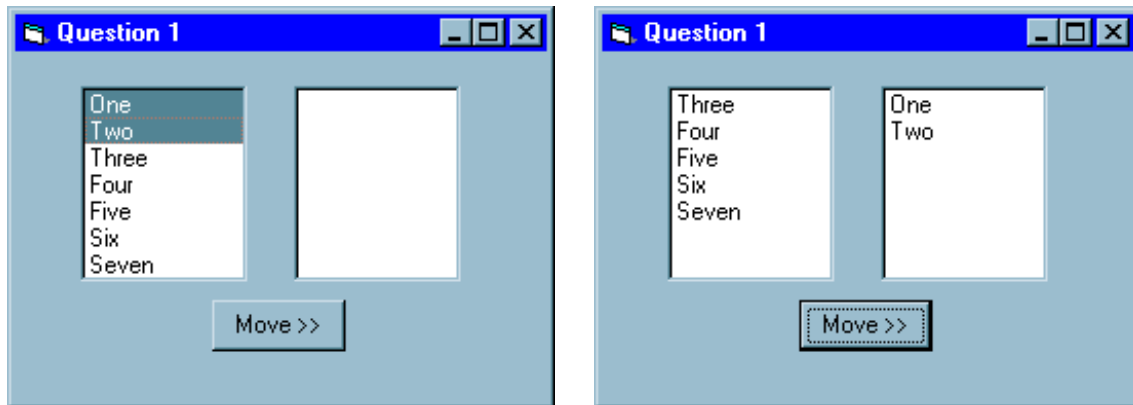
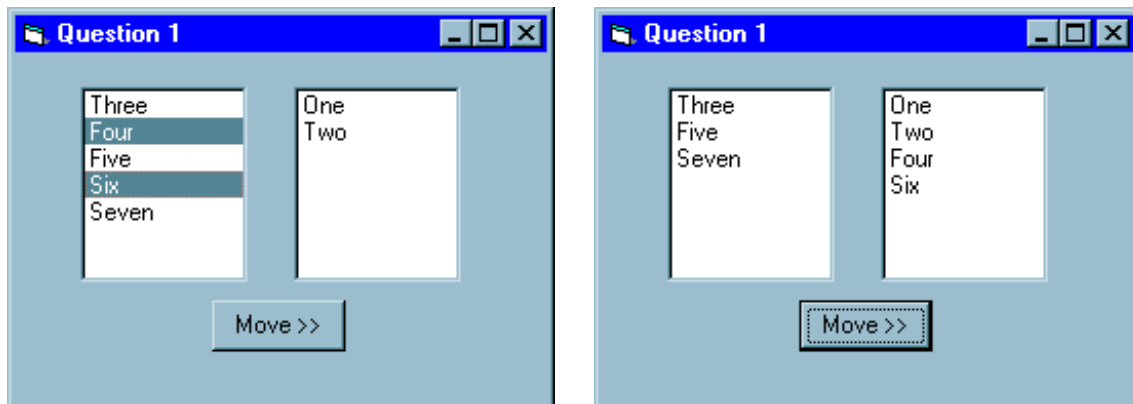


1. (13) The figures below show two list boxes before and after the user has clicked on the “**Move>>**” button. Notice that the selected items have been moved to the new list in the same relative order they had in the original list.



The second set of figures below show the two list boxes again in a before and after the move.



Here you see the selected items moved to the new list. They retain their same relative order from the original list and are placed after the existing items in the new list.

Write the code for the click event for the “**Move>>**” button that emulates the behavior described above. Assume you already have the following code:

```
Private Sub cmdMoveOldToNew_Click()  
    'complete the code here  
End Sub
```

Also assume that the two list boxes are named `lstOld` and `lstNew` and already have been populated with values.

