1. (20) A palindrome is a word or phrase that reads the same backward or forward. For example "Name no one man".

   Assume that the user has entered a phrase into a text box. Write a VB code segment to determine if this phrase is a palindrome. If it is, use a message box to display the message "It is a palindrome". Otherwise, display the message "It is not a palindrome".

   Do not worry about blanks or case (upper/lower). Assume that the user has not typed any blanks into the phrase and has entered the phrase as only lower case. For example, the phrase above would have been entered into the text box as " namenooneman".

   For 10 bonus points, allow the user to enter blanks and upper/lower case. (Note: don't consider this until you have the initial version of code written.)

2. (18) Define a user-defined data type that can store information on products and the parts that are used to assemble the products. Specifically, the data type should be able to store the ProdNo (string), ProdDesc (string), NoParts (integer - the number of parts used in the product), and a list of up to 30 parts. The part information should include the PartNo (string), PartDesc (string), and Quantity (integer - the number of the part used in the product).

   After defining the product data type, declare a variable of the type. Then show the VB statements that define the following product.

ProdNo	A123
ProdDesc	ABD Widget
NoParts	3
PartNo	444
PartDesc	Red Thingy
Quantity	3
PartNo	555
PartDesc	Bolts
Quantity	5
PartNo	666
PartDesc	Cover
Quantity	1

3. (15) You have an array that stores product numbers and corresponding prices. You use this array to look up prices by performing a binary search for product numbers. Assume that you need to be able to add new products to the array during execution. Suggest a way of adding a new product to the array **without** performing an entire multipass sort (like the bubble sort). You do not need to write code - just explain in words a strategy you might use. The binary search should still work after you add the new product to the array.

4. (15) Your program searches an array for information on parts stored in inventory. You use a sequential search approach. Analysis of your data reveals that 80% of your searches search for 10% of the parts. That is, 10% of the parts are very popular and the remainder are not used very often. What can you do to make your sequential searches as efficient as possible?

5. (17) You have a two-dimensional array with an equal number of rows and columns. Write a complete VB procedure that swaps the elements of the array along its left-to-right diagonal. For example, the array following array

43	14	34	25
32	**23**	18	21
75	43	**28**	55
48	36	19	**22**

would become:

43	32	75	48
14	**23**	43	36
34	18	**28**	19
25	21	55	**22**

The procedure sh ould be passed two arguments - the array and an integer value which indicates the number of rows/columns. The procedure should swap the values in the actual array passed to the procedure.

6. (15) VB does not allow an array to be passed to a procedure using the ByVal qualifier. Explain why this restriction applies to arrays and not to simple variables.

Answer Key

1.
```
Private Sub Command1_Click()
Dim S As String
Dim I As Integer, K As Integer

S = Text1.Text

For I = 1  To Len(S) / 2
       K = Len(S) + 1 - I
       If Mid$(S, I, 1) <>  Mid$(S, K, 1)  Then
            MsgBox "Its not a  palindrone"
            Exit Sub
       End If
Next I

MsgBox "It is a palindrome"

End Sub
```

Bonus code (replace `"S = Text1.Text"` with the following):

```
Dim NewS As String
S = Text1.Text
For I = 1  To Len(S)
  If Mid$(S, I, 1) <> " "  Then
       NewS = NewS & LCase$(Mid$(S, I, 1))
  End If
Next I
S = NewS
```

2.
```
Type Part
  PartNo  As String
  PartDesc  As String
  Quantity  As Integer
End Type

Type Product
  ProdNo  As String
  ProdDesc  As String
  NoParts  As Integer
  PartList (1 to 30)  As Part
End Type

Dim SampleProduct  As Product

SampleProduct.ProdNo = "A123"
```

```
SampleProduct.ProdDesc = "ABD Widget"
SampleProduct.NoParts = 3

SampleProduct.PartList(1).  PartNo = "444"
SampleProduct.PartList(1).  PartDesc = "Red  Thingy"
SampleProduct.PartList(1).Quantity = 3

SampleProduct.PartList(2).  PartNo = "555"
SampleProduct.PartList(2).  PartDesc = "Bolts"
SampleProduct.PartList(2).Quantity = 5

SampleProduct.PartList(3).  PartNo = "666"
SampleProduct.PartList(3).  PartDesc = "Cover"
SampleProduct.PartList(3).Quantity = 1
```

3. You can "insert" the new product into the array at its proper spot. To do this, you can scan the array beginning at the first row until you find a product number greater than the one you are going to insert. Remember the row where you found this value (call it FoundLoc). Then start with the last row and move every row down one. Do this for the last row up to and including FoundLoc. Finally store the new product information in row FoundLoc.

4. Put the most popular parts at the beginning of the array and the least popular at the end of the array. Since a sequential search always starts at the beginning of the array, you will find the popular ones fast because they are at the beginning of the array.

5.
```
Private Sub Invert(X()  As Integer, ByVal N As Integer)
Dim I As Integer, J As Integer
Dim T As Integer

For I = 1  To N
       For J = I + 1  To N
            T = X(I, J)
            X(I, J) = X(J, I)
            X(J, I) = T
       Next J
Next I

End Sub
```

6. There are two reasons which both come from the fact that a call by value uses a "copy" of the argument. Making a copy of a simple variable is fast and doesn't consume too much memory. However, making a copy of an array consumes both a lot of memory and processing time. It is not operationally feasible.