To get the extra credit for this assignment:

Step 1: Go to: http://ccse.kennesaw.edu/fye/pseudocode/ProblemSolvingGuide.pdf and read the guide thoroughly. First, you have to understand the process. Then, you will try to improve the pseudocode instructions suggested in the guide. After that, you will be ready to work on your assignment 8. At the end, you will have an opportunity to reflect on your experience.

To receive extra credit, you must show your thought process in step 2 and reflect on your experience in step 3.

Step 2: To complete Assignment 8 pseudocode based on the guide, use the following template to write your answers:

Solution for Problem 1

The problem:

I think the first problem is that we need to find whether two two-dimensional arrays are equivalent or not, i.e, we need to find whether the arrays contain same elements no matter what the order is.

Initial Plan:

Take two two-dimensional arrays as input.

Check whether the elements of both the arrays are equal or not.

Write the output depending on whether the arrays are equal or not.

Execution and Evaluation:

Reading through the initial plan, we encounter some problems. Some of them are:

From where do we take arrays?

How do we check whether the elements are equal or not?

Where and how should I write the output?

Revised Plan:

First we write the MAIN method of the program

CREATE two 3-by-3 two-dimensional arrays of integer type.

PRINT “Enter 9 integers to store in first array: ”

```
READ user input for the integers
Store the input value in the array
PRINT “Enter 9 integers to store in second array: ”
READ user input for the integers
Store the user input in the array
PRINT “Array A:”
PRINT the first array
PRINT “Array B:”
PRINT the second array
CALL a method isEquivalent with arguments as first array and second array
IF return value of the methods isEquivalent is equal to true
    PRINT “Judgment: The arrays are equivalent”
ELSE
    PRINT “Judgment: The arrays are not equivalent”
End the MAIN method
Now we write method isEquivalent with return type as Boolean and parameters as arrays of type integer called arr1 and arr2
We define a FOR loop whic swap the elements if the next element of the first array or arr1 is greater than the current element of that array. In short, it sorts the array.
Again we define a FOR loop which swap the elements if the next element of the second array or arr2 is greater than the current element of that array. In short, it sorts the array.
CREATE a variable a of type Boolean
STORE false as its value
Now define two loops(one outer and one inner)
CHECK whether arr1 with index as value of inner and outer loop is equal to arr2 with the same indices.
If the condition is correct then set the value of variable a as true otherwise set the value of variable a as false.
Now end the FOR loops
RETURN a to the MAIN method
END the method isEquivalent
```

END the class

Execution and Evaluation:

When the program runs each step is executed as such:-

First the program creates two 3-by-3 two-dimensional arrays of integer data-type called arr1 and arr2

Now it prints “Enter 9 integers to store in first array:”

Then it takes input form the user as integers and store in the first array

Then, it prints “Enter 9 integers to store in second array:”

Then it takes input for the user as integers and store in the second array

Now it prints “Array A:”

It then prints the elements of the first array

The program then prints “Array B:”

It then prints the elements of the second array

Now, it calls the method isEquivalent and checks whether the return value is equal to true or not.

If it is true the program prints “Judgment: The arrays are equivalent”

If the condition is not true, the program prints “Judgment: The arrays are not equivalent”

Solution for Problem 2

The problem:

The main aim of the program is to print the hours worked by three employees in a week, find the day on which each employee worked maximum hours and to print the total hours and corresponding employee to that hour in ascending order.

Initial Plan:

Take a two-dimensional array as input.

Take the hours worked by each employee as a random number between 0 and 10.

Print the hours worked by each employee on each day.

Print the day on which each employee worked the maximum hours.

Print the total hours of each employee in ascending order.

Execution and Evaluation:

Reading through the initial plan, we encounter some problems. Some of them are:

From where do we take array?

From where should we take the hours of each employee?

How should we find the day on which the employee worked maximum hours?

How should we total the hours of each employee?

Where and how should I write the output?

Revised Plan:

First we write the MAIN method of the program

CREATE a 3-by-7 two-dimensional array called Hour of integer type.

