

## ជំពូកទី២ :

# ការធ្វើការជាមួយ File

នៅក្នុងមេរៀននេះលោកអ្នកនឹងធ្វើការសិក្សាពី :

☞ Sequential File

☞ របៀបបញ្ចូលទិន្នន័យទៅអោយ Sequential File

☞ របៀបអានទិន្នន័យចេញពី Sequential File

☞ ការប្រើអនុគមន៍ដែលទាក់ទងនឹង File

☞ លំហាត់អនុវត្តន៍លើ Sequential File (អាន លប់ ស្វែងរក និង បំប្លែងទិន្នន័យ

ដែលមាននៅក្នុង Sequential File អោយខុសពីទិន្នន័យដើម “ Encrypt Data ”)

☞ Random Access File

☞ របៀបបញ្ចូលទិន្នន័យទៅអោយ Random Access File

☞ របៀបអានទិន្នន័យចេញពី Random Access File

☞ លំហាត់អនុវត្តន៍លើ Random Access

ជាទូទៅរាល់ឯកសារដែលលោកអ្នកចង់រក្សាទុកនៅក្នុង Computer លោកអ្នក ត្រូវតែ Save ជា File ទៅតាម Extension ណាមួយដែលលោកអ្នកចង់បាន។

ចំពោះភាសា Visual Basic វិញដើម្បីទាញទិន្នន័យពី File ឬ បញ្ជូនទិន្នន័យទៅក្នុង File មានបីរបៀបគឺ:

- Sequential File
- Random Access File
- Binary File

## I. Sequential File

ចំពោះ Sequential File ការអាន និងការបញ្ជូនទិន្នន័យរបស់វា គឺទៅតាម លំដាប់លំដោយដូចជា កាសែតម៉ាញ៉េ កាសែតវីដេអូ ឬក៏ខ្សែអាត់ជាដើម។

### I.1. របៀបបើក Sequential File

ជាទូទៅលោកអ្នកបើក File ក្នុងគោលបំណងបីយ៉ាងគឺ :

- ◆ ដើម្បីបញ្ជូនទិន្នន័យទៅអោយ File ( Write data to file )
- ◆ ដើម្បីបញ្ជូនទិន្នន័យបន្ថែមទៅអោយ File ( Append data to file)
- ◆ ដើម្បីអានទិន្នន័យពី File ( Read data from file)

Syntax:

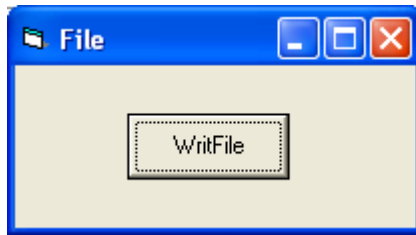
Open *FileNameAndPath* For *Mode* As #*FNumber*

| សមាសភាពនៅក្នុង Syntax  | អធិប្បាយ   |
|------------------------|--|
| Open                   | ជា Statement សំរាប់បើក File ។  |
| <i>FileNameAndPath</i> | ជាទីតាំងនិងឈ្មោះរបស់ File ដែលស្ថិត នៅក្នុង Hard Disk ។   |
| For <i>Mode</i>        | ជាទំរង់នៃ File ដែលលោកអ្នកត្រូវបើក ដែលទំរង់របស់វាបែងចែកចេញជាបីគឺ For OutPut, For Input និង For Append ។ |
| For OutPut             | បើក File ដើម្បីបញ្ជូន ទិន្នន័យ (Writ data to file or Over write data to file)                          |

|                |   |
|----------------|---|
| For Inuput     | បើក File ដើម្បីអានទិន្នន័យ (Read data from file )                       |
| For Append     | បើក File ដើម្បីបញ្ចូល ទិន្នន័យ បន្ថែម (Append data to file)             |
| <i>FNumber</i> | សំរាប់សំគាល់ទៅលើលក្ខណៈរបស់ File នីមួយៗ<br>ដោយកំណត់ជាលេខពី (1 ដល់ 512) ។ |

**ឧទាហរណ៍ទី១:** ឧបមាថាលោកអ្នកមាន Form ដូចរូបភាពទី៧.១ :

**រូបភាពទី៧.១**



☞ Source Code:

```
Option Explicit
Dim FName As String
```

---

```
Private Sub CmdWriteFile_Click( )
    FName = InputBox("Please, enter Paht and File Name")
    If FName = "" Then
        MsgBox "You must enter Path and file name", vbCritical
        Exit Sub
    Else
        Open FName For Output As #1
    End If
End Sub
```

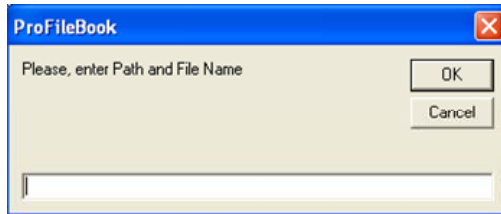
---

**វិភាគទៅលើលក្ខណៈ :**

នៅពេលដែលលោកអ្នកចុចទៅលើ Button WriteFile ពេលនោះ Event On Click កើតឡើងទៅលើ WriteFile Button ដែលជំហានធ្វើការរបស់វាមានជា បន្តបន្ទាប់ដូចខាងក្រោម:

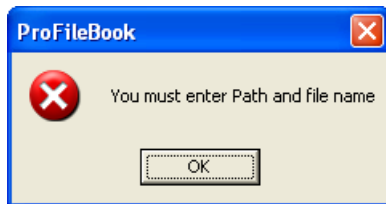
▪ FName = InputBox("Please, enter Paht and File Name") នៅពេលដែល Code នេះដំណើរការវានឹងមានផ្ទាំង Dialog Box មួយបង្ហាញ ដូចរូប ទី៧.២ ដែលទាមទារអោយ លោកអ្នកវាយ បញ្ចូលទីតាំងរបស់ File និងឈ្មោះរបស់ File ។

**រូបភាពទី៧.២**



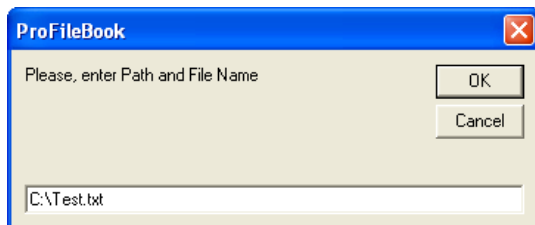
▪ If FName = "" Then ចំពោះ Code នេះគឺយើងធ្វើការពិនិត្យមើលលក្ខខណ្ឌ ថាតើលោកអ្នកបានវាយបញ្ចូល Path របស់ File និង ឈ្មោះរបស់ File ទៅក្នុង Input Box ហើយ ឬ នៅ។ ប្រសិនបើលោកអ្នក មិនបានវាយបញ្ចូលទេ នោះ FName="" ( លក្ខខណ្ឌពិត ) នោះវានឹងបង្ហាញ Message Box ដូចរូបទី៧.៣ ហើយចាកចេញពី Sub Program ។

**រូបភាពទី៧.៣**



ផ្ទុយទៅវិញប្រសិនបើលោកអ្នកបានវាយបញ្ចូល Path File ភ្ជាប់ជាមួយ File Name ដូចរូបទី៧.៤ នោះ FName = "" ( លក្ខខណ្ឌមិនពិត ) ពេលនោះវាអនុវត្ត Else Statement ( Open FName For OutPut As #1) ដែលនៅក្នុង Else Statement នេះលោកអ្នកធ្វើការបើក File ទៅតាម Path និង FileName ដែលលោកអ្នកវាយ បញ្ចូលក្នុង InputBox នៃរូបភាពទី៧.៤។

**រូបភាពទី៧.៤**



ឧបមាថាលោកអ្នកបានវាយបញ្ចូល Path និង File Name ដូចក្នុងរូបភាពទី៧.៤ ដូច្នោះនៅពេលដែល Code (Open Fname For OutPut As #1) ដំណើរការពេល នោះវាផែកមើលថាតើ File Test.txt មាននៅលើ Drive C:\ ដែរ ឬទេ ប្រសិនបើមាន Open statement ចាប់ផ្តើមបើក Test.txt File ក្នុងគោលបំណងដើម្បីបញ្ចូលទិន្នន័យ ហើយវាបានកំណត់យក File Number ស្មើអោយលេខ 1 ដែលតំណាងអោយ Test.txt File ផ្ទុយទៅវិញប្រសិនបើមិនមាន File Test.txt នៅក្នុង Drive C:\ ទេនោះនៅ ពេលដែល Open Statement ដំណើរការ វានឹងបង្កើត File Test.txt នៅលើ Drive C:\ ដោយស្វ័យប្រវត្តិ ហើយទើបវា Open Test.txt File ដើម្បីបញ្ចូលទិន្នន័យ។

**ឧទាហរណ៍ទី២ :**

ឧបមាថាលោកអ្នកមាន Form ដូចរូបភាពទី៧.៥ ។

**រូបភាពទី៧.៥**



☞ Source Code:

```
Option Explicit
Dim Fname As String
```

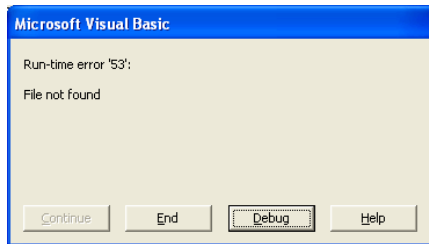
```
.....
Private Sub CmdReadFile_Click()
    Fname = InputBox("Please, enter Paht and File Name")
    If Fname = "" Then
        MsgBox "You must enter Path and file name", vbCritical
        Exit Sub
    Else
        Open Fname For Input As #1
    End If
End Sub
.....
```

**វិភាគទៅលើលទ្ធផល :**

ចំពោះឧទាហរណ៍ទី២ នេះដំណើរការនៃ Code របស់វាស្រដៀងគ្នាទៅនឹង ឧទាហរណ៍ទី១ ដែរ តែខុសគ្នា ត្រង់ថា :

- Code នៅត្រង់ចំណុច (Open Fname For Input As #1) គឺវាបើក File ដើម្បី អានទិន្នន័យមានន័យថាវា មិនអនុញ្ញាតអោយលោកអ្នកបញ្ជូនទិន្នន័យទៅក្នុង File លេខ 1 ឡើយ។
- ប្រសិនបើ File Name ដែលលោកអ្នកបើកមិនមាននៅក្នុង Partition ណាមួយ របស់ Hard Disk ទៅនឹងបង្ហាញផ្ទាំងសារ Error ដូចរូបទី៧.៦ ។

**រូបភាពទី៧.៦**



**ឧទាហរណ៍ទី៣ :**

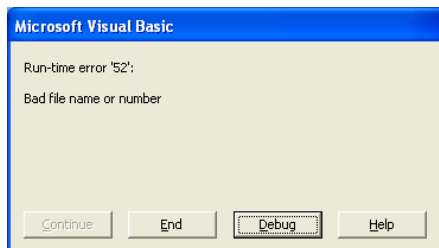
ឧបមាថាលោកអ្នកមាន Form ដូចរូបទី៧.១ ហើយលោកអ្នកសាកល្បងបើក File ដោយកំនត់ File Number ទៅអោយ File Name ដែលមិតនៅខុសពីចន្លោះលេខ 1 ដល់ 512 ពេលនោះវានឹងមានសារ Error កើតឡើងដូចរូបភាពទី៧.៧ ។

☞ Source Code:

```
Option Explicit  
Dim Fname As String
```

```
.....  
Private Sub CmdWriteFile_Click()  
    Open "C:\Test.txt" For Output As #513  
End Sub  
.....
```

## រូបភាពទី៧.៧



### I.2.ប្រូស៊ីន Sequential File

រាល់ពេលដែលលោកអ្នកបើក File ដើម្បី Read, Write ឬ Append រួចហើយ លោកអ្នកត្រូវតែប្រើ

Close Statement ដើម្បីបិទ File ក្នុងគោលបំណង:

- ដើម្បីកំណត់ Pointer អោយវាចង្អុលទៅកាន់ទីបញ្ចប់នៃ File ដោយ Set EOF
- ដើម្បីបញ្ចប់ការ អាន បញ្ចូលទិន្នន័យ ឬ បញ្ចូលទិន្នន័យបន្ថែម

Syntax ទី១:

`Close # FNumber`

Syntax ទី២:

`Close # FNumber1, FNumber2, FNumber3, . . .`

**ឧទាហរណ៍ទី១ :**

ឧបមាថាលោកអ្នកមាន Form ដូចរូបភាពទី៧.៩

☞ Source Code:

```
Option Explicit  
Dim Fname As String
```

```
.....  
Private Sub CmdWriteFile_Click()  
    Open "C:\Test.txt" For Output As #1  
    Close #1  
End Sub  
.....
```

**វិភាគលើលទ្ធផល :**

នៅពេលដែលលោកអ្នកចុចទៅលើ WriteFile Button ពេលនោះ File Test.txt ត្រូវបានបើក ហើយវាត្រូវបានបិទទៅវិញនៅពេលដែលជួប Close #1 ។

**ឧទាហរណ៍ទី២ :**

ឧបមាថាលោកអ្នកមាន Form ដូចរូបភាពទី៧.១

☞ Source Code:

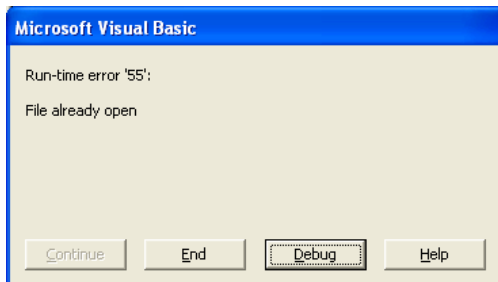
```
Option Explicit
Dim FName As String
```

```
.....
Private Sub CmdWriteFile_Click()
    Open "C:\Test.txt" For Output As #1
End Sub
.....
```

**វិភាគលើលទ្ធផល :**

លទ្ធផលនឹងមានផ្ទាំងសារ Error កើតឡើងដូចរូបភាពទី៧.៨ នៅពេលដែល លោកអ្នកចុច លើសពីមួយលើកទៅលើ WriteFile Button ។ មូលហេតុដែល Error គឺកើតឡើងដោយសារ ពេលដែលលោកអ្នកចុចម្តងទៅលើ WriteFile Button នោះ Test.txt File ត្រូវបានបើករួចហើយ តែលោកអ្នកមិនបានបិទវាវិញទេ ដល់ពេលដែលលោកអ្នកចុចទៅ លើ Button WriteFile ជាលើកទី២ នោះ Test.txt File នឹងត្រូវបានបើកជាលើកទី២ ដែលជាមូលហេតុអោយមានសារ Error មួយបង្ហាញថា File already opened ។

**រូបភាពទី៧.៨**





**ឧទាហរណ៍ទី៣ :**

សូមលោកអ្នកសាកល្បងចូលទៅក្នុង Drive C:\ ហើយបើក Test.txt និង BBU.txt ហើយវាយទិន្នន័យចូល File ជាមុនសិន។

☞ Source Code:

```

Private Sub Form_Load()
    Open "C:\Test.txt" For Append As #1
    Open "C:\BBU.Doc" For Append As #2
    Line Input #1, St1$
    Line Input #2, St2$
    MsgBox St1 & Space(3) & St2
    Close #1, 2
End Sub

```

**I.3. របៀបបញ្ចូលទិន្នន័យទៅអោយ Sequential File**

ដើម្បីបញ្ចូលទិន្នន័យទៅអោយ Sequential File លោកអ្នកអាចប្រើ Statement ពីរគឺ Print ឬ Write ។

**I.3.1. ការប្រើ Print Statement**

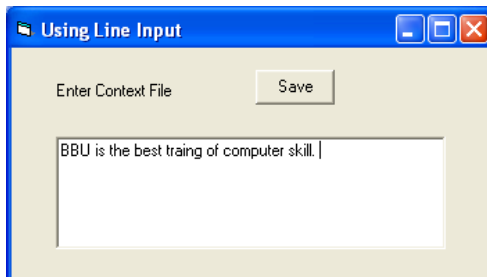
Syntax:

```
Print #Fnumber, TextString
```

**ឧទាហរណ៍ទី១ :**

ឧបមាថាលោកអ្នកមាន Form ដូចរូបទី៧.៩

**រូបភាពទី៩**



**☞ Source Code:**

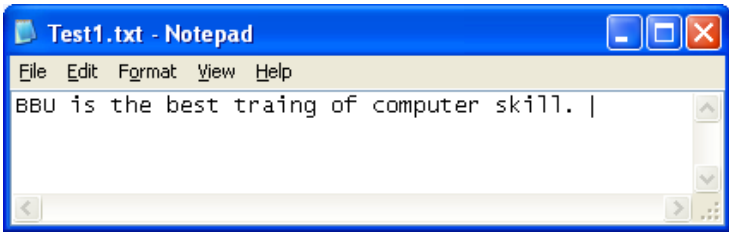
```
Private Sub CmdSave_Click()  
    Open "C:\Test.txt" For Output As #1  
    Print #1, Me.txtContext  
    Close #1  
End Sub
```

**🔍 វិភាគលើលទ្ធផល :**

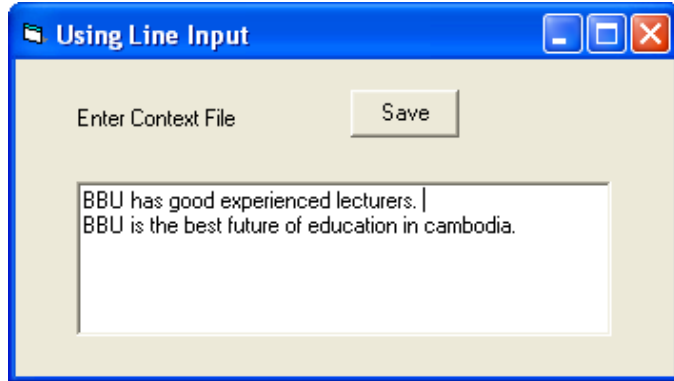
- នៅពេលដែលដំណើរការ Code មកដល់ក្រុង Pirnt #1, Me.txtContext នោះ Print Statement ចាប់ផ្តើមទាញយកទិន្នន័យដែលមាននៅក្នុង txtContext TextBox នៅបញ្ចូលក្នុង File Test.txt ដែលមានទំរង់ដូចរូបទី៧.១០។

- នៅពេលដែលលោកអ្នកវាយបញ្ចូលទិន្នន័យផ្សេងទៀតទៅក្នុង txtContext TextBox (ដូចរូបទី៧.១១) ហើយលោកអ្នកចុច Button Save ម្តងទៀតពេល នោះទិន្នន័យថ្មី ដែលមាននៅក្នុង txtContext TextBox នឹងចូលទៅជំនួសទិន្នន័យចាស់ ដែលមាននៅក្នុង File Test.txt ។ បញ្ហាដែលទិន្នន័យថ្មីទៅជំនួសទិន្នន័យចាស់នេះ បណ្តាលមកពីលោកអ្នក បើក File Test.txt ក្នុងគោលបំណង ដើម្បីបញ្ចូលទិន្នន័យ ( For OutPut) ពេលនោះ Pointer ត្រូវបានកំណត់នៅទីតាំងដែលជាចំណុចចាប់ផ្តើមនៃ File (Begin Of File <<BOF>>) ប្រសិនបើលើលោកអ្នកចង់កុំអោយបាត់បង់ទិន្នន័យចាស់ លោកអ្នកត្រូវ បើក File ក្នុងទំរង់បញ្ចូលទិន្នន័យបន្ថែម (For Append) វិញព្រោះ ពេលលោកអ្នកបើក File ក្នុងទំរង់បញ្ចូលទិន្នន័យបន្ថែម នោះ Pointer បានទៅកំណត់ការចាប់ផ្តើមបញ្ចូល ទិន្នន័យពីចំណុច ចុងក្រោយនៃ File (End Of File <<EOF>>) វិញ។

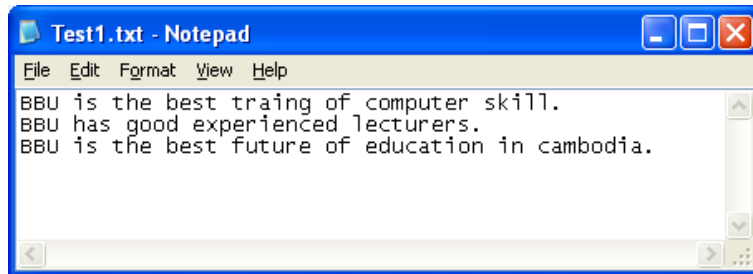
**រូបភាពទី៧.១០**



## រូបភាពទី១១



រូបភាពនៃការផ្លាស់ប្តូរទិន្នន័យក្នុង Test1.txt File



### ឧទាហរណ៍ទី២ :

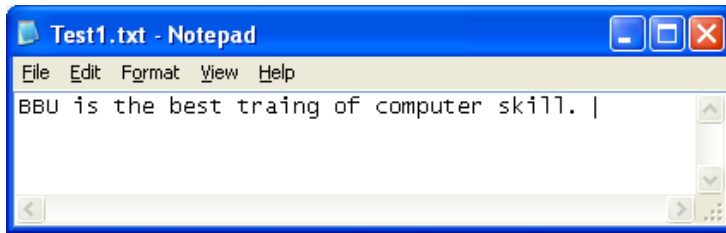
ឧបមាថាលោកអ្នកមាន Form ដូចរូបភាពទី៧.៩

#### ☞ Source Code:

```
Private Sub CmdSave_Click()  
    Open "C:\Test1.txt" For Append As #1  
    Print #1, Me.txtContext  
    Close #1  
End Sub
```

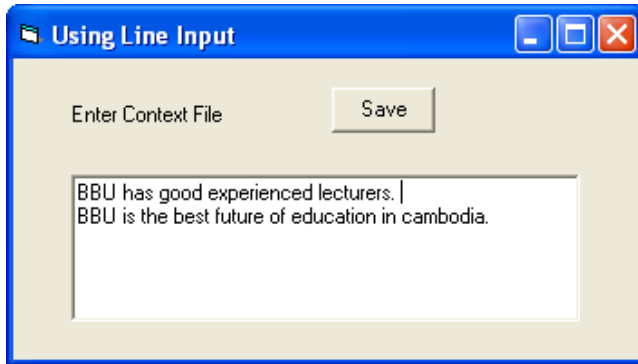
#### ✎ វិភាគលើលទ្ធផល :

- លទ្ធផលនៃការចុច Button Save លើកទីមួយ

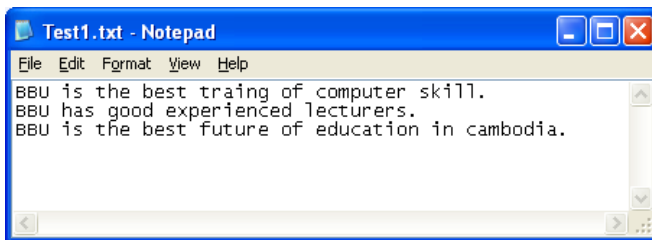


- ឧបមាថាលោកអ្នកធ្វើការបញ្ចូលទិន្នន័យថ្មីទៅក្នុង txtContext TextBox ដូចរូបទី៧.១១ បន្ទាប់មកចុចទៅលើ Button Save ម្តងទៀត ពេលនោះលោកអ្នក នឹងបានឃើញថាមិនមានការបាត់បង់ទិន្នន័យចាស់ដែលមាននៅក្នុង Test1.txt File នោះទេ ព្រោះលោកអ្នកបានបើក Test1.txt File ក្នុងគោលបំណងដើម្បី បញ្ចូលទិន្នន័យបន្ថែម។ ទិន្នន័យដែលមាននៅក្នុង Test1.txt គឺដូចរូបទី៧.១២។

**រូបភាពទី៧.១១**



**រូបភាពទី ៧.១២**



### I.3.2. ការធ្វើ Write Statement

Write Statement ប្រើសំរាប់បញ្ចូលទិន្នន័យទៅអោយ File ទៅតាមប្រភេទនៃ Variable ។

Syntax ទី១:

Write #FNumber , Variable

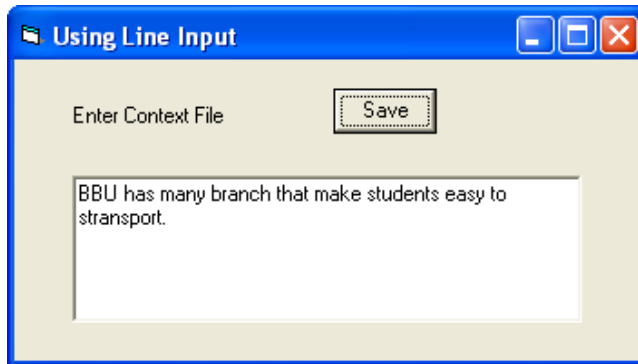
Syntax ទី២:

Write #FNumber , Variable1, Variable2, Variable3, . .

