Chapter 10 Student Book Answers

What you should already know

```
DECLARE Largest, Smallest, Counter : INTEGER
DECLARE NumberStore : ARRAY[1:5} OF INTEGER
FOR Counter ← 1 To 5
    OUTPUT "Enter Number ", Counter
    INPUT NumberStore[Counter]
NEXT Counter
Largest ← NumberStore[1]
Smallest ← NumberStore[1]
FOR Counter ← 2 To 5
    IF NumberStore[Counter] > Largest
      THEN
        Largest ← NumberStore[Counter]
    ENDIF
    IF NumberStore[Counter] < Smallest
      THEN
        Smallest ← NumberStore[Counter]
    ENDIF
NEXT Counter
OUTPUT "Largest number ", Largest
OUTPUT "Smallest number ", Smallest
```

Activity 10A

a) String
b) Integer
c) Real
d) Boolean
e) Date

Activity 10B

```
DECLARE MyName : STRING
DECLARE NumberOfChildren : INTEGER
DECLARE RaceTime : REAL
DECLARE DoorState : BOOLEAN
```

Python

```
MyName = "Helen"
NumberOfChildren = 27
RaceTime = 45.13
DoorState = True
```

VB

```
Dim MyName As String
Dim NumberOfChildren As Integer
Dim RaceTime As Decimal
Dim DoorState As Boolean
```

Java

```
String MyName;
int NumberOfChildren;
float RaceTime;
boolean DoorState;
```

Activity 10C

```
TYPE
TStudentRecord
    DECLARE name : STRING
    DECLARE dateOfBirth : DATE
    DECLARE class : STRING
    DECLARE gender : STRING
ENDTYPE
DECLARE MyStudent : TStudentRecord
MyStudent.name ← "Ahmad Sayed"
MyStudent.dateOfBirth ← 10/March/2010
MyStudent.class ← "5A"
MyStudent.gender ← "Male"
OUTPUT "Name ", MyStudent.name
OUTPUT "Date of birth ", MyStudent. dateOfBirth
OUTPUT "Class ", MyStudent.class
OUTPUT "Gender ", MyStudent.gender
```

Python

```
#student record
import datetime
class TStudentRecord:
    def _init_(self):
        self.name =""
        self.dateOfBirth = datetime.datetime()
        self.group = ""
        self.gender =""

myStudent = TStudentRecord()
myStudent.name = "Ahmad Sayed"
myStudent.dateOfBirth = datetime.datetime(2010, 3, 10)
myStudent.group = "5A"
myStudent.gender ="Male"
print ("Name ", myStudent.name)
print ("Date of Birth ", myStudent.dateOfBirth)
print ("Class ", myStudent.group)
print ("Gender ", myStudent.gender)
```

VB

```
'student record
Module Module1

    Sub Main()

        Dim myStudent As TStudentRecord
        myStudent.name = "Ahmad Sayed"
        myStudent.dateOfBirth = #10/03/2010#
        myStudent.group = "5A"
        myStudent.gender = "Male"
        Console.WriteLine("Name " + myStudent.name)
        Console.Write("Date of Birth ")
        Console.WriteLine(myStudent.dateOfBirth)
        Console.WriteLine("Class " + myStudent.group)
        Console.WriteLine("Gender " + myStudent.gender)
        Console.ReadKey()
    End Sub
    Structure TStudentRecord
        Dim name As String
        Dim dateOfBirth As Date
        Dim group As String
        Dim gender As String
    End Structure
End Module
```

Java

```
//student record
import java.util.*;
class ACTIVITY10C {
    static class TStudentRecord {
        String name;
        String dateOfBirth;
        String group;
        String gender;

        public void TstudentRecord() {
            name = "";
            dateOfBirth = "";
            group ="";
            gender ="";
        }
    }

   public static void main(String args[]){
      TStudentRecord myStudentRecord = new TStudentRecord();
      myStudentRecord.name = "Ahmad Sayed";
      myStudentRecord.dateOfBirth = "10/03/2010";
      myStudentRecord.group = "5A";
      myStudentRecord.gender = "Male";

      System.out.println("Name  " + myStudentRecord.name);
      System.out.println("Date of Birth  " +
myStudentRecord.dateOfBirth);
      System.out.println("Group  " + myStudentRecord.group);
      System.out.println("Gender " + myStudentRecord.gender);
      }
    }
```

