Please use your own paper to answer the following questions. Point values are shown in parentheses.

1. (10) Assume that you have a **sequential** search function designed to search an **ordered**, one-dimensional array for a given target value. If the value is found, then the function returns the array element number where the target value was located. If the value is not found, then the function returns a -1.

   What would happen if you called this function and passed it an **unordered** array instead of an ordered array? Explain.

2. (10) Consider the following procedure:

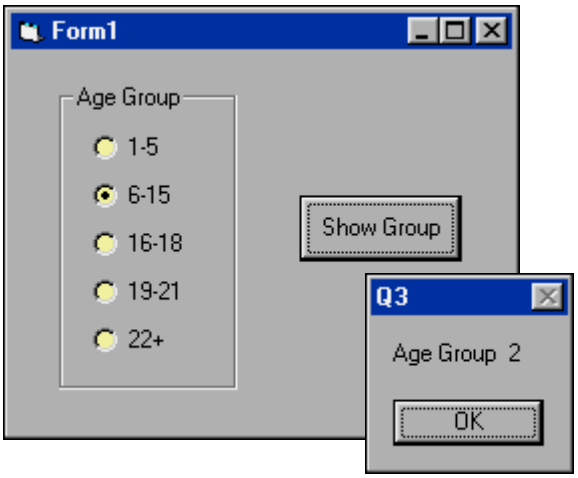
```
Private Sub cmdLooper_Click()
Dim K As Single

K = 0.01
Do Until K = 1
    K = K + 0.001
    Form1.Cls       ‘ clear the form print area
    Form1.Print K   ‘ print K on the form
Loop
Beep

End Sub
```

   There is a potentially serious problem with this code. What is the problem and how would you fix it?

3. (25) Consider the following form and message box generated when the user clicks on the Show Group button:



You want to write code for the Show Group button that determines which option button was chosen and displays via a message box the appropriate group number (1 - 5 in this example). Your solution should be designed so that the number of lines in the click event do not change if you add or delete option buttons in the frame. That is, if there are a new number of options, you should only have to change the value of a variable to indicate the new number of option buttons. In addition, there should be no need to add/delete any other lines of code in any other event procedures.

a. Describe your approach. That is, indicate what you need to set up in order to be able to accomplish the objectives described above. [Hint: think about how you can treat a group of controls as if they were one.]

b. Write the code for the click event procedure. My solution has 8 lines of code (including the Sub and End Sub statements) and works regardless of the number of option buttons.

4. (30) You need a programmer-defined data type that describes a product and the parts used in assembling the product. Information on each product includes a product number (10-character string), a product description (30-character string), and an integer value indicating the total number of different parts used to assemble the product. Assume that the maximum number of parts will not exceed 100. Then for each part, you need to store the part number (5-chartacter string), the quantity (integer) of the specific part used in this product (like 2 axles [parts] in a wagon [product]), and finally the unit cost of the part (currency).

   a. Write the valid VB statement(s) to define this data type.

   b. Write the valid VB statement(s) to declare an array that will be able to store information on up to 50 products.

   c. Write a VB procedure that is passed the array described in (b) and computes the total parts cost for each product. These total costs should be returned in another array. Consider using the following procedure heading:

```
Sub ProdCost ( ProdArray() As ProdType, _
               NumProds As Integer, _
               PartCost() As Currency )
```

   where ProdType is the name of the data type defined in (a), NumProds is the actual number of products stored in the array declared in (b), and PartCost is an array that has one cell for each product. The value you store in PartCost(1) will be the total cost of parts for product 1 stored in ProdArray(1).

   A product's total cost for parts is computed by going through all the parts for that particular product and summing the product of the quantity of each part times its unit cost.

5. (10) VB uses the concept of a recordset to describe the set of records associated with a specific data control.

   Explain how a recordset and a table in a relational database are similar and how they are different.

6. (15) You are given the following code segment:

```
For K = 1 To N
    Form1.Print K
Next K
```

   a. Write equivalent loops using a

      (1) **Do Until**…**Loop** structure

      (2) **Do While**…**Loop** structure

   b. Can you write an equivalent loop using the **Do**…**Loop Until** structure? Explain.

Answer Key

1. The search would fail. A sequential search of an ordered array stops searching when it finds a value in the array that is greater than or equal to the target being searched for. With an ordered array, if you encounter a value greater than the target, the target cannot possible exist in the array so you stop searching. This termination assumption is not valid if the array is unordered. With an unordered array, if you stop when you find a value in the array greater than the target, you have no guarantee than your target does not exist further down in the array.

2. The problem relates to the Single data type and data representation errors. The fractional numbers (.01 and .001) have representation errors. If you accumulate the number .001, the representation errors are compounded. The result is the sum (K) may never equal exactly 1. In fact, this program goes into an infinite loop in VB.

   To solve the problem you could:

   a. change K to Currency type, or
   b. change the terminal condition to K >= 1.

3. (a) The approach requires the use of a control array. The option buttons should be placed in a control array so that they can be referenced via their Index property. Option buttons added to the form would also be added to the control array with their own unique Index property.

   (b) The solution below assumes that the Index property of the first option button is set to zero. This is why the expression “K + 1” is used in the message box.

```
Private Sub cmdShowAge_Click()
Dim K As Integer

For K = 0 To 4
    If optAge(K).Value Then
        MsgBox "Age Group " & Str$(K + 1)
    End If
Next K

End Sub
```

4. (a)

```
Public Type PartType
    PartNumber As String * 5
    Quantity As Integer
    UnitCost As Currency
End Type

Public Type ProdType
    ProdNumber As String * 10
    ProdDesc As String * 30
    NumParts As Integer
    PartList(1 To 100) As PartType
End Type
```

(b)

```
Dim Products(1 To 50) As ProdType
```

(c)

```
Private Sub ProdCost(ProdArray() As ProdType, _
                                  NumProds As Integer, _
                                  PartCost() As Currency)
Dim K As Integer, J As Integer
Dim Sum As Integer

For K = 1 To NumProds
   Sum = 0
   For J = 1 To ProdArray(K).NumParts
     Sum = Sum + ProdArray(K).PartList(J).Quantity * _
                 ProdArray(K).PartList(J).UnitCost
   Next J
   PartCost(K) = Sum
Next K

End Sub
```

5. A recordset and a database table are essentially the same thing as long as the data control is associated with a database table, i.e., the data control's RecordSource property is set equal to a table. However, if the RecordSource is set equal to an SQL query, then the recordset has no direct relationship to a database table.

6. (a)

```
K = 1
Do Until K > N
    Form1.Print K
    K = K + 1
Loop

K = 1
Do While K <= N
    Form1.Print K
    K = K + 1
Loop
```

(b) Both the **For…Next** and the **Do Until…Loop** and **Do While…Loop** structures have the potential to be executed no times. The **Do…Loop Until** structure must be executed at least one time. Thus, there is no way to write an equivalent **Do…Loop Until** structure.