**ឧទាហរណ៍ទី១ :**

ឧបមាថាលោកអ្នកមាន Form ដូចរូបទី៧.១៣

**រូបភាពទី១៣**



☞ Source Code:

```

Private Sub CmdSave_Click()
    Open "C:\Test3.txt" For Append As #1
    Write #1, Me.txtContext
    Close #1
End Sub

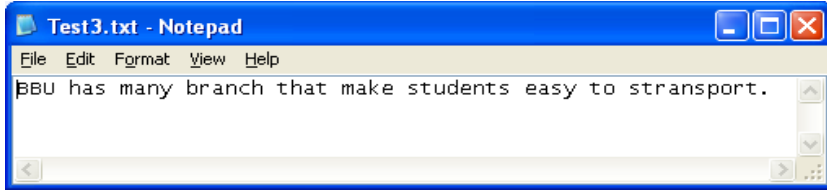
```

**វិភាគលើលទ្ធផល :**

នៅពេលដែលលោកអ្នកប្រើ Write Statement ដើម្បីបញ្ចូលទិន្នន័យក្នុង Test3.txt File Write Statement មានលក្ខណៈពិសេសគឺនៅពេលដែល ទិន្នន័យធ្លាក់ចូលក្នុង File គឺវាកំណត់ប្រភេទទិន្នន័យច្បាស់លាស់។ ជាក់ស្តែងនៅក្នុង ឧទាហរណ៍ទី១ បន្ទាប់ពីលោកអ្នកចុច Button Save ពេលនោះទិន្នន័យ ទាំងអស់ដែលមាននៅក្នុង txtContext Text Box ដែលមានប្រភេទទិន្នន័យជា

String ត្រូវបានរក្សាទុកក្នុង Test3.txt File ហើយនៅក្នុងចន្លោះសញ្ញាចេញកណ្តុរ ( “ “ ) ដូចរូបទី៧.១៤។

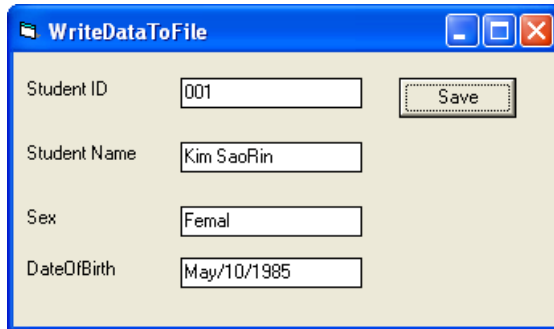
**រូបភាពទី៧.១៤**



**ឧទាហរណ៍ទី២ :**

ឧបមាថាលោកអ្នកមាន Form ដូចទំរង់ដូចរូបភាពទី៧.១៥ ។

**រូបភាពទី៧.១៥**



☞ Source Code:

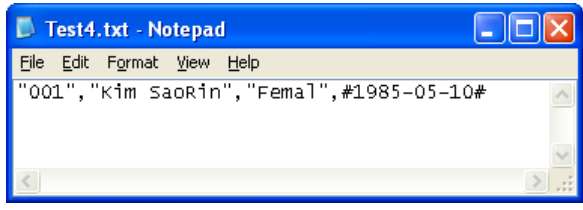
```

Private Sub CmdSave_Click()
    Open "C:\Test4.txt" For Append As #1
    Write #1, Me.txtStudentID.Text, Me.txtStudentName.Text,
    Me.txtSex.Text , CDate(Me.txtDateOfBirth)
    Close #1
End Sub
    
```

**វិភាគលើលទ្ធផល :**

ចំពោះការប្រើ Syntax ទី២ របស់ Write Statement នេះមានន័យថាលោកអ្នកចង់ បញ្ជូនទិន្នន័យជាទំរង់ Record ទៅកាន់ File ណាមួយ ជាក់ស្តែងដូចជានៅក្នុង ឧទាហរណ៍ទី២នេះ លោកអ្នកបានបញ្ជូនទិន្នន័យទិន្នន័យចំនួន៤ Fields ទៅជាក់ក្នុង Test4.txt File ។ នៅពេលដែលទិន្នន័យធ្លាក់ទៅដល់ក្នុង Test4.txt File នោះពី Field មួយទៅ Field មួយគឺវាខ័ណ្ឌគ្នាដោយសញ្ញា Comma (,) ដែលលទ្ធផល របស់វាគឺដូចរូបទី៧.១៦។

**រូបភាពទី៧.១៦**



### I.4. របៀបអានទិន្នន័យចេញពី Sequential File

ដើម្បីទាញទិន្នន័យចេញពី File លោកអ្នកត្រូវប្រើ Statement ពីរគឺ Line Input ឬ Input ។

#### I.4.1. ការប្រើ Line Input Statement

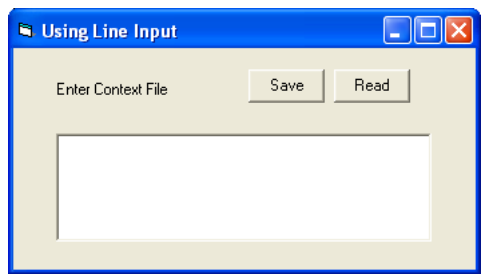
ប្រើសំរាប់ទាញទិន្នន័យម្តងមួយបន្ទាត់ចេញពី File ហើយគេច្រើនប្រើវាជាមួយ ការបញ្ជូនទិន្នន័យទៅអោយ File ដោយប្រើ Print Statement ។  
Syntax:

```
Line Input #FNumber, Variable
```

#### ឧទាហរណ៍ទី១ :

ឧបមាថាលោកអ្នកមាន Form ដូចរូបទី៧.១៧ ។

**រូបភាពទី៧.១៧**



☞ Source Code:

```

Private Sub CmdSave_Click()
    Open "C:\Test1.txt" For Append As #1
    Print #1, Me.Text1
    Close #1
End Sub

```

```

Private Sub CmdRead_Click()
    Dim St$
    Open "C:\Test1.txt" For Input As #1
    Line Input #1, St
    MsgBox St
    Close #1
End Sub

```

✎ **វិភាគលើលទ្ធផល :**

នៅពេលដែលលោកអ្នកចុចទៅលើ Button Read ពេលនោះទិន្នន័យ ដែលបានចេញមកគឺបានតែមួយបន្ទាត់ប៉ុណ្ណោះគឺដូចទៅនឹងរូបភាពទី៧.១៨។ ប្រសិនបើ លោកអ្នកចង់បានទិន្នន័យទាំងអស់ដែលមាននៅក្នុង File លោកអ្នកត្រូវប្រើ អនុគមន៍មួយឈ្មោះថា EOF (End Of File) ដើម្បីត្រួតពិនិត្យមើលថាទិន្នន័យដែល លោកអ្នកអានចេញពី File នោះថា តើដល់ទីបញ្ចប់នៃ File ហើយឬនៅ។

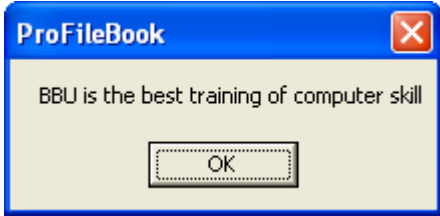
✎ **ចំណាំ:**

Syntax:

```
EOF(FileNumber As Integer) As Boolean
```

(សូមមើលចំណុចទី I.5. ការប្រើអនុគមន៍មួយចំនួនដែលទាក់ទងនឹង Files)

**រូបភាពទី៧.១៨**





## ឧទាហរណ៍ទី២ :

ឧបមាថាលោកអ្នកមាន Form ដូចឧទាហរណ៍ទី១។

☞ Source Code:

```

Private Sub CmdSave_Click()
    Open "C:\Test1.txt" For Append As #1
    Print #1, Me.Text1
    Close #1
End Sub

```

```

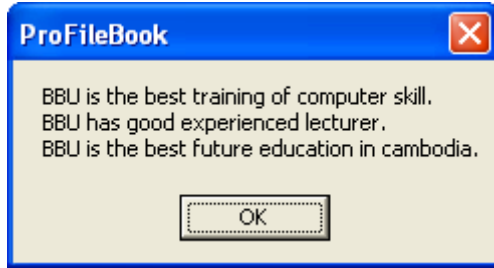
Private Sub CmdRead_Click()
    Dim St$
    Open "C:\Test1.txt" For Input As #1
    Do While EOF(1) = False
        Line Input #1, St
        Data = Data & St & vbCrLf
    Loop
    Close #1
    MsgBox Data
End Sub

```

## 🔗 វិភាគលើលទ្ធផល :

នៅក្នុងឧទាហរណ៍ទី២ នេះមានសរសេរ Code ដោយប្រើ Do While EOF(1) = False ដែលបញ្ជាក់ថា តើការអានទិន្នន័យចេញពី Test1.txt File នោះដល់ទីបញ្ចប់នៃ File ហើយឬនៅ ប្រសិនបើមិនទាន់ដល់ទីបញ្ចប់នៃ File ទេនោះ វាត្រូវបានអាន ជាបន្តបន្ទាប់ ម្តងមួយបន្តាត់ទៅដាក់ចូលក្នុងអថេរ St ជាបណ្តោះអាសន្ន ហើយបញ្ជូនពី St ទៅជួរ ជាលទ្ធផលជាបន្តបន្ទាប់នៅក្នុងអថេរ Data រហូតដល់វាជួបទីបញ្ចប់នៃ File (EOF) ទើបវាចាកចេញពី Loop ហើយបោះលទ្ធផលដែលទទួលបានដែលដូចគ្នាទៅនឹង រូបភាពទី៧.១៩។

## រូបភាពទី៧.១៩



### I.4.2. ការប្រើ Input Statement

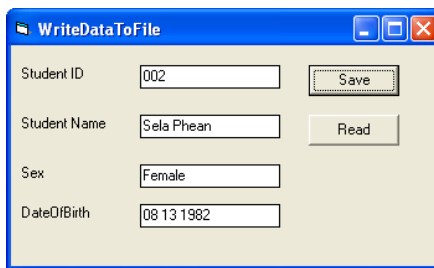
ជាទូទៅគេប្រើ Input Statement ជាមួយនិងការបញ្ជូនទិន្នន័យទៅអោយ File ដោយប្រើ Write Statement ព្រោះការប្រើ Write លោកអ្នកអាចបញ្ជូនទិន្នន័យ ជាទំរង់ Record ហើយទិន្នន័យដែលធ្លាក់ចូល File គឺមាននិមិត្តសញ្ញានៅអមសងខាង ដែលបញ្ជាក់ពីប្រភេទ Data Type នៃទិន្នន័យ ដូច្នេះពេលដែលអានទិន្នន័យចេញវិញ គឺទាមទារអោយលោកអ្នកអានទិន្នន័យជាទំរង់ Record វិញដែរ បើមិនដូច្នោះទេ ទិន្នន័យដែលទទួលបាន នឹងមកទាំងនិមិត្តសញ្ញាមកជាមួយ។

Syntax ទី១ :

Input #FNumber, Variable1, Variable2, . .