2. (13) Assume you have a dynamic array a() declared as follows:

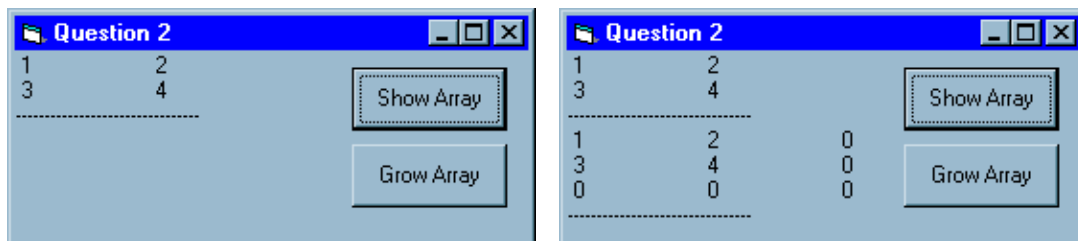
```
Dim a() As Integer
```

Assume that you initially define this array as a 2-dimensional array with 2 rows and 2 columns and then populate it with values as follows (assume the variables i, j and k are declared elsewhere):

```
ReDim a(1 To 2, 1 To 2)
For i = 1 To 2
  For j = 1 To 2
    k = k + 1
    a(i, j) = k
  Next j
Next i
```

You now want to increase the size of the array by adding 1 additional row and 1 additional column. You also want the original values of the array to exist in this new array.

The figure below on the left shows the initial array and the figure on the right shows the array after adding the new row and column (the user clicked on the “Grow Array” button and then clicked on the “Show Array” button.)

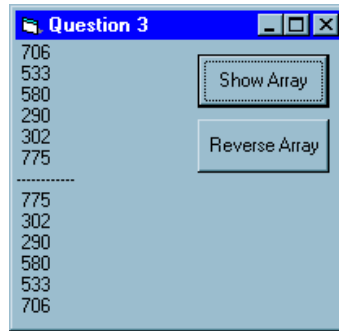


Write the code for the “Grow Array” click event. You are given the first and last lines of code as follows:

```
Private Sub cmdGrow_Click()
    'complete the code here
End Sub
```

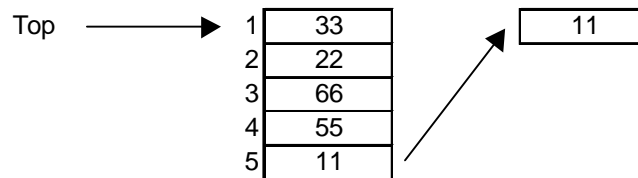
You may assume that the array is growing from 2 rows and 2 columns to 3 rows and 3 columns. That is, you do not have to write generic code that adds an additional row and column to an array that has an arbitrary number of rows and columns.

3. (13) You are to write the code that reverses the contents of an array. The figure below shows the array before and after the reverse operation has been performed.

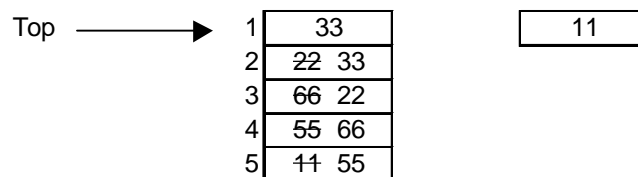


Your code needs to follow a specific algorithm. The pseudocode for this algorithm is:

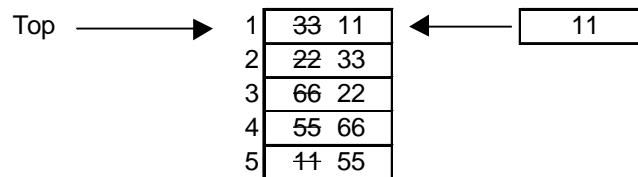
Step 1: Make a copy of the value in the last cell of the array.



Step 2: Between the last cell and the cell pointed to by "Top", move values down one cell.



Step 3: Move copy of value of former last cell to "Top".



Step 4: Move "Top" down one cell and do everything again.

Step 5: Terminate when "Top" points to last cell.