Activity 10D

```
OUTPUT "Enter these 9 values in order 27, 19, 36, 42, 16, 89, 21, 16,
55"
FOR counter ← 0 TO 8
    OUTPUT "Enter next value "
    INPUT myList[counter]
NEXT counter
```

Python

```
# 1-D list
myList = [27, 19, 36, 42, 16, 89, 21, 16, 55]
print (*myList)
```

VB

```
'1-D Array
Module Module1

    Sub Main()
        Dim myList = New Integer() {27, 19, 36, 42, 16, 89, 21, 16, 55}
        Dim index As Integer
        For index = 0 To myList.GetUpperBound(0)
            Console.Write(myList(index))
            Console.Write(" ")
        Next
        Console.ReadKey() 'wait for keypress
    End Sub

End Module
```

Java

```
//1-D Array

class ACTIVITY10D {
   public static void main(String args[]){
      int[] myList = {27, 19, 36, 42, 16, 89, 21, 16, 55};
      for(int index = 0; index < myList.length; index ++){

      System.out.println (myList[index] + " ");

    }
   }
}
```

Activity 10E

```
OUTPUT "Enter these values in order 27, 19, 36, 42, 16, 89, 21, 16, 55"
OUTPUT "Enter these values in order 31, 67, 98, 22, 35, 46, 71, 23, 11"
OUTPUT "Enter these values in order 17, 48, 29, 95, 61, 47, 28, 13, 77"
FOR columnCounter ← 0 TO 2
FOR rowCounter ← 0 TO 8
    OUTPUT "Enter next value "
    INPUT myArray[rowCounter, columnCounter]
NEXT rowCounter
NEXT columnCounter
```

Python

```
# 2-D list
myArray = [[27, 31, 17], [19, 67, 48],[36, 98, 29],[42, 22, 95],
   [16, 35, 61], [89, 46, 47], [21, 71, 28], [16, 23, 13], [55, 11, 77]]
print(*myArray)
```

VB

```
' 2-D Array
Module Module1

    Sub Main()
        Dim myArray = New Integer(8, 2) {{27, 31, 17}, {19, 67, 48},
            {36, 98, 29}, {42, 22, 95}, {16, 35, 61}, {89, 46, 47},
            {21, 71, 28}, {16, 23, 13}, {55, 11, 77}}
        Dim rowIndex, columnIndex As Integer
        For rowIndex = 0 To myArray.GetUpperBound(0)
            For columnIndex = 0 To myArray.GetUpperBound(1)
                Console.Write(myArray(rowIndex, columnIndex))
                Console.Write(" ")
            Next
            Console.WriteLine()
        Next
        Console.ReadKey() 'wait for keypress
    End Sub

End Module
```

Java

```
//2-D Array

class ACTIVITY10E {
   public static void main(String args[]){
      int[][] myArray = {{27, 31, 17}, {19, 67, 48},
            {36, 98, 29}, {42, 22, 95}, {16, 35, 61}, {89, 46, 47},
            {21, 71, 28}, {16, 23, 13}, {55, 11, 77}};
      for(int rowIndex = 0; rowIndex < myArray.length; rowIndex ++){
      for(int columnIndex = 0; columnIndex < myArray[rowIndex].length;
columnIndex ++){
      System.out.print(myArray[rowIndex][columnIndex] + " ");
      }
      System.out.println();
    }
   }
}
```

Activity 10F

1-D Array uses:

List of names, list of exam results, leader board scores for a game

2-D Array uses:

Seats booked for a show, homework marks for a class, list of book titles and their authors

Records are used when several pieces of information need to be stored about each indexed item and the information has different data types. For example, student name, exam mark, exam grade and date of birth.