**ឧទាហរណ៍ទី១ :**

ឧបមាថាលោកអ្នកមានទំរង់ Form ដូចខាងក្រោម :



☞ Source Code:

```
Private Sub CmdSave_Click()
    Open "C:\Test4.txt" For Append As #1
    Write #1, Me.txtStudentID.Text, Me.txtStudentName.Text _
        , Me.txtSex.Text, CDate(Me.txtDateOfBirth.Text)
    Close #1
    Call Clear(Me)
End Sub
```

End Sub

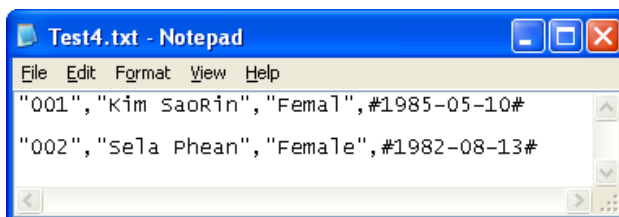
```
Private Sub CmdRead_Click()  
    Dim StdID, StdName, StdSex As String  
    Dim StdDB As Date  
    Dim Data As String  
    Data = "StudentID" & Space(5) & "StudentName" & Space(5) _  
        & "Sex" & Space(5) & "DateOfBirth" & vbCrLf  
    Open "C:\Test4.txt" For Input As #1  
    Do While Not EOF(1)  
        Input #1, StdID, StdName, StdSex, StdDB  
        Data = Data & StdID & Space(16) & StdName & Space(8) & _  
            StdSex & Space(3) & Format(StdDB, "mmm/dd/yyyy") & vbCrLf  
    Loop  
    MsgBox Data  
    Close #1  
End Sub
```

```
Sub Clear(FrmName As Form)  
    Dim Ctr As Control  
    For Each Ctr In FrmName.Controls  
        If TypeOf Ctr Is TextBox Then  
            Ctr.Text = ""  
        End If  
    Next  
End Sub
```

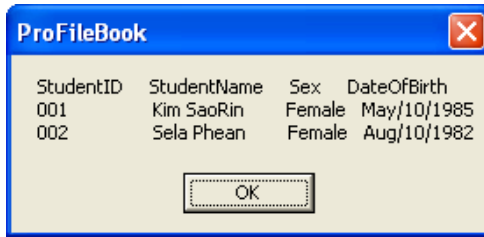
🗑️ **លទ្ធផលដែលទទួលបាន :**

បន្ទាប់ពីលោកអ្នកចុចទៅលើ Save Button ពេលនោះ Record ថ្មីនឹងធ្លាក់ចូល បន្ថែមទៀតទៅក្នុង File Test4.txt ដូចរូបទី៧.២០ រួចហើយពេលដែលលោកអ្នក ចុចទៅលើ Button Read ពេលនោះទិន្នន័យនឹងបង្ហាញដូចរូបភាពទី៧.២១។

**រូបភាពទី៧.២០**



## រូបភាពទី៧.២១



### ឧទាហរណ៍ទី២ :

ឧបមាថាឧទាហរណ៍ទី២នេះដូចគ្នាទៅនឹងឧទាហរណ៍ទី១តែខុសគ្នាត្រង់ ការអាន ទិន្នន័យចេញពី File វិញ គឺនៅក្នុងនៅក្នុងឧទាហរណ៍ទី២ នេះធ្វើការអាន ទិន្នន័យចេញពី File Test4.txt ដោយប្រើ Line Input Statement វិញ។

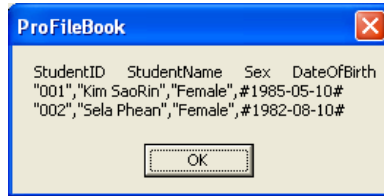
☞ Source Code:

```
.....  
Private Sub CmdRead_Click()  
    Dim StdID, StdName, StdSex As String  
    Dim StdDB As Date  
    Dim Data, Temp As String  
    Data = "StudentID" & Space(5) & "StudentName" & _  
        Space(5) & "Sex" & Space(5) & "DateOfBirth" & vbCrLf  
    Open "C:\Test4.txt" For Input As #1  
    Do While Not EOF(1)  
        Line Input #1, Temp  
        Data = Data & Temp & vbCrLf  
    Loop  
    Close #1  
    MsgBox Data  
    Call Clear(Me)  
End Sub  
.....
```

### 🗑 វិភាគលើលទ្ធផល :

យើងឃើញថាការប្រើ Line Input Statement ដើម្បីអានទិន្នន័យចេញពី File ដែលបញ្ចូលទិន្នន័យដោយប្រើ Write Statement ជាទំរង់ Record វានឹងអោយលទ្ធផល ដូចរូបទី៧.២២ ដែលមានភ្ជាប់មកជាមួយនូវនិមិត្តសញ្ញាដែលនៅអមសងខាងទិន្នន័យ។

**រូបភាពទី៧.២២**



**Syntax ទី២ :**

**Variable = Input( LenghtOfChr As long, #FNumber)**

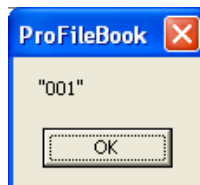
ចំពោះ Syntax ទី២ នេះ Input Statement ដើរតួនាទីជា Function ដែល Return តំលៃជា String ពី #FNumber ទៅតាម LenghtOfChr Argument ដែលលោកអ្នក បោះតំលៃអោយ។

**ឧទាហរណ៍ :**

☞ Source Code:

```
Private Sub Form_Load()
    Dim St As String
    Open "C:\Test4.txt" For Input As #1
    St = Input(5, #1)
    MsgBox St
    Close #1
End Sub
```

**លទ្ធផល :**



## I.5.ការប្រើប្រាស់អនុគមន៍ និង Statement មួយចំនួនដែលទាក់ទងនឹង Sequential File

### I.5.1.ការប្រើប្រាស់អនុគមន៍ EOF (End Of File)

Syntax:

```
EOF(FileNumber As Integer) As Boolean
```

អនុគមន៍នេះត្រូវបានប្រើសំរាប់ត្រួតពិនិត្យថាតើ File ដែលលោកអ្នកអាននោះ ដល់ទីបញ្ចប់នៃ File (End Of File ) ហើយឬនៅ។ ប្រសិនបើអនុគមន៍ទទួលបានតំលៃ True បន្ទាប់ពីការហៅ មានន័យថា File ដែលត្រូវបានអាននោះដល់ទីបញ្ចប់នៃ File ហើយ ផ្ទុយទៅវិញប្រសិនបើអនុគមន៍ទទួលបានតំលៃ False បន្ទាប់ពីការហៅមានន័យថា File ដែលបានអាននោះមិនទាន់ដល់ទីបញ្ចប់នៃ File ឡើយ។

### I.5.2.ការប្រើប្រាស់អនុគមន៍ FileLen (File Length)

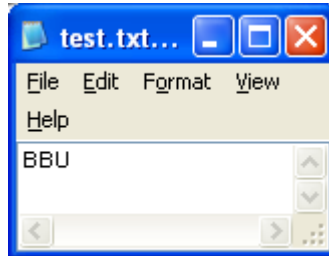
Syntax:

```
FileLen(PathName As String) As Long
```

ចំពោះអនុគមន៍នេះត្រូវបានប្រើដើម្បីទទួលយកទំហំរបស់ File ដោយគិតជា Bytes (Characters) ។ លោកអ្នកអាចប្រើអនុគមន៍នេះបានដោយមិនចាំបាច់បើក File ជាមុនឡើយ។

**ឧទាហរណ៍ទី១ :**

ឧបមាថាលោកអ្នកមានទិន្នន័យក្នុង File មួយឈ្មោះថា Test.txt ដូចរូបខាងក្រោម :



☞ Source Code:

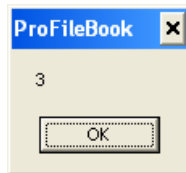
```
Private Sub Form_Load()  
    MsgBox FileLen("C:\test.txt")
```

End Sub

 **វិភាគលទ្ធផល :**

លទ្ធផលនឹងទទួលបានចំនួនតួអក្សរទាំងអស់ដែលមាននៅក្នុង File Test.txt ។ ដោយនៅ ក្នុង Test.txt File មានតួអក្សរចំនួនបីតួ ដូច្នេះលទ្ធផលដែលទទួលបានគឺ លេខ បីដែលដូចរូបទី៧.២៣។

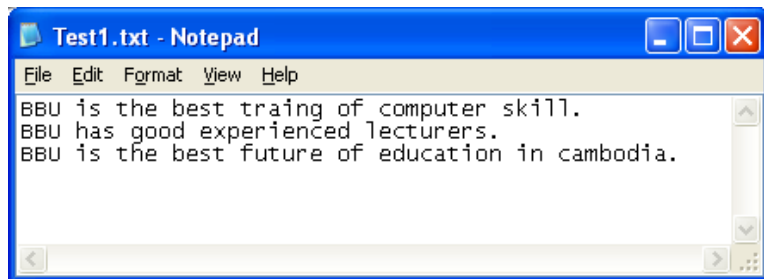
**រូបភាពទី២៣**



**ឧទាហរណ៍ទី២ :**

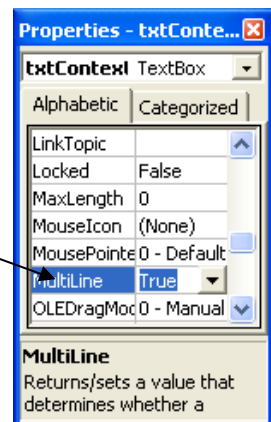
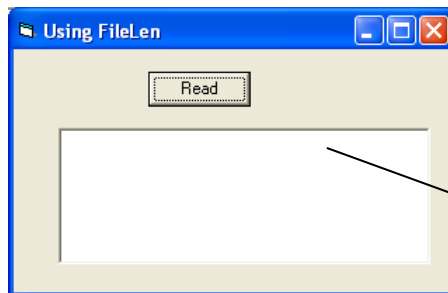
ឧបមាថាលោកអ្នកមានទិន្នន័យក្នុង File មួយឈ្មោះថា Test1.txt ដូចរូបខាង ក្រោម:

**រូបភាពទី៧.២៣**



ហើយលោកអ្នកមានទម្រង់ Form ដូចខាងក្រោម :

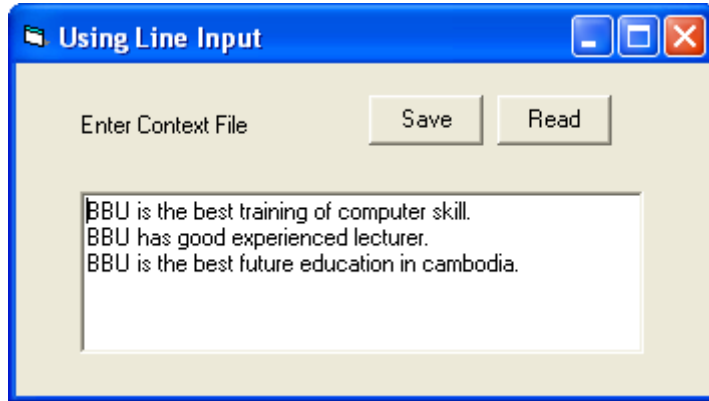
**រូបភាពទី៧.២៤**



☞ Source Code:

```
Private Sub CmdRead_Click()  
    Open "C:\test1.txt" For Input As #1  
    Me.txtContext = Input(FileLen("c:\test1.txt"), #1)  
    Close #1  
End Sub
```

លទ្ធផល :



### 1.5.3. ការប្រើប្រាស់អនុគមន៍ LOF (Length Of File)

Syntax:

LOF(FileNumber As Integer) As Long

អនុគមន៍នេះត្រូវបានប្រើក្នុងគោលបំណងដូចគ្នាទៅនឹងអនុគមន៍ FileLen ដែរ តែវាខុសគ្នាត្រង់  
Argument ដែលត្រូវបោះទៅអោយអនុគមន៍ LOF ជា File Number  
ហើយអនុគមន៍នេះអាចប្រើបានលុះត្រាតែលោកអ្នកបើក File ជាមុនសិន។

