rate number(8,2),avail_qty number(4));
```
Table created.

CREATION OF INVENTORY TRANSACTION TABLE

```
SQL>create table inventory_trans(prono number(4) references inventory_master(prono)
  2 purchase_date date,pur_qty number(4));
```
Table created

CREATION OF TRIGGER FOR DATA VALIDATION

```
SQL>ed trigger1
  Create or replace trigger inv_master_trigger
Before insert or update on inventory_master for each row
Begin
If(:new.rate<=0)then
Raise_application_error(-20001,'invalid rate value');
End if;
End;
/
SQL>@trigger1
```
Trigger created

```
SQL>insert into inventory_master values(&prono,'&proname',&rate,&avail_qty);
```
Enter value for prono:5
Enter value for proname:pencil
Enter value for rate:2.50
Enter value for avail_qty:25
Old 1:insert into inventory_master values(&prono,'&proname',&rate,&avail_qty)
New 1:insert into inventory_master values(5,'pencil',2.50,25)
1 row created
SQL>/
Enter value for prono:6
Enter value for proname:geometric box
Enter value for rate:0
Enter value for avail_qty:4
Old 1:insert into inventory_master values(&prono,'&proname',&rate,&avail_qty)
New 1:insert into inventory_master values(6,'geometric box',0,4)
*

ERROR at line 1
ORA-20001:invalid rate value
ORA-06512:at “14BSCA01.INV_MASTER_TRIGGER”,line3
ORA-04088:error during execution of trigger’14BSCA01.INV_MASTER_TRIGGER’

SQL>select * from inventory_master;

PRONO	PRONAME	RATE	AVAIL_QYT
1	rice	40	45
2	pen	20	36
3	eraser	2	4
5	pencil	2.5	25

SQL> edit trigger2
Create or replace trigger inv_trans_trigger
Before insert or update on inventory_trans for each row
Begin
If(:new.pur_rate<=0)then
Raise_application_error(-20001,’invalid purchase quantity’);
End if;
End;
/
SQL> @ trigger2

Trigger created.

SQL>insert into inventory_trans values(&prono,’&purchase_date’,&pur_qty);
Enter value for prono:1
Enter value for purchasedate:26-sep-16
Enter value for pur_qty:0
Old 1:insert into inventory_trans values(&prono,’&purchase_date’,&pur_qty)
New 1:insert into inventory_trans values(1,’26-sep-16,0)
*
ERROR at line 1
ORA-20001:invalid purchase quantity
ORA-06512:at “14BSCA01.INV_TRANS_TRIGGER”,line3
ORA-04088:error during execution of trigger’14BSCA01.INV_TRANS_TRIGGER’

SQL>/
Enter value for prono:1
Enter value for purchasedate:13-oct-16
Enter value for pur_qty:3
1 row Created.

/*** 10. IMPLEMENTATION OF TRIGGERS ***/

SQL> select * from inventory_tans;

PRONO	PURCHASE_	PUR_QTY
2	27-SEP-16	56
1	13-OCT-16	3

SQL>commit;
Commit completed.

SQL>exit

/*** 11. IMPLEMENTATION OF PROCEDURES ***/

SQL> set serveroutput on;