Activity 10G

```
DECLARE myList : ARRAY[0:8] OF INTEGER
DECLARE upperBound : INTEGER
DECLARE lowerBound : INTEGER
DECLARE index : INTEGER
DECLARE item : INTEGER
DECLARE found : BOOLEAN
DECLARE foundIndex : INTEGER
upperBound ← 8
lowerBound ← 0
OUTPUT "Please enter item to be found"
INPUT item
found ← FALSE
index ← lowerBound
REPEAT
    IF item = myList[index]
      THEN
        found ← TRUE
    ENDIF
    index ← index + 1
UNTIL (found = TRUE) OR (index > upperBound)
IF found
  THEN
    OUTPUT "Item found at index ", foundIndex
  ELSE
    OUTPUT "Item not found"
ENDIF
```

Python

```
#linear search
myList = [27, 19, 36, 42, 16, 89, 21, 16, 55]
print (*myList)
item = int(input("Please enter item to be found "))
lowerBound = 0
upperBound = len(myList) - 1
found = False
index = lowerBound
while (not found) and (index <= upperBound):
    if item == myList[index]:
        found = True
        foundIndex = index
    index = index + 1
if found:
    print ("Item found at index ", foundIndex)
else:
    print ("Item not found")
```

VB

```
'linear search
Module Module1

    Sub Main()
        Dim myList = New Integer() {27, 19, 36, 42, 16, 89, 21, 16, 55}
        Dim item, index, foundIndex As Integer
        Dim found = False
        Console.Write("Please enter your item to be found ")
        item = Console.ReadLine()
        index = myList.GetLowerBound(0)
        Do
            If item = myList(index) Then
                found = True
                foundIndex = index
            End If
            index = index + 1
        Loop Until found Or index > myList.GetUpperBound(0)
        If found Then
            Console.WriteLine("Item found at index " +
foundIndex.ToString())
        Else
            Console.WriteLine("Item not found")
        End If
        Console.ReadKey() 'wait for keypress
    End Sub

End Module
```

Java

```
//linear search
import java.util.Scanner;
class ACTIVITY10G {
   public static void main(String args[]){
      Scanner myObj = new Scanner(System.in);
      int[] myList = {27, 19, 36, 42, 16, 89, 21, 16, 55};
      int index = 0;
      int foundIndex = -1;
      boolean found = false;
      System.out.println("Please enter your item to be found ");
      int item = myObj.nextInt();
      do {
         if (item == myList[index]) {
          found = true;
          foundIndex = index;
        }
        index = index + 1;
      }
      while (!found && index< myList.length);
      if (found){
         System.out.println ("Item found at index + foundIndex);
        } else {
            System.out.println ("Item not found");
    }
   }
}
```

Activity 10H

Python

```
#bubble sort
myList = [27, 19, 36, 42, 16, 89, 21, 16, 55]
print (*myList)
lowerBound = 0
top = len(myList) - 1
swap = True

while (swap and top > lowerBound):
     swap = False
     for index in range (lowerBound, top):
           if myList[index] > myList[index + 1]:
            temp = myList[index]
            myList[index] = myList[index + 1]
            myList[index + 1] = temp
            swap = True
     top = top - 1
print (*myList)
```

VB

```
'Bubble Sort
Module Module1
    Sub Main()
        Dim myList = New Integer() {27, 19, 36, 42, 16, 89, 21, 16, 55}
        Dim index, top, temp As Integer
        Dim swap As Boolean
        top = myList.GetUpperBound(0) - 1
        For index = myList.GetLowerBound(0) To myList.GetUpperBound(0)
            Console.Write(myList(index).ToString() + " ")
        Next
        Console.WriteLine()
        Do
            swap = False
            For index = myList.GetLowerBound(0) To top
                If myList(index) > myList(index + 1) Then
                    temp = myList(index)
                    myList(index) = myList(index + 1)
                    myList(index + 1) = temp
                    swap = True
                End If
            Next
            top = top - 1
        Loop Until Not swap Or top = myList.GetLowerBound(0)
        For index = myList.GetLowerBound(0) To myList.GetUpperBound(0)
            Console.Write(myList(index).ToString() + " ")
        Next
        Console.WriteLine()
        Console.ReadKey() 'wait for keypress
    End Sub
End Module
```

Java

```
//bubble sort
class ACTIVITY10H {
   public static void main(String args[]){
      int[] myList = {27, 19, 36, 42, 16, 89, 21, 16, 55};
      int top = myList.length - 1;
      int temp, index;
      int lowerBound = 0;
      boolean swap;
      for (index = lowerBound; index <= top; index++){
          System.out.print(myList[index] + " ");
        }
      System.out.println();
      do {
         swap = false;
          for (index = lowerBound; index < top; index++){
            if (myList[index] > myList[index + 1]) {
             temp = myList[index];
             myList[index] = myList[index + 1];
             myList[index + 1] = temp;
             swap = true;
            }
          }
          top = top - 1;
       }
      while (swap && (top >  lowerBound));
      for (index = lowerBound; index <= myList.length - 1; index++){
          System.out.print(myList[index] + " ");
        }
      System.out.println();
    }
}
```

Activity 10I

Extended pseudocode

```
DECLARE textLn : STRING
DECLARE myFile : STRING
myFile ← "myText.txt"
OPEN myFile FOR WRITE  // write lines of text to file
REPEAT
    OUTPUT "Please enter a line of text"
    INPUT textLn
    IF textLn <> ""
      THEN
        WRITEFILE, textLn
      ELSE
        CLOSEFILE(myFile)
    ENDIF
UNTIL textLn = ""
OUTPUT "The file contains these lines of text:"
OPEN myFile FOR READ // read lines of text to file
REPEAT
    READFILE, textLn
    OUTPUT textLn
UNTIL EOF(myFile)
CLOSEFILE(myFile)
OPEN myFile FOR APPEND // write more lines of text to file
REPEAT
    OUTPUT "Please enter a line of text"
    INPUT textLn
    IF textLn <> ""
      THEN
        WRITEFILE, textLn
      ELSE
        CLOSEFILE(myFile)
    ENDIF
UNTIL textLn = ""
OUTPUT "The file now contains these lines of text:"
OPEN myFile FOR READ // rad all the lines of text to file
REPEAT
    READFILE, textLn
    OUTPUT textLn
UNTIL EOF(myFile)
CLOSEFILE(myFile)
```

Python

```
# writing to and reading a line of text from a file
myFile = open ("myText.txt","w")
textLn = "start"
while textLn != "":
    textLn = input("Please enter a line of text ")
    if textLn != "":
        myFile.write(textLn + "\n")
    else: myFile.close()
print("The file contains these lines of text")
myFile = open ("myText.txt","r")
textLn = myFile.read()
print(textLn)
```

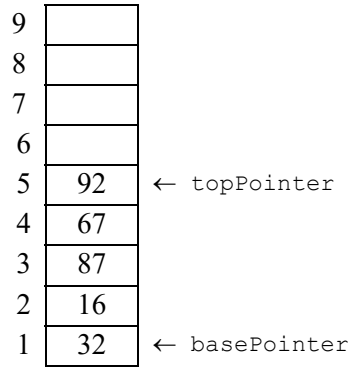
VB

```
'writing to and reading from a text file
Imports System.IO

Module Module1

    Sub Main()
        Dim textLn As String
        Dim objMyFileWrite As StreamWriter
        Dim objMyFileRead As StreamReader
        objMyFileWrite = New StreamWriter("textFile.txt")
        Do
            Console.Write("Please enter a line of text  ")
            textLn = Console.ReadLine()
            objMyFileWrite.WriteLine(textLn)
        Loop Until textLn = ""
        objMyFileWrite.Close()

        objMyFileRead = New StreamReader("textFile.txt")
        Do
            textLn = objMyFileRead.ReadLine
            Console.WriteLine(textLn)
        Loop Until textLn = ""
        objMyFileRead.Close()

        Console.ReadLine()

    End Sub

End Module
```

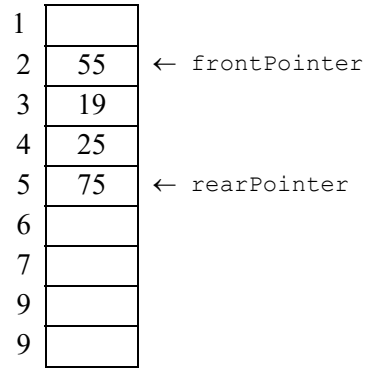
Java

```
//writing to and reading from a text file
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

class ACTIVITY10I {

    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        String textLn;
        try {
            FileWriter  myFileWriter  =  new  FileWriter("textFile.txt",
false);
            PrintWriter myPrintWriter = new PrintWriter(myFileWriter);

            do {
                System.out.println("Please enter a line of text  ");
                textLn = myObj.next();
                myPrintWriter.printf("%s" + "%n", textLn);
              }
            while (!textLn.equals("end"));
            myPrintWriter.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

        try {
            FileReader myFileReader  = new FileReader("textFile.txt");
            BufferedReader myBufferReader  = new
BufferedReader(myFileReader);

            do {
                textLn = myBufferReader.readLine();
                System.out.println(textLn);
            }
            while(!textLn.equals("end"));
            myFileReader.close();

        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}
```

Activity 10J

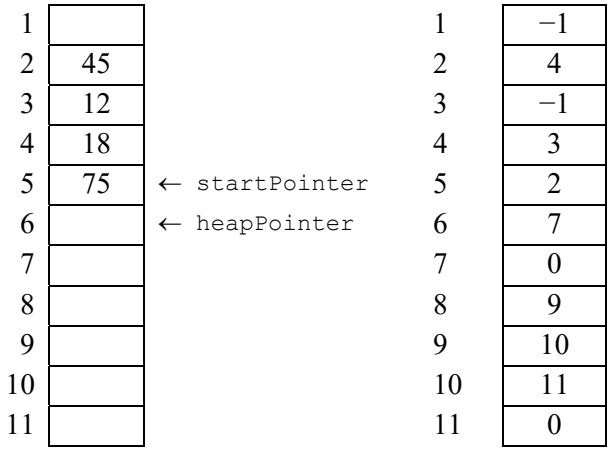


Activity 10K



Length of queue is 4

Activity 10L



End of chapter questions

1 a) Stack – see 10.4

b) Queue – see 10.4

c) Linked List – see 10.4

A composite data type is defined using an existing data type, for example a record of name and address which are both of type string.

Read the all data items from a stack (FIFO) into a queue (LIFO) then read them back into the stack.

Top of Queue
Item 1
Item 2
Item 3
Item 4

Top of Stack
Item 1
Item 2
Item 3
Item 4

Top of Queue
Item 4
Item 3
Item 2
Item 1

Items read into stack in the order 1, 2, 3, and 4

Items popped off the stack and placed back in the queue in the order 4, 3, 2 and 1

2 a)
```
myFile ← "Student.txt"
CREATE myFile
OPEN myFile FOR WRITE
```

b)
```
OPEN myFile FOR APPEND
WRITEFILE myFile, StudentRecord
```

c)
```
OPEN myFile FOR READ
OPEN myNewFile FOR WRITE
REPEAT
   READFILE myFile, StudentRecord
   IF StudentRecord.name <> "Not Wanted"
     THEN
       WRITEFILE myFile, StudentRecord
   ENDIF
UNTIL EOF (myFile)
CLOSEFILE(myFile)
CLOSEFILE(myNewFile)
```

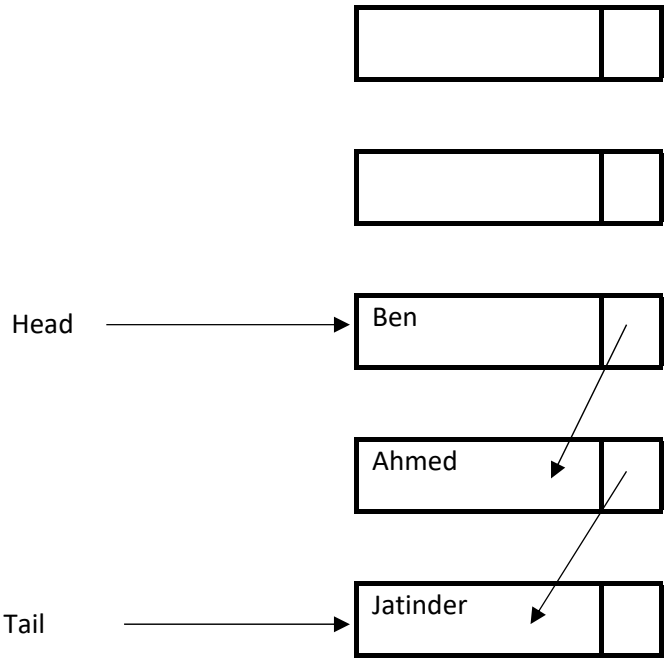
d)
```
OPEN myFile FOR READ
REPEAT
   READFILE myFile, StudentRecord
   OUTPUT StudentRecord
UNTIL EOF (myFile)
CLOSEFILE(myFile)
```

5 a) outer loop is executed 9 times
inner loop is executed 9 times (for each iteration of the outer loop)
not dependant on the dataset

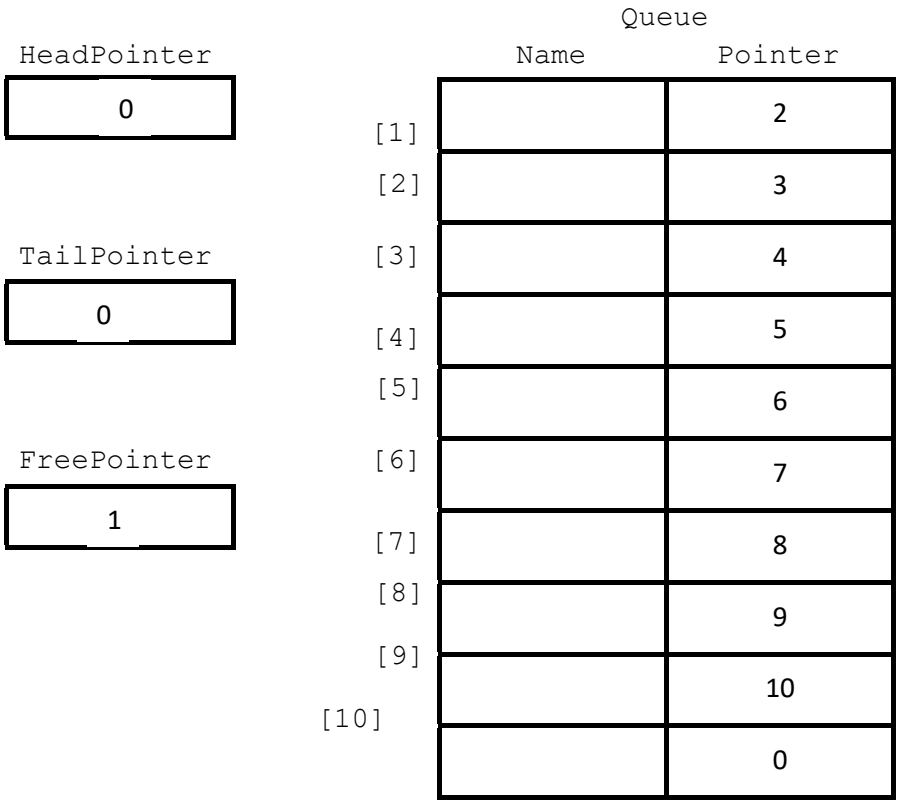
ii)

```
NumberOfItems ← 10
REPEAT
NoMoreSwaps ← TRUE

FOR Pointer ← 1 TO NumberOfItems – 1

    IF NameList[Pointer] > NameList[Pointer + 1] THEN
            NoMoreSwaps ← FALSE
            Temp ← NameList[Pointer]
            NameList[Pointer] ← NameList[Pointer + 1]
            NameList[Pointer + 1] ← Temp
        ENDIF
      NEXT
   NumberOfItems ← NumberOfItems - 1
   UNTIL NoMoreSwaps = TRUE
```

6 a)



b) i)



ii)

```
PROCEDURE RemoveName()
    // Report error if Queue is empty
    IF HeadPointer = 0
        THEN
            Error
        ELSE
            OUTPUT Queue[HeadPointer].Name
            // current node is head of queue
            CurrentPointer ← HeadPointer
            // update head pointer
            HeadPointer ← Queue[CurrentPointer].Pointer
            //if only one element in queue,then update tail p
            IF HeadPointer = 0
                THEN
                    TailPointer ← 0
            ENDIF
                // link released node to free list
            Queue[CurrentPointer].Pointer ← FreePointer
            FreePointer ← CurrentPointer

    ENDIF
ENDPROCEDURE
```