ឧទាហរណ៍ទី១ :

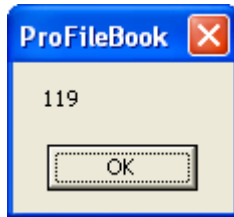
ឧបមាថា លោកអ្នកមានទិន្នន័យក្នុង File ដូចរូបភាពទី៧.២៣។

☞ Source Code:



```
Private Sub Form_Load()
    Open "C:\test1.txt" For Input As #1
    MsgBox LOF(1)
    Close #1
End Sub
```

**លទ្ធផល :**



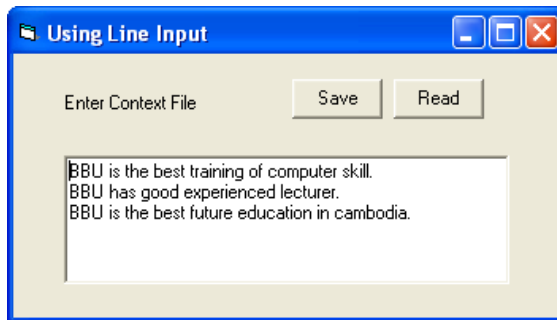
**ឧទាហរណ៍ទី២:**

ឧបមាថា លោកអ្នកមាន Form ដូចរូបភាពទី ៧.24 ។

☞ Source Code:

```
Private Sub Form_Load()
    Open "C:\test1.txt" For Input As #1
    Me.txtContext = Input(LOF(1), #1)
    Close #1
End Sub
```

**លទ្ធផល :**



## 1.5.4. ការប្រើប្រាស់ Seek Statement

Syntax:

```
Seek #FileNumber, PointerPosition as Long
```

Seek វាមិនមែនជាអនុគមន៍ឡើយវាជា Statement មួយដែលមានតួនាទី សំរាប់កំណត់ទីតាំង Pointer អោយចង្អុលទៅកាន់ទីតាំង Byte (Character) ណាមួយ ដែលមាននៅក្នុង File ។

**ឧទាហរណ៍ :**

ឧបមាថា លោកអ្នកមានទម្រង់ Form ដូចរូបភាពទី ៧.២៤។

☞ Source Code:

```
.....
Private Sub CmdRead_Click()
  Open "C:\test1.txt" For Input As #1
  Seek #1, 25
  Do While Not EOF(1)
    Input #1, St$
    Me.txtContext = Me.txtContext & St & vbCrLf
  Loop
  Close #1
End Sub
.....
```

### 1.5.5.ការប្រើប្រាស់អនុគមន៍ Dir (Directory)

Syntax:

```
Dir([PathName]) As String
```

អនុគមន៍នេះត្រូវបានប្រើដើម្បីត្រួតពិនិត្យមើល File ណាមួយទៅតាម Path ថាតើ File នោះមានដែរឬទេ។ ប្រសិនបើ File ដែលលោកអ្នកពិនិត្យនោះមាន អនុគមន៍ និងទទួលបានតំលៃជា ឈ្មោះរបស់ File ហើយនឹង Extension ផ្ទុយទៅវិញ ប្រសិនបើអនុគមន៍ត្រួតពិនិត្យមើលមិនឃើញ File នោះទេ អនុគមន៍នឹងទទួលបានតំលៃ Zero Length ("") ។

**ឧទាហរណ៍ទី១ :**

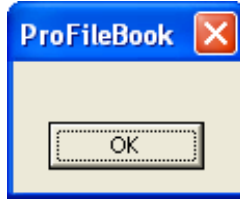
ឧបមាថាលោកអ្នកអត់មាន File ឈ្មោះថា BBU.txt នៅក្នុង Drive C:\ ទេ ហើយលោកអ្នកប្រើអនុគមន៍ Dir ដើម្បីត្រួតពិនិត្យមើល។

☞ Source Code:

```
.....
Private Sub Form_Load()
.....
```

```
MsgBox Dir("C:\BBU.txt")  
End Sub
```

លទ្ធផល :



ឧទាហរណ៍ទី២ :

☞ Source Code:

```
Private Sub Form_Load()  
MsgBox Dir("C:\test1.txt"), , "Using Dir Function"  
End Sub
```

លទ្ធផល :



ឧទាហរណ៍ទី៣ :

☞ Source Code:

```
Private Sub CmdRead_Click()  
If Dir("C:\test1.txt") <> "" Then  
Open "C:\test1.txt" For Input As #1  
Seek #1, 25  
Do While Not EOF(1)  
Input #1, St$  
Me.txtContext = Me.txtContext & St & vbCrLf  
Loop  
Close #1  
Else  
MsgBox "C:\Test1.txt Not Found", vbInformation  
End If  
End Sub
```

## 1.5.6. ការប្រើប្រាស់អនុគមន៍ FileDateTime (Directory)

Syntax:

```
FileDateTime(PathName As String)
```

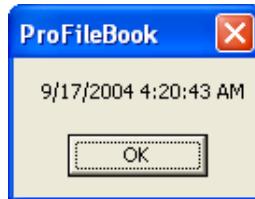
អនុគមន៍នេះត្រូវបានប្រើសំរាប់ទទួលយកថ្ងៃខែឆ្នាំព្រមទាំងម៉ោងដែលលោកអ្នក បានបង្កើត File ឬ Folder ។

**ឧទាហរណ៍ :**

☞ Source Code:

```
Private Sub Form_Load()  
    MsgBox FileDateTime("C:\test1.txt")  
End Sub
```

**លទ្ធផល :**



## 1.5.7. ការប្រើប្រាស់ Kill Statement

Syntax:

```
Kill(PathName)
```

គឺជា Statement មួយដែលប្រើសំរាប់លុប File ទៅតាម Path Argument ចេញពី Partition របស់ Hard Disk ។

**ឧទាហរណ៍ :**

☞ Source Code:

```
Private Sub Form_Load()  
    Kill "C:\Test1.txt"  
End Sub
```

## I.5.8. ការប្រើប្រាស់ Name Statement

Syntax:

```
Name "SourceFileNameAndPath" As  
"NewFileNameAndPath"
```

វាជា Statement មួយដែលប្រើសំរាប់ កែឈ្មោះ File ពី SourceFile ទៅ NewFile ។

**ឧទាហរណ៍ :**

☞ Source Code:

```
Private Sub Form_Load()  
    Name "C:\Test4.txt" As "C:\BBU.txt"  
End Sub
```

## I.5.9. ការប្រើប្រាស់ Mkdir Statement

Syntax:

```
Mkdir(Path As String)
```

វាជា Statement មួយដែលប្រើសំរាប់បង្កើត Folder ឬ Directory ទៅ តាម Path Argument ។

**ឧទាហរណ៍ :**

☞ Source Code:

```
Private Sub Form_Load()  
    Mkdir "C:\ BBU "  
End Sub
```

## I.5.10. ការប្រើប្រាស់ Rmdir Statement

Syntax:

```
Rmdir(Path As String)
```

គឺជា Statement មួយដែលប្រើសំរាប់លុប Folder/ Directory ទៅតាម Path Argument ។

**ឧទាហរណ៍ :**

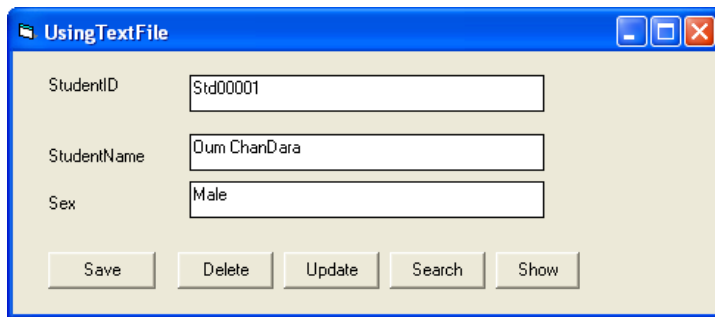
☞ Source Code:

```
Private Sub Form_Load()
  Rmdir ("C:\ BBU")
End Sub Private Sub Form_Load()
```

## II. លំហាត់អនុវត្តលើ Sequential File

### II.1. លំហាត់ទី១

ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចខាងក្រោម :



#### គោលបំណងនៃលំហាត់

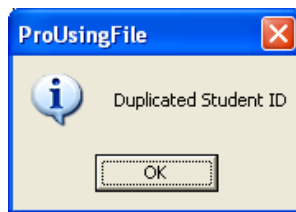
- ពេលដែលលោកអ្នកចុចលើ Button Save ពេលនោះទិន្នន័យនឹងធ្លាក់ចូល ក្នុង File មួយដែលមានឈ្មោះថា Student.txt ប៉ុន្តែមិនអនុញ្ញាតអោយលេខ StudentID ស្ករឡើយ ។ ប្រសិនបើលេខ Code StudentID ស្ករសូមបង្ហាញសារឡើង វិញដូចរូបភាពទី៧.២៥។

- ពេលដែលលោកអ្នកចុចលើ Button Show សូមបង្ហាញទិន្នន័យទាំងអស់របស់ Student.txt File មកដាក់បញ្ចូលក្នុង ListView ខាងក្រោមដូចរូបភាពទី៧.២៦ ។

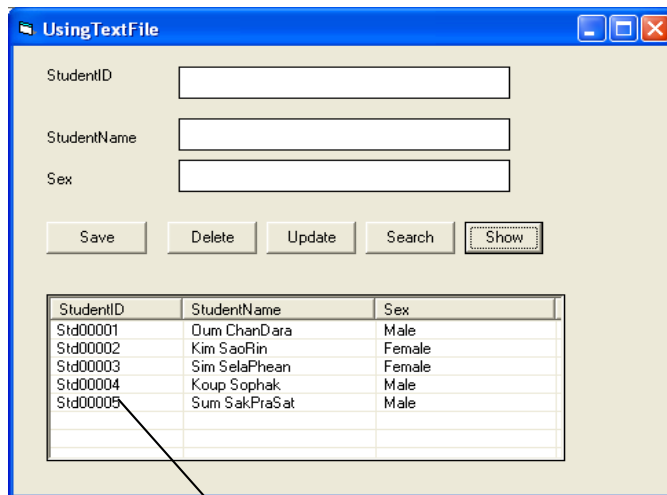
- ពេលដែលលោកអ្នកចុចទៅលើ Button Delete សូមអោយទៅលប់ទិន្នន័យក្នុង Student.txt ម្តងមួយ Record ទៅតាមលេខកូដ StudentID ដែលលោកអ្នកវាយបញ្ចូល តែក្នុងករណីដែលលេខ Code StudentID អត់មានសូមបង្ហាញសារថា “ Sudent ID Not Found”

- ពេលលោកអ្នកចុចទៅលើ Button Update សូមកែ Record ណាមួយទៅតាម StudentID ដែលលោកអ្នកវាយបញ្ចូល ហើយកែ Filed ផ្សេងៗទៀតទៅតាម Text Box នៃ Field ខាងលើដែល លោកអ្នកវាយបញ្ចូល តែក្នុងករណីដែល លេខកូដ StudentID អត់មានសូមបង្ហាញសារថា “ Sudent ID Not Found” ។

**រូបភាពទី៧.២៥**



**រូបភាពទី៧.២៦**



↳ ListView Control

**✍ - ណែនាំ :**

- ◆ សូមមើលរបៀបប្រើ ListView Control នៅក្នុងមេរៀនទី៩ ជាមុនសិន។
- ◆ ដើម្បីលប់ទិន្នន័យរបស់ File មួយ លោកអ្នកត្រូវអនុវត្តតាមជំហានដូច ខាងក្រោម:
  - ស្វែងរក Record ដែលលោកអ្នកចង់លប់ជាមុនសិន។
  - ចំលងយកទិន្នន័យដែលលោកអ្នកមិនចង់លប់ចេញទៅដាក់ក្នុង File ថ្មីមួយ
  - លប់ឈ្មោះ File ដែលលោកអ្នកចង់លប់ទិន្នន័យចោល
  - កែឈ្មោះ: File ថ្មីដែលលោកអ្នកបានចំលង ទិន្នន័យដែលលោកអ្នក