```
declare
a number;
b number;
c number;
procedure findmin(x IN number,y IN number,z OUT number)
IS
begin
if x<y then
z:=x;
else
z:=y;
end if;
end;
begin
a:=23;
b:=45;
findmin(a,b,c);
DBMS_OUTPUT.PUT_LINE('minimum of (23,45):'||c);
end;
/
```

SQL>minimum of (23,45):23

SQL>exit

/*** 12. STUDENT DATABASE MANAGEMENT SYSTEM ***/

ORACLE Database Creation

SQL> connect

Enter user-name: system

Enter password:

Connected.

SQL> create table student1 (sno number(3), sname varchar2(20),m1 number(3), m2 n

umber(3), m3 number(3));

Table created.

SQL> desc student;

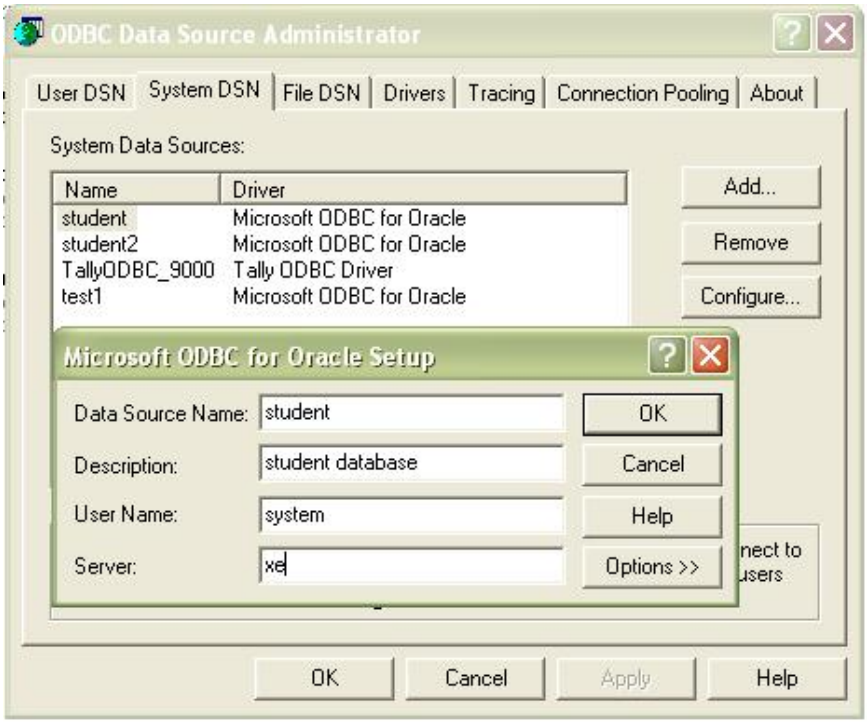
Name	Null?	Type
SNO		NUMBER(10)
SNAME		VARCHAR2(20)
M1		NUMBER(3)
M2		NUMBER(3)
M3		NUMBER(3)

SQL> select * from student;

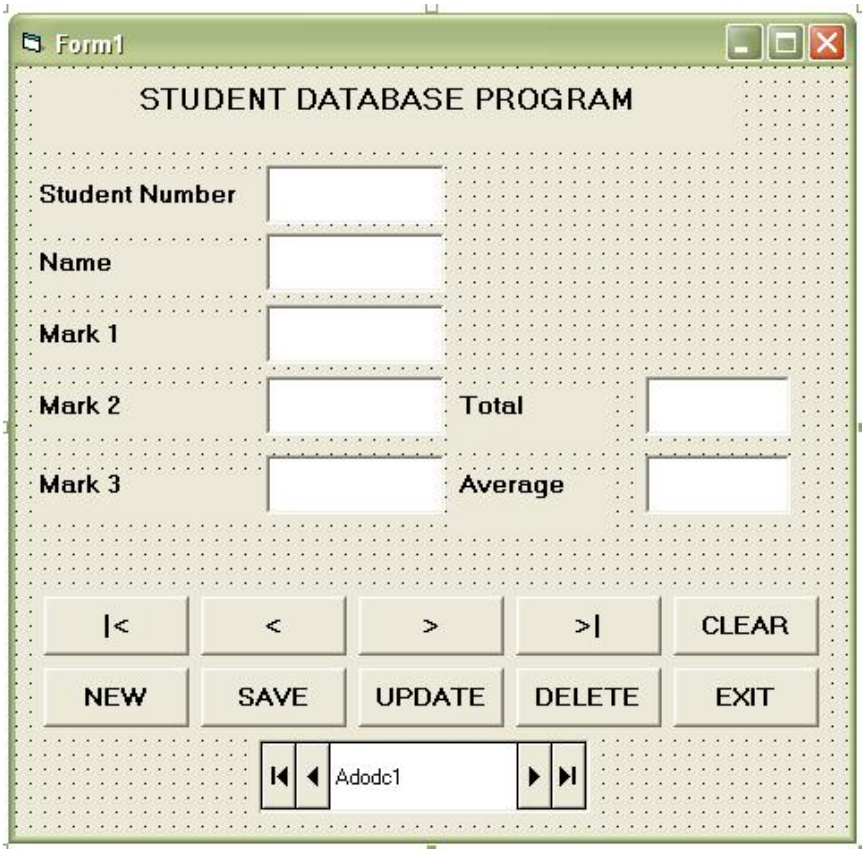
SNO	SNAME	M1	M2	M3
115	Elizabeth	88	77	88
118	Ramu	60	75	65
111	arun	88	99	77
112	banu	88	99	77
113	chitra	99	99	99

5 rows selected.

ODBC for Oracle setup



Form Design



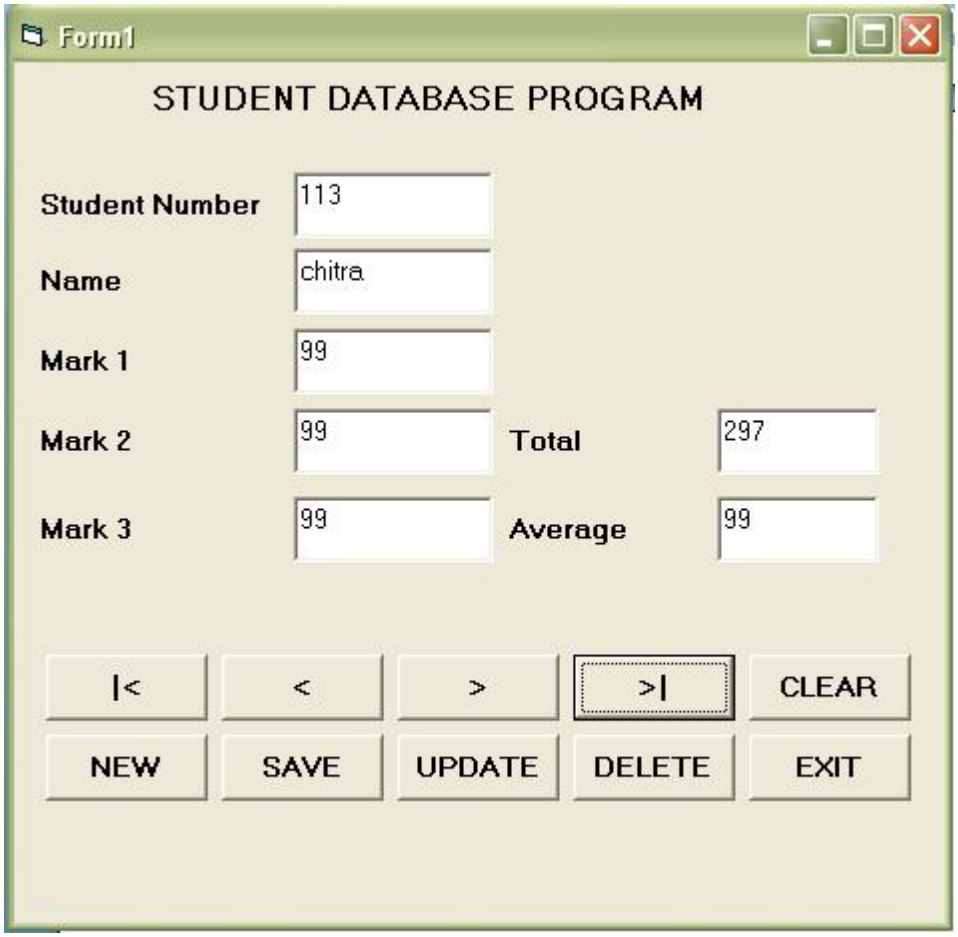
CODING IN FRONTEND TO ACCESS BACKEND