Assume that the array is named a() and has been declared and populated. You need to complete the cmdReverse_Click() event.

```
Option Explicit
Const n = 6
Dim a(1 To n) As Integer

Private Sub cmdReverse_Click()
    ' complete the code here
End Sub
```

4. (12) For each of the following code segments, determine the value of the variable K in the form print statement.

a. K = 0
For I = 1 To 4
 For J = 1 To 3
 K = K + 1
 Next J
Next I
Form1.Print K

b. K = 0
For I = 1 To 5 Step 2
 K = K + 1
 For J = 1 To K
 K = K + 1
 Next J
Next I
Form1.Print K

c. K = 0
For I = 1 To 4
 For J = I To 1 Step -1
 K = K + 1
 Next J
Next I
Form1.Print K

5. (13) Given the following narrative, construct the appropriate programmer-defined type definitions. Then use the type definitions to define a variable that can store information on up to 50 Customers.

“A Customer includes a string Name field, a string Address field, an integer field to store the Number of Purchases, and a field to store up to 100 Purchases. Each Purchase includes a Product field, an integer Quantity field, and a currency Price field. Finally a Product includes string Name and Description fields.”

6. (12) Briefly explain why most database related run-time errors occur in the Recordset's Refresh method.
7. (12) What is the **primary** difference between Recordset Find methods and Move methods?

8. (12) You are given the following code:

```
For J = 1 To 10 Step 2
  For K = J To 5
    X = X + 1
  Next K
Next J
```

Rewrite this code with an equivalent “Do” loop structure. The code you write should be totally equivalent to the given code.

1. The following code moves the items from one list to the other:

```
Private Sub cmdMoveOldToNew_Click()  
    Dim oldIndex As Integer  
    Dim newIndex As Integer  
  
    oldIndex = 0  
    newIndex = lstNew.ListCount  
  
    Do While oldIndex < lstOld.ListCount  
        If lstOld.Selected(oldIndex) Then  
            lstNew.AddItem lstOld.List(oldIndex), newIndex  
            lstOld.RemoveItem oldIndex  
            newIndex = newIndex + 1  
        Else  
            oldIndex = oldIndex + 1  
        End If  
    Loop  
  
End Sub
```

2. The following code “grows” the 2 x 2 array into a 3 x 3 array and retains the original values.

```
Private Sub cmdGrow_Click()  
    Dim x(1 To 2, 1 To 2) As Integer  
    Dim i As Integer, j As Integer  
    'copy original array  
    For i = 1 To 2  
        For j = 1 To 2  
            x(i, j) = a(i, j)  
        Next j  
    Next i  
    'add a new row and col to original array  
    ReDim a(1 To 3, 1 To 3)  
    'restore original values  
    For i = 1 To 2  
        For j = 1 To 2  
            a(i, j) = x(i, j)  
        Next j  
    Next i  
End Sub
```

3. The following code reverses the contents of the array as described in the algorithm.

```
Private Sub cmdReverse_Click()  
    Dim top As Integer, j As Integer  
    Dim t As Integer  
    For top = 1 To n - 1  
        t = a(n)  
        For j = n To top + 1 Step -1  
            a(j) = a(j - 1)  
        Next j  
        a(top) = t  
    Next top  
End Sub
```

4. a. 12
b. 14
c. 10

5. The following code defines the types described in the problem.

```
Type Product  
    Name As String  
    Description As String  
End Type
```

```
Type Purchase  
    theProduct As Product  
    Quantity As Integer  
    Price As Currency  
End Type
```

```
Type Customer  
    Name As String  
    Address As String  
    NoPurchases As Integer  
    PurchaseList(1 To 100) As Purchase  
End Type
```

```
Dim CustList(1 To 50) As Customer
```

6. The Refresh method of the data control actually causes the information defined in the VB code (such as database name and record source) to be passed to the DB engine for processing. The DB engine does the actual processing on the database and returns either a valid record set or an error indicator. Thus, when a DB error occurs, VB associates it with the code that it (VB) was executing, i.e., the Refresh method.

7. Both the Find and Move methods move through the record set (in this way they are similar). They differ in that the movement for the Move methods is associated with the physical record position in the database. MoveNext move to the next physical record in the database. On the other hand, the Find methods are associated with “logical” records (records with an ordering associated with a user- or problem-defined sequence such as the “next” Freshman or the “Last” Junior).

8. The following code is equivalent to the code given in the problem.

```
J = 1
Do While J <= 10
  K = J
  Do While K <= 5
    X = X + 1
    K = K + 1
  Loop
  J = J + 2
Loop
```