មិនចង់លប់ចូលទៅជាឈ្មោះ File ចាស់វិញ។

- ◆ ដើម្បីកែទិន្នន័យក្នុង File លោកអ្នកត្រូវអនុវត្តតាមជំហានដូចខាងក្រោម :
  - ស្វែងរក Record ដែលលោកអ្នកចង់កែជាមុនសិន។
  - ប្រសិនបើជួបសូមបញ្ចូលទិន្នន័យថ្មីដែលលោកអ្នកចង់កែព្រមទាំងលេខ Code StudentID

ដែលលោកអ្នកស្វែងរកឃើញចូលទៅ File ថ្មីមួយទៀត។

- ប្រសិនបើមិនជួបលោកអ្នកក៏ត្រូវតែបញ្ចូលទិន្នន័យទៅអោយ File ថ្មីដែលប៉ុន្តែ
- ទិន្នន័យដែលបញ្ចូលនេះមិនមែនជាទិន្នន័យថ្មីទេវាគឺជាទិន្នន័យដែលលោកអ្នកបានអានចេញពី File ចាស់។
- លប់ File ចាស់ចោល ។
- កែឈ្មោះ File ថ្មីមកជា File ចាស់វិញ។

## II.2. លំហាត់ទី២

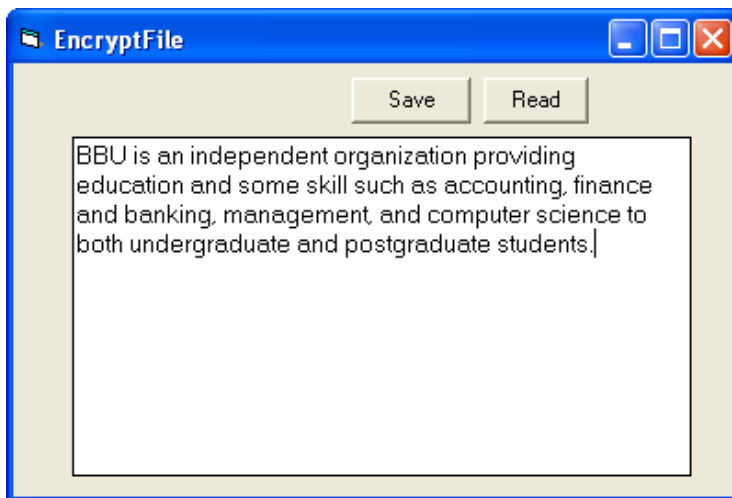
ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចភាពទី៧.២៧ ។

### គោលបំណងនៃលំហាត់

- នៅពេលដែលលោកអ្នកវាយបញ្ចូលអត្ថបទដែលដូចគ្នាក្នុង Text Box រួច សូម ចុចទៅលើ Button Save ពេលនោះទិន្នន័យដែលមាននៅក្នុង Text Box នឹងធ្លាក់ចូលទៅក្នុង File មួយដែលមានឈ្មោះថា BBU.txt ហើយទិន្នន័យដែលមាននៅក្នុង File នោះ លោកអ្នកមិនអាចយល់ពីអត្ថន័យរបស់វាឡើយ (Encrypt Data) គឺមានទម្រង់ដូចរូបទី៧.២៨។

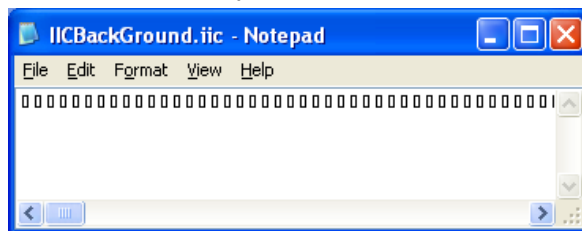


## រូបភាពទី៧. ២៧



- ពេលដែលលោកអ្នកចុចទៅលើ Button Read ចាប់ផ្តើមធ្វើការអានទិន្នន័យ ដែលមើលមិនយល់ ដែលមាននៅក្នុង BBU.txt មកដាក់លើ Text Box វិញដែលអាច អោយលោកអ្នកអានទិន្នន័យនោះយល់ (Decrypt Data ) ដែលដូចទៅនឹង រូបភាពទី៧ . ២៧។

## រូបភាពទី២៨



 ណែនាំ :

### ◆ របៀប Encrypt Data

- លោកអ្នកត្រូវកាត់យកតួអក្សរម្តងមួយ ហើយប្តូរជា Ascii code រួចបូកបន្ថែម លេខណាមួយ ដើម្បីប្តូរទៅជាតួអក្សរមួយផ្សេងទៀត។ លោកអ្នកត្រូវ ធ្វើម្តងមួយតួអក្សរ រហូតដល់អស់តួអក្សរ ដែលមាននៅក្នុងអត្ថបទរបស់លោកអ្នក ហើយទើបបញ្ជូនតួអក្សរថ្មី ទាំងនោះ ទៅអោយអថេរមួយ។

- លោកអ្នកត្រូវប្រើ អនុគមន៍មួយឈ្មោះថា StrReverse ដើម្បីត្រលប់តួអក្សរ ទាំងនោះវិញ។
- ដើម្បីអោយកាន់តែស្មុគស្មាញទៀតនោះលោកអ្នកគួរតែ យកតួអក្សរដែលត្រលប់

ទីតាំងទាំងនោះមកកាត់ម្តងមួយ រហូតដល់អស់ដោយមានភ្ជាប់ជាមួយនិមិត្តសញ្ញាអ្វីមួយ នៅពេលកាត់ផង។

◆ **របៀប Decrypt Data**

- ស្វែងរកតួអក្សរដែលខុសពីនិមិត្តសញ្ញាដែលលោកអ្នកបានបន្ថែមចូល
- ត្រលប់តួអក្សរអោយដូចដើមវិញ
- ត្រូវដកលេខ Ascii Code នៃតួអក្សរនីមួយៗវិញដើម្បីអោយតួអក្សរនីមួយៗ ដូចតួអក្សរដើម។

### II.3. លំហាត់ទី៣

សូមលោកអ្នកទៅធ្វើលើលំហាត់លេខ១ ឡើងវិញដោយពេលបញ្ចូលទិន្នន័យជា Record នោះសូមអោយ Encrypt Data ហើយនៅពេលដែលអានទិន្នន័យពី File មកវិញសូម Decrypt Data ។

### 🔗 កំណែលើលំហាត់ទី១ (II.1. លំហាត់ទី១)

👉 Source Code:

```
Private Sub Form_Load( )
    AddColumHader Me.ListView1, "StudentID", "StudentName", "Sex"
    Me.Width = 7305
    Me.Height = 3180
End Sub
```

```
Private Sub CmdSave_Click()
    Open App.Path & "\" & "Student.txt" For Append As #1
    Close #1
    If CheckField(App.Path & "\student.txt", Me.txtStdID) = False Then
        Call WriteRecordToFile("Student.txt", Me.txtStdID, Me.txtName,
                               Me.txtSex)
        Call Clear(Me)
    Else
        MsgBox "Duplicated Student ID", vbInformation
    End If
Exit Sub
End Sub
```

---

```
Private Sub CmdDelete_Click()
    If CheckField(App.Path & "\Student.txt", Me.txtStdID) = True Then
        Call DeleteRecord("Student.txt", "Temp.txt")
    Else
        MsgBox "Student ID not found", vbInformation
    End If
End Sub
```

---

```
Private Sub CmdUpdate_Click()
    Call UpdateRecord("Student.txt", "Temp.txt", Me.txtStdID, Me.txtName,
                    Me.txtSex)
End Sub
```

---

```
Private Sub CmdSearch_Click()
    Dim Search$
    Search = InputBox("Please, enter student name that you want to search",
                    "Search Student")
    If Search = "" Then
        MsgBox "You must enter Name!", vbCritical
        Exit Sub
    Else
        Me.ListView1.ListItems.Clear
        Open App.Path & "\ & "Student.txt" For Input As #1
        Do Until EOF(1)
            Input #1, StdID$, stdName$, StdSex$
            If UCase(stdName) = UCase(Trim(Search)) Then
                Call AddDataToLvw(Me.ListView1, StdID, stdName, StdSex)
            End If
        Loop
        Close #1
    End If
End Sub
```

---

```
Private Sub CmdShow_Click()
    Call RedRecordFromFile(App.Path & "\Student.txt", Me.ListView1)
    Me.Width = 7305
    Me.Height = 5715
End Sub
```

---

```
Sub WriteRecordToFile(fileName, id As String, name As String, sex As String)
    Open App.Path & "\ & fileName For Append As #1
```

```
    Write #1, id, name, sex
  Close #1
End Sub
```

---

```
Sub RedRecordFromFile(fileName, Lvw As ListView)
  Lvw.ListItems.Clear
  Open fileName For Input As #1
  Do While Not EOF(1)
    Input #1, StdID$, stdName$, StdSex$
    AddDataToLvw Lvw, StdID, stdName, StdSex
  Loop
  Close #1
End Sub
```

---

```
Sub UpdateRecord(FName As String, TempFname As String, txtStdID$,
                txtstdName$, txtStdSex$)
  Open App.Path & "\" & FName For Input As #1
  Open App.Path & "\" & TempFname For Append As #2
  Do While Not EOF(1) = True
    Input #1, StdID$, stdName$, StdSex$

    If StrComp(Trim(StdID), Trim(Me.txtStdID), vbTextCompare) = 0 Then
      Write #2, txtStdID, txtstdName, txtStdSex
    Else
      Write #2, StdID, stdName, StdSex
    End If
  Loop
  Close #2, 1
  Kill App.Path & "\" & FName
  Name App.Path & "\" & TempFname As App.Path & "\" & FName
End Sub
```

---

```
Sub DeleteRecord(FName As String, TempFname As String)
  Open App.Path & "\" & FName For Input As #1
  Open App.Path & "\" & TempFname For Append As #2
  Do While Not EOF(1)
    Input #1, StdID$, stdName$, StdSex$
    If StrComp(Trim(StdID), Trim(Me.txtStdID), vbTextCompare) <> 0
      Then
        Write #2, StdID, stdName, StdSex
      End If
  Loop
  Close #1, 2
```

```
Kill App.Path & "\" & FName
Name App.Path & "\" & TempFname As App.Path & "\" & FName
```

```
End Sub
```

---

```
Function CheckField(ByVal FileName As String, FieldName As String) As
```

```
    Boolean
```

```
    Open FileName For Input As #1
```

```
    Do While Not EOF(1)
```

```
        Input #1, StdID$, stdName$, StdSex$
```

```
        If StrComp(Trim(StdID), Trim(FieldName), vbTextCompare) = 0 Then
```

```
            CheckField = True
```

```
            Close #1
```

```
            Exit Function
```

```
        End If
```

```
    Loop
```

```
    Close #1
```

```
End Function
```

---

```
Sub Clear(frm As Form)
```

```
    Dim ctr As Control
```

```
    For Each ctr In frm.Controls
```

```
        If TypeOf ctr Is TextBox Then
```

```
            ctr.Text = ""
```

```
        End If
```

```
    Next
```

```
End Sub
```

---

```
Sub AddColumnHader(Lvw As ListView, ParamArray Col())
```

```
    For i = 0 To UBound(Col)
```

```
        Lvw.ColumnHeaders.Add , , Col(i)
```

```
    Next
```

```
    Lvw.View = lvwReport
```

```
    Lvw.FullRowSelect = True
```

```
    Lvw.GridLines = True
```

```
End Sub
```

---

```
Sub AddDataToLvw(Lvw As ListView, ParamArray Data())
```

```
    Dim Litem As ListItem
```

```
    Set Litem = Lvw.ListItems.Add(, , Data(0))
```


```
    For i = 1 To UBound(Data)
```

```
        Litem.SubItems(i) = Data(i)
```

```
Next
End Sub
```