```
Dim con As New ADODB.Connection
Dim rs As New ADODB.Recordset

Private Sub cmdclear_Click()
Text1 = ""
Text2 = ""
Text3 = ""
Text4 = ""
Text5 = ""
Text6 = ""
Text7 = ""
Text1.SetFocus
End Sub

Private Sub cmddelete_Click()
rs.Delete
MsgBox "Record Deleted"
End Sub

Private Sub cmdexit_Click()
End
End Sub

Private Sub cmdfirst_Click()
rs.MoveFirst
Call display
End Sub

Private Sub cmdlast_Click()
rs.MoveLast
Call display
End Sub

Private Sub cmdnew_Click()
Call cmdclear_Click
End Sub

Private Sub cmdnext_Click()
rs.MoveNext
If rs.EOF Then
MsgBox "Last Record"
Exit Sub
Else
Call display
End If
End Sub



Private Sub cmdprevious_Click()
rs.MovePrevious
```

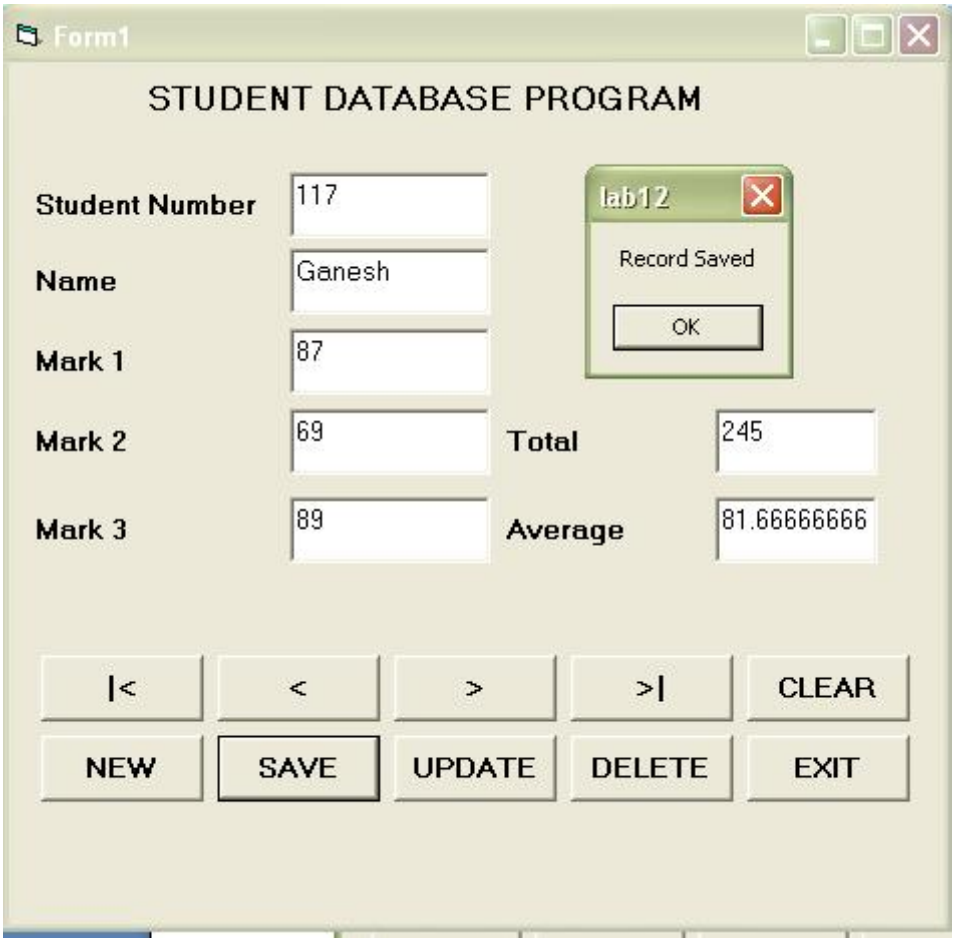
```
If rs.BOF Then
MsgBox "First Record"
Exit Sub
Else
Call display
End If
End Sub

Private Sub cmdsave_Click()
rs.AddNew
rs.Fields(0) = Text1
rs.Fields(1) = Text2
rs.Fields(2) = Text3
rs.Fields(3) = Text4
rs.Fields(4) = Text5
strsql = "insert into student(sno,sname,m1,m2,m3) values('" & rs.Fields(0).Value & "' ,
'" & rs.Fields(1).Value & "','" & rs.Fields(2).Value & "','" & rs.Fields(3).Value & "',
'" & rs.Fields(4).Value & "')"
Set rs = con.Execute(strsql)
MsgBox "Record Saved"
End Sub

