Sorting and merging sequences of student records

```
/* Comments - Reads a file containing a sequence of records
 *            representing students and places them into an
 *            array table1. It does the same for a second file
 *            placing result in table2. It then sorts table1
 *            and table2 in non descending order on the basis
 *            of their name fields. It now stores the merge of
 *            the two sorted sequences in an array table3.
 *            Finally it writes the result to a third file.
 *            The names of the three files are passed as
 *            parameters in the main function
 */


/* Comment – Define a data type with the following elements in it      */
/*           name, midterm marks, final marks, homework marks          */
/*           The data type is named 'student'                          */

Define data type student {
                Name,
                Midterm,
                Final,
                Homeworks
}


/* Comment - filename is the name of the file where we will write.     */
/*           Array a[] is of type student - contains the student records */

Function writeStudentArray (filename, Array a[])

        Assign i=0
        Assign n = size of a[]

        Open file filename in write mode
        If array size of a[] = 0 Then
                Exit Function
        End If

        For i = 0 to i < n
                Write to filename the elements a[i]->name, midterm, final, homeworks
                Assign i = i + 1
        Loop
        End For
        Close file filename
```

End Function

/* Comment - It reads all the student records */
/* from file filename and stores them in an array named a[] of type student. */
/* It returns the number of records actually read. */
/* */

Function readStudentArray (**filename**, Array **a**[])

Assign **i**=0

/* Comment – Read the entire file line by line */
Open file **filename** in read mode

Do While not **End of File**
Read the line number **i** of filename
Write the elements of the line to **a[i]->name, midterm, final, homeworks**
Assign **i=i**+1
Loop

Close file **filename**

End Function

/* Comment - It sorts in nondecreasing order on the basis of their */
/* names the first n records of table. */
/* */

Function sortStudentArray (student **table**[])

Sort in ascending order the records in **table**[] on the basis of **table[]->Name**

}

/* Comment - It merges into table3 all the student records of array table1 */
/* and then the entire student records of array table2. The records in */
/* table1 and table2 are sorted in non decreasing order of their */
/* name fields. The arrays table1 and table2 are of type student */
/* */

Function mergeStudentArray (Array **table3**[], Array **table1**[],
Array **table2**[])

```
        Assign i=0
        Assign j=0
        Assign n1 = size of table1[]
        Assign n2 = size of table2[]

        For i = 0 to i < n1
                Assign table3[j] = table1[i]
                Assign j=j+1
                Assign i=i+1
        Loop
        End For

        For i = 0 to i < n2
                Assign table3[j] = table2[i]
                Assign j=j+1
                Assign i=i+1
        Loop
        End For

End Function

/* Comment – Program starts in this function                    */
 /*              main function calls the other functions.       */
 /*                                                             */

Function main (filename1, filename2, filename3)

        /* Comment – Number of students in each sequence */
        Assign SIZE=25

        /* Comment - Students in first sequence */
        Define Array table1[SIZE] of type student
        /* Comment - Students in second sequence */
        Define Array table2[SIZE] of type student
        /* Comment - Students in merged sequence */
        Define Array table3[SIZE+SIZE] of type student

        Invoke function readStudentArray (filename1, table1[])
        Invoke function sortStudentArray (table1[])
        Invoke function readStudentArray (filename2, table2[])
        Invoke function sortStudentArray (table2[])
        Invoke function mergeStudentArray (table3[], table1[], table2[])
        Invoke writeStudentArray (filename3, table3[])

End Function
```