---

## កំណែស៊ីមប្រាក់ទី២ (II.2.ស៊ីមប្រាក់ទី២)

 Source Code:

---

```
Private Sub CmdRead_Click()
    Me.txtContext = ReadContextFile("BBUBackGround.bbu ")
End Sub
```

---

```
Private Sub CmdSave_Click()
    Call WriteContextToFile("BBUBackGround.bbu", Me.txtContext.Text)
End Sub
```

---

```
Function Encrypt(ByVal sText As String) As String
    Dim i As Integer
    Dim St, St1 As String
    For i = 1 To Len(sText)
        St = St & Chr(Asc(Mid(sText, i, 1)) + 10)
    Next
    St = StrReverse(St)
    For i = 1 To Len(St)
        St1 = St1 & Mid(St, i, 1) & ":"
    Next
    Encrypt = St1
End Function
```

---

```
Function Decrypt(ByVal sText As String) As String
    Dim i As Integer
    Dim St, St1 As String
    For i = 1 To Len(sText)
        If Mid(sText, i, 1) <> ":" Then
            St = St & Mid(sText, i, 1)
        End If
    Next
    St = StrReverse(St)
    For i = 1 To Len(St)
        St1 = St1 & Chr(Asc(Mid(St, i, 1)) - 10)
    Next
```

```
Decrypt = St1
End Function
```

```
Sub WriteContextToFile(Fname As String, Context As String)
  Open App.Path & "\" & Fname For Append As #1
  Print #1, Encrypt(Context)
  Close #1
End Sub
```

```
Function ReadContextFile$(Fname As String)
  Dim St, St1 As String
  Open App.Path & "\" & Fname For Input As #1
  Do Until EOF(1)
    Input #1, St
    St1 = St1 & St
  Loop
  Close #1
  ReadContextFile = Decrypt(St1)
End Function
```

## 🗑️ កំណែលើបំហាត់ទី៣ (II.2.បំហាត់ទី២)

សូមលោកអ្នករៀនធ្វើលំហាត់ទី៣ដោយខ្លួនឯង។

### III. Random Access File

Random Access File គឺមានលក្ខណៈខុសពី Sequential File នៅក្នុងថា ការ ផ្ទុក និងការអានរបស់វា មានទីតាំងកំណត់ច្បាស់លាស់ ដែលអនុញ្ញាតអោយលោកអ្នក ដំណើរការទៅទាញយកទិន្នន័យ បានយ៉ាងងាយស្រួលពីទីតាំងណាមួយរបស់ File ដោយ មិនចាំបាច់ដំណើរការតាមលំដាប់ដោយដូចជា Text File ឡើយ។ Random Access File គឺវាមានទំរង់ដូចជា Floppy Disk, Hard Disk, CD ជាដើម ហើយជាទូទៅគេប្រើវា ជាមួយនឹង UDT (User Define DataType ) ។

### III.1. របៀបបើក Random Access File

Syntax:

```
Open PathAndFileName For RandomAccessMod As #FNumber [Len=Len(RecordLenght)]
```

◆ RandomAccessMod:

- Random Access Read : បើក File ដើម្បីអាន។
- Random Access Write : បើក File ដើម្បីបញ្ចូលទិន្នន័យ។
- Random Access Read Write / Random: បើក File ដើម្បីអាននិង បញ្ចូលទិន្នន័យ។

◆ Len = Len(RecordLenght) : ធ្វើការគណនាទៅលើទំហំរបស់ File ក្នុងមួយ Record ។

លោកអ្នកបានឃើញហើយថា ការបើក File ប្រភេទ Random Access គឺមិនមាន ទំរង់ជា Append ដូចជា TextFile ឡើយ ដូច្នេះដើម្បីបើក File ប្រភេទ Random Access ក្នុងគោលបំណងដើម្បី បញ្ចូលទិន្នន័យបន្ថែម លោកអ្នក ត្រូវរំកិលទីតាំង Record អោយទៅដល់ ទីតាំង Record ថ្មី។

### III.2. របៀបបញ្ចូលទិន្នន័យទៅអោយ RandomAccessFile

ដើម្បីបញ្ចូលទិន្នន័យទៅអោយ Random Access File លោកអ្នកត្រូវប្រើ Statement មួយឈ្មោះថា Put ។

Syntax:

```
Put #FNumber, [RecordPosition] , RecordVariable
```

- RecordPosition គឺជាទីតាំង Record ដែលមាននៅក្នុង File ។
- RecordVariable គឺជាអថេរដែលមានប្រភេទជា User Define Data Type

**ឧទាហរណ៍ :**

ឧបមាថាលោកអ្នកមាន Form ដូចរូបភាពទី៧.២៩ ។

☞ Source Code:

Option Explicit



```

Private Type Employee
    EmpId As String * 10
    EmpName As String * 20
    EmpSex As String * 6
End Type
Dim Emp As Employee

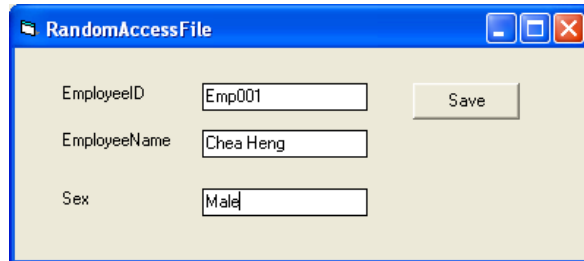
```

```

Private Sub CmdSave_Click()
    Emp.EmpId = Me.txtEmpID
    Emp.EmpName = Me.txtEmpName
    Emp.EmpSex = Me.txtEmpSex
    Open "C:\BBUEmployee.txt" For Random Access Write As #1
        Put #1, 1, Emp
    Close #1
    Call Clear(Me)
End Sub

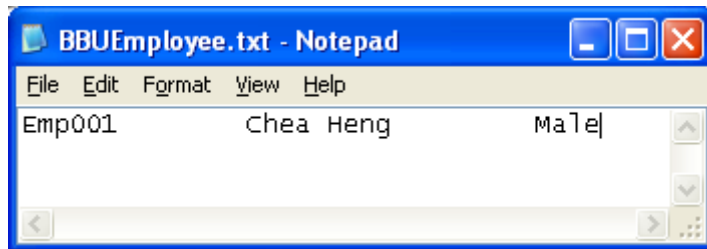
```

### រូបភាពទី២៩



លទ្ធផល:

### រូបភាពទី៣០



**វិភាគលទ្ធផល :**

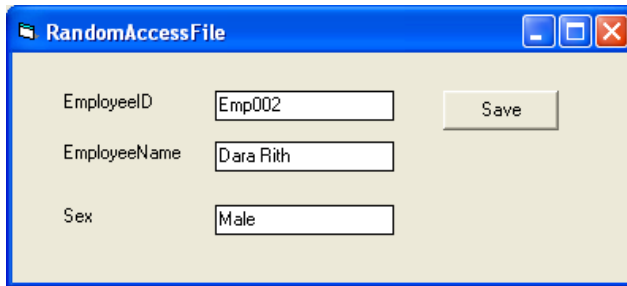
សូមពិនិត្យមើលបន្ទាត់ Put #1, 1, Emp មានន័យថាលោកអ្នកបញ្ជូនទិន្នន័យ ទៅទីតាំងនៃ Record ទីមួយ។ ប្រសិនបើលោកអ្នកវាយទិន្នន័យផ្សេងហើយចុចទៅលើ Button Save ទៀតនោះទិន្នន័យថ្មីនឹងទៅជំនួសទិន្នន័យចាស់។ ដូច្នេះដើម្បីជៀសវាង កុំអោយទិន្នន័យថ្មី ទៅជំនួសទិន្នន័យចាស់លោកអ្នកត្រូវកំណត់ទីតាំង Record អោយកើន ផងដែលនៅពេលដែល លោកចុចទៅលើ Button Save ។ ដើម្បីអោយកាន់តែច្បាស់ សូមមើលចំណុចខាងក្រោម។

### III.3. របៀបបញ្ជូនទិន្នន័យបន្ថែមទៅអោយ RandomAccessFile

**ឧទាហរណ៍ :**

ឧបមាថាលោកអ្នកមាន Form ដូចរូបភាពទី៧.៣១ ។

**រូបភាពទី៧.៣១**



**Source Code:**

```
Private Type Employee
    EmpId As String * 10
    EmpName As String * 20
    EmpSex As String * 6
End Type
Dim Emp As Employee

.....

Private Sub CmdSave_Click()
    Emp.EmpId = Me.txtEmpID
    Emp.EmpName = Me.txtEmpName
    Emp.EmpSex = Me.txtEmpSex
    Open "C:\BBUEmployee.txt" For Random Access Write As #1 Len =
        Len(Emp)
```

```

RecPost% = LOF(1) / Len(Emp) + 1
Put #1, RecPost, Emp
Close #1
Call Clear(Me)
End Sub

```

**វិភាគលទ្ធផល :**

បើយើងធ្វើការពិនិត្យទៅលើ Code ដែលមាននៅក្នុងឧទាហរណ៍នេះ គឺវា ស្រដៀងគ្នាទៅនឹង ឧទាហរណ៍ខាងលើ តែវាខុសគ្នាត្រង់ចំណុចមួយចំនួនដូចជា :

- Open "C:\BBUEmployee.txt" For Random Access Write As #1 Len=Len(Emp) មានន័យថាលោកអ្នកចាប់ផ្តើមបើក File BBUEmployee.txt ដើម្បីបញ្ចូលទិន្នន័យ ហើយចាប់ផ្តើមគិតចំនួន Bytes ក្នុងមួយ Record របស់ File នេះទៀត។

- RecPost% = (LOF(1) / Len(Emp) ) + 1  
 + LOF(1) / Len(Emp) មានន័យថារកទីតាំង Record ចុងក្រោយ ឬ ទីតាំង Byte

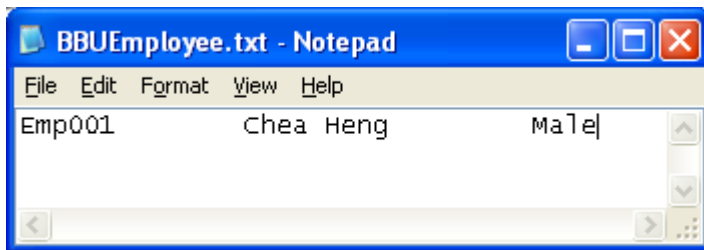
ចុងក្រោយនៃ File ។

+ (LOF(1) / Len(Emp) ) + 1 រំកិលទៅកាន់ទីតាំង Record ថ្មី ។

- Put #1, RecPost, Emp : បញ្ចូលទិន្នន័យទៅអោយ File ប៉ុន្តែទៅតាមទីតាំង PostRec ។

លទ្ធផលដែលទទួលបានគឺដូចរូបភាពទី៧.៣២ ដែលគ្មាន Record ណាទៅជំនួស Record ណាឡើយ។

**រូបភាពទី៧.៣២**



### III.4. របៀបអានទិន្នន័យចេញពី RandomAccessFile

#### III.4.1. របៀបអានទិន្នន័យចេញពី RandomAccessFile ម្តងម្កាយ Record

ដើម្បីអានទិន្នន័យចេញពី Random Access File លោកអ្នកត្រូវប្រើ Statement មួយឈ្មោះថា Get ។

Syntax:

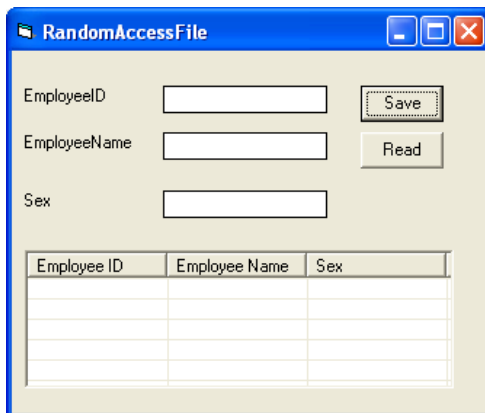
```
Get [#]filename, [recnumber], varname
```