Private Sub cmdupdate_Click()
rs.Fields(0) = Text1
rs.Fields(1) = Text2
rs.Fields(2) = Text3
rs.Fields(3) = Text4
rs.Fields(4) = Text5
rs.Update
MsgBox "Record Updated"
End Sub


Private Sub Form_Load()
con.Open "student", "system", "manager"
rs.Open "select * from student", con, adOpenDynamic, adLockOptimistic
Adodc1.Visible = False
End Sub


Private Sub Text5_Change()
Text6 = Val(Text3) + Val(Text4) + Val(Text5)
Text7 = Val(Text6) / 3
End Sub


Public Sub display()
Text1.Text = rs.Fields("sno")
Text2 = rs.Fields(1)
Text3 = rs.Fields(2)
Text4 = rs.Fields(3)
Text5 = rs.Fields(4)
End Sub
```

/** OUTPUT - STUDENT DATABASE MANAGEMENT SYSTEM **/

Output – Navigating Records



Output – Adding new



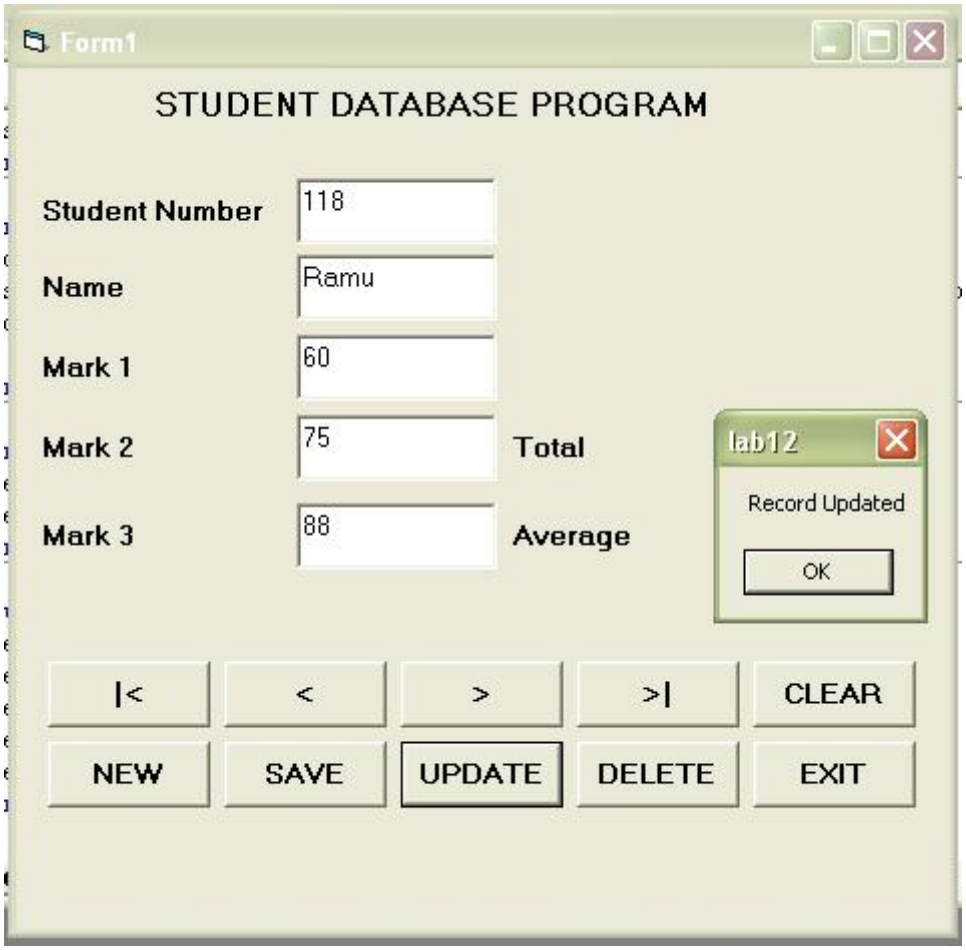
Records

SQL> select * from student;

SNO	SNAME	M1	M2	M3
115	Elizabeth	88	77	88
118	Ramu	60	75	65
114	dworthy	88	88	88
117	Ganesh	87	69	89
111	arun	88	99	77
112	banu	88	99	77
113	chitra	99	99	99

7 rows selected.

Output – Updating a Record

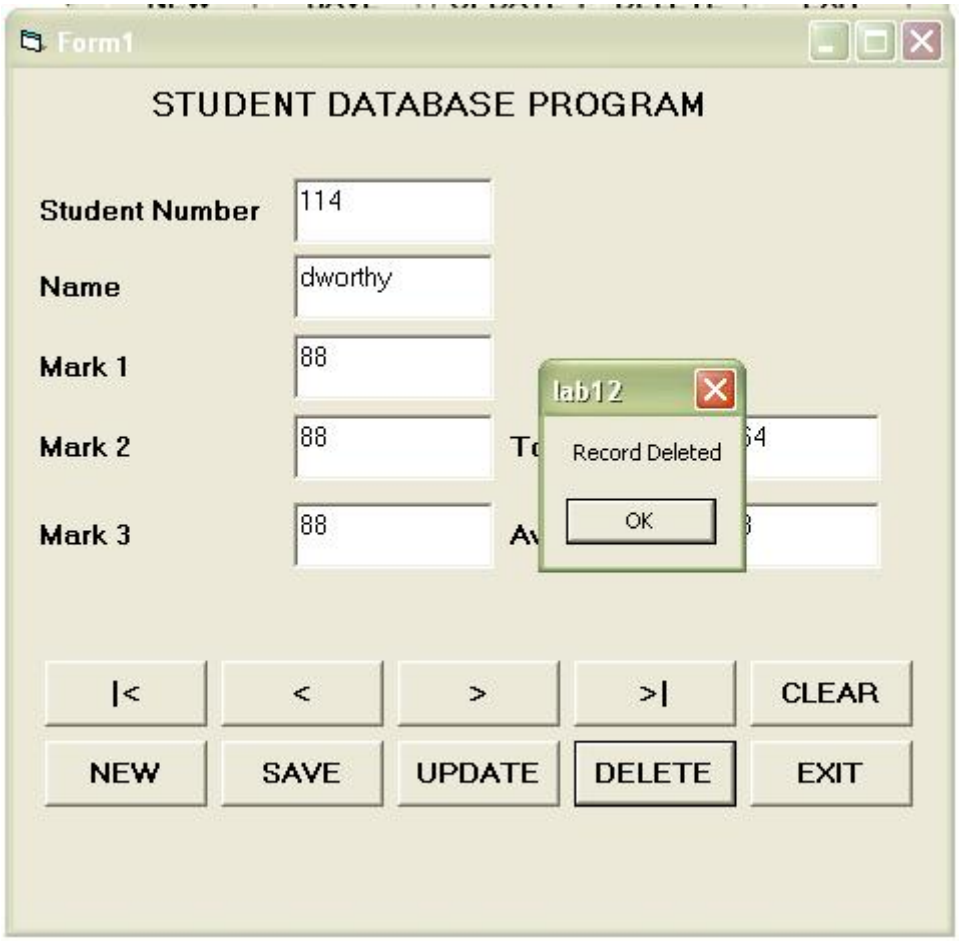


SQL> select * from student;

SNO	SNAME	M1	M2	M3
115	Elizabeth	88	77	88
118	Ramu	60	75	88
114	dworthy	88	88	88
117	Ganesh	87	69	89
111	arun	88	99	77
112	banu	88	99	77
113	chitra	99	99	99

7 rows selected.

Output – Deleting a Record



SQL> select * from student;

SNO	SNAME	M1	M2	M3
115	Elizabeth	88	77	88
118	Ramu	60	75	88
117	Ganesh	87	69	89
111	arun	88	99	77
112	banu	88	99	77
113	chitra	99	99	99

6 rows selected.