READ random integer between 0 and 10

Store the random value in the array Hours

PRINT “Employees Data: ”

PRINT “   Mon   Tue   Wed   Thu   Fri   Sat   Sun”

PRINT “Employee1   ”

CREATE a FOR loop that CREATES a variable i that executes the loop 7 times with incrementation each time

    Each time the loop the executed PRINT the array value stored in row number 0 and column number same as the value of l.

END FOR loop

PRINT “Employee2   ”

Now, again CREATE a FOR loop that CREATES a variable l that executes the loop 7 times with incrementation each time

    Each time the loop the executed PRINT the array value stored in row number 1 and column number same as the value of l.

END FOR loop

PRINT “Employee3   ”

Now again CREATE a FOR loop that CREATES a variable l that executes the loop 7 times with incrementation each time

Each time the loop the executed PRINT the array value stored in row number 2 and column number same as the value of l.

END FOR loop

PRINT a line

PRINT a line

CALL method addHours with argument as array Hours

End the MAIN method

Now we write method namedaddHours with no return typeand parameters as array of type integer called hours

CREATE an array a of data type integer and size 7

CREATE an array b of data type integer and size 7

CREATE an array c of data type integer and size 7

CREATE an array days of data type String and size 7

STORE “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday”, “Sunday” in array called days

CREATE variables add1, add2 and add3 of integer data type with initial values as 0

Inside the for loop we STORE the value of array hours with row 0 and column as i in array a with index i

Also, we add the value of array hours with row as 0 and column as i to the initial value of variable add1 and STORE the resultant in variable add1.

Then END FOR loop

Again, We define a FOR loop which CREATE a variable i with initial value as 0 and it runs till 7 and increments the value each time the loop is executed.

Inside the for loop we STORE the value of array hours with row 1 and column as i in array b with index i

Also, we add the value of array hours with row 1 and column as i to the initial value of variable add2 and STORE the resultant in variable add2.

Then END FOR loop

Again, We define a FOR loop which CREATE a variable i with initial value as 0 and it runs till 7 and increments the value each time the loop is executed.

```
        Inside the for loop we STORE the value of array hours with row 2 and column as i in array c with index i
        Also, we add the value of array hours with row 2 and column as i to the initial value of variable add3 and STORE the resultant in variable add3.
Then END FOR loop
CREATE  array sum of integer data type and size 3
STORE the value of add1 in array sum with index 0
STORE the value of add2 in array sum with index 1
STORE the value of add3 in array sum with index 2
CREATE variables max1, max2, max3 with integer data-type
STORE the value of array a with index 0 in variable max1
STORE the value of array b with index 0 in variable max2
STORE the value of array c with index 0 in variable max3
CREATE variables index1, index2 index3 of integer data-type and store 0 as initial value
We define a FOR loop which CREATE a variable i with initial value as 0 and it runs till 7 and increments the value each time the loop is executed.
         Inside the loop we define an IF statement.
         It has a condition which checks whether the value of array a with i is greater than max1
            If the condition is true STORE the value of array a with index i to max1
            Also, STORE the value of i to variable index1
       END IF loop
END FOR loop
We define a FOR loop which CREATE a variable i with initial value as 0 and it runs till 7 and increments the value each time the loop is executed.
         Inside the loop we define an IF statement.
         It has a condition which checks whether the value of array b with i is greater than max
            If the condition is true STORE the value of array b with index i to max2
            Also, STORE the value of i to variable index2
       END IF loop
END FOR loop
```

```
We define a FOR loop which CREATE a variable i with initial value as 0 and it runs till 7 and increments the value each time the loop is executed.

     Inside the loop we define an IF statement.

     It has a condition which checks whether the value of array c with i is greater than max3

       If the condition is true STORE the value of array c with index i to max

       Also, STORE the value of i to variable index3

    END IF loop

END FOR loop

PRINT “Employee1 worked most hours on”  days with index as value of variable index1

PRINT “Employee2 worked most hours on”  days with index as value of variable index2

PRINT “Employee3 worked most hours on”  days with index as value of variable index3

PRINT a line

PRINT “Employee#      Weekly  Hours”

We define a FOR loop which CREATE a variable i with initial value as 0 and runs till 2 and increments the value each time the loop is executed.

      Inside the loop we define another for loop which CREATE a variable j with initial value as 1 and runs till 3 and increments the value each time the loop is executed

           Inside the loop we check whether the value of array