[recnumber] ជាទីតាំង Record ដែលលោកអ្នកចង់អានទិន្នន័យ ចេញពី File ។ ហើយចំពោះ [recnumber] កាលណាលោកអ្នកមិនបោះតំលៃអោយវា មានន័យថា វាយកតំលៃ Record ទី១ ដោយស្វ័យប្រវត្តិ។

**ឧទាហរណ៍ទី១ :**

ឧបមាថាលោកអ្នកធ្វើបន្តពីរូបភាពទី៧.៣១ អោយមាន Form ទំរង់ដូច រូបភាពទី៧ .៣៣ ហើយពេលដែលលោកអ្នកចុច ទៅលើ Button Read សូមអាន Record ទី១ របស់ File BBUEmployee.bbu ចេញមកដាក់ក្នុង List View Control ទៅតាម Columns នីមួយៗ ។

**រូបភាពទី៧.៣៣**



☞ Source Code:  
Private Type Employee  
EmpId As String \* 10

```
EmpName As String * 20
EmpSex As String * 6
End Type
Dim Emp As Employee
```

---

```
Private Sub Form_Load()
    Call AddColumnHeader(Me.ListView1, "Employee ID", "Employee Name",
        "Sex")
End Sub
```

---

```
Private Sub CmdRead_Click()
    Open "C:\BBUEmployee.txt" For Random Access Read As #1 Len =
        Len(Emp)
    Get #1, , Emp
    Call AddDataToLvw(Me.ListView1, Emp.EmpId, Emp.EmpName,
        Emp.EmpSex)
    Close #1
End Sub
```

---

**លទ្ធផល :**

| Employee ID | Employee Name | Sex  |
|-------------|---------------|------|
| Emp001      | Chea Heng ... | Male |
|             |               |      |
|             |               |      |

**ឧទាហរណ៍ទី២ :**

ឧបមាថាលោកអ្នកចង់ធ្វើការអានយក Record ទី២ វិញ ដូច្នោះ Code នៅក្នុង Event Button Read ត្រូវកែប្រែដូចខាងក្រោម :

☞ Source Code:

```

.....
Private Sub CmdRead_Click()
    Open "C:\BBUEmployee.txt" For Random Access Read As #1 Len =
        Len(Emp)
    Get #1, 2, Emp
    Call AddDataToLvw(Me.ListView1, Emp.EmpId, Emp.EmpName,
        Emp.EmpSex)
    Close #1
End Sub
.....

```

**III.4.2. របៀបអានទិន្នន័យ (Records) ទាំងអស់ចេញពី  
RandomAccessFile**

**ឧទាហរណ៍ :**

ឧបមាថាលោកអ្នកមាន Form ដូចរូបភាពទី៧.៣៣

☞ Source Code:

```

.....
Private Sub CmdRead_Click()
    Dim i%
    Open "C:\BBUEmployee.txt" For Random Access Read As #1 Len =
        Len(Emp)
    Records% = LOF(1) / Len(Emp)
    For i = 1 To Records
        Get #1, i, Emp
        Call AddDataToLvw(Me.ListView1, Emp.EmpId, Emp.EmpName,
            Emp.EmpSex)
    Next
    Close #1
End Sub
.....

```

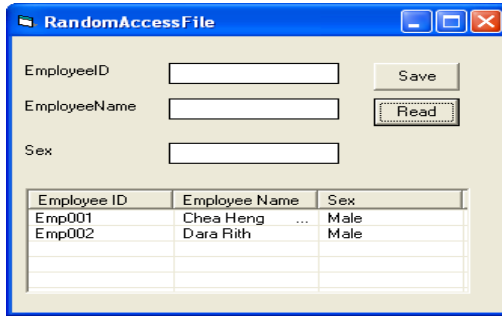
វិធាន :

-  $Records\% = LOF(1) / Len(Emp)$  គណនាចំនួន Record ទាំងអស់ដែលមាន នៅក្នុង File ។

- Get #1, i, Emp អានទិន្នន័យមួយ Record ទៅតាមតំលៃ i រហូតដល់តំលៃ i ស្មើអោយ

Record ចុងក្រោយ។

លទ្ធផល :



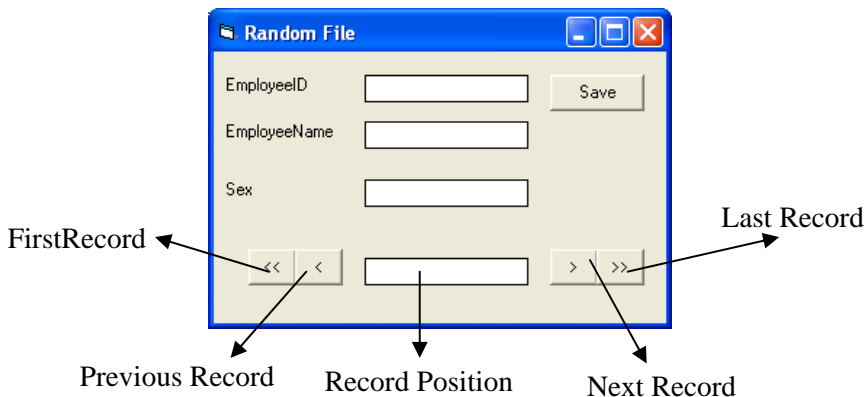
## IV. លំហាត់អនុវត្តន៍លើ Random Access File

### IV.1. លំហាត់ទី១

សូមលោកអ្នកយកលំហាត់ទី១ ដែលមាននៅក្នុង លំហាត់អនុវត្តន៍លើ Sequential File មកធ្វើសារឡើងវិញដោយប្រើជាទំរង់ Random Access File វិញ។

### IV. 2.លំហាត់ទី២

ឧបមាថាលោកអ្នកមានទំរង់ Form ដូចខាងក្រោម :



**គោលបំណងនៃលំហាត់**

- ពេលដែលលោកអ្នកចុចទៅលើ Button Save សូមបញ្ជូលទិន្នន័យទៅក្នុង File BBUEmployee.bbu ប៉ុន្តែមិនអោយស្វ័យលេខកូដ អត្តលេខរបស់បុគ្គលិកឡើយ។
- ពេលចុចលើ Button First Record សូមបង្ហាញទិន្នន័យដែលមាននៅក្នុង Record ទី១ អោយឃើញលើ Form ទៅតាម Fields នីមួយៗ ដូចជា EmployeeID, EmployeeName, Sex ហើយអោយបង្ហាញទីតាំង Record លើ Text Box Record Position ។
- ពេលចុច Button Next Record សូមរំកិលទីតាំង Record ដែលនៅខាងមុខ វាហើយបង្ហាញទិន្នន័យ។
- ពេលចុច Button Previous Record សូមរំកិលទីតាំង Record មកខាង ក្រោយតាមលំដាប់វិញ ហើយបង្ហាញទិន្នន័យនៃទីតាំងនោះ។
- ពេលចុច Button Last Record សូមរំកិលទីតាំង Record មកទីតាំងចុង ក្រោយ ហើយបង្ហាញទិន្នន័យ។

**🔍 ណែនាំ :**


- ដើម្បីរំកិលមកកាន់ទីតាំង Record ទី១ លោកអ្នកត្រូវអានទិន្នន័យពី File ដោយ បោះទីតាំង Record ទី១ ទៅអោយ RecPosition Parameter ។
- ដើម្បីរំកិលទីតាំង Record អោយទៅមុខបន្ទាប់លោកអ្នកត្រូវបោះទីតាំង Record អោយកើនម្តងមួយ ទៅអោយ RecPosition Parameter ។
- ដើម្បីរំកិលទីតាំង Record អោយថយក្រោយបន្ទាប់លោកអ្នកត្រូវ បោះទីតាំង Record អោយថយចុះម្តង១ ទៅអោយ RecPosition Parameter ។
- ដើម្បីរំកិលទីតាំង Record អោយទៅដល់ទីតាំងចុងក្រោយ លោកអ្នកត្រូវ គណនារកទីតាំង Record ចុងក្រោយអោយឃើញសិន។



### IV.3.លំហាត់ទី៣

សូមលោកអ្នកធ្វើបន្ថែមទៅលើលំហាត់ទី២ ដោយសូមអោយ Encrypt Data ទាំងអស់ពេលបញ្ចូលទិន្នន័យ ហើយពេលអានមកវិញសូម Decrypt Data ។

#### កំណែសេរីលំហាត់ទី១ (IV.1)

 Source Code:

```
Private Type Employee
    EmpId As String * 10
    EmpName As String * 20
    EmpSex As String * 6
End Type
Dim Emp As Employee
Dim Post As Integer
.....
Private Sub Form_Load()
    Post = 1
End Sub
.....
Private Sub CmdSave_Click()
    If isFieldExist("C:\BBUEmployee.bbu", Me.txtEmpID) = True Then
        MsgBox "The Employee ID=" & Me.txtEmpID & "already exist",
            vbInformation
    Exit Sub
Else
    Emp.EmpId = Me.txtEmpID
    Emp.EmpName = Me.txtEmpName
    Emp.EmpSex = Me.txtEmpSex
    Open "C:\BBUEmployee.bbu" For Random Access Write As #1 Len =
        Len(Emp)
    PostRec% = (LOF(1) / Len(Emp)) + 1
    Put #1, PostRec, Emp
    Close #1
    Call Clear(Me)
End If
End Sub
.....
Private Sub CmdPrevious_Click()
```

```

Open "C:\BBUEmployee.bbu" For Random Access Read As #1 Len =
    Len(Emp)
Records = (LOF(1) / Len(Emp))
If Post > 1 Then
    Post = Post - 1
    Get #1, Post, Emp
    Me.txtEmpID = Emp.EmpId
    Me.txtEmpName = Emp.EmpName
    Me.txtEmpSex = Emp.EmpSex

    Close #1
ElseIf Post = 1 Then
    Post = 1
    Get #1, Post, Emp
    Me.txtEmpID = Emp.EmpId
    Me.txtEmpName = Emp.EmpName
    Me.txtEmpSex = Emp.EmpSex
    Close #1
End If
Close #1
End Sub

```

---

```

Private Sub CmdNext_Click()

```

```


    Open "C:\BBUEmployee.bbu" For Random Access Read As #1 Len =
        Len(Emp)
    Records = (LOF(1) / Len(Emp))
    If Post < Records Then
        Post = Post + 1
        Get #1, Post, Emp
        Me.txtEmpID = Emp.EmpId
        Me.txtEmpName = Emp.EmpName
        Me.txtEmpSex = Emp.EmpSex
        Close #1
    ElseIf Post = Records Then
        Post = Records
        Get #1, Post, Emp
        Me.txtEmpID = Emp.EmpId
        Me.txtEmpName = Emp.EmpName
        Me.txtEmpSex = Emp.EmpSex
        Close #1
    End If
    Close #1

```

End Sub

```
.....  
Private Sub CmdLast_Click()  
    Open "C:\BBUEmployee.bbu" For Random Access Read As #1 Len =  
        Len(Emp)  
    Post = (LOF(1) / Len(Emp))  
  
    Get #1, Post, Emp  
    Me.txtEmpID = Emp.EmpId  
    Me.txtEmpName = Emp.EmpName  
    Me.txtEmpSex = Emp.EmpSex  
    Close #1  
End Sub
```

```
.....  
Function isFieldExist(FileName As String, FieldName As String) As Boolean  
    Open "C:\BBUEmployee.bbu" For Random As #1  
    Records = (LOF(1) / Len(Emp))  
    For i = 1 To Records  
        Get #1, i, Emp  
        If StrComp(Trim(Emp.EmpId), Trim(FieldName), vbTextCompare) = 0  
            Then  
                isFieldExist = True  
            Close #1  
            Exit Function  
        End If  
    Next  
    Close #1  
End Function
```

 **កំណត់ចំណាត់ទី២ និង ចំណាត់ទី៣**

សូមលោកអ្នករៀនធ្វើលំហាត់ទី២ និង ទី៣ដោយខ្លួនឯង។

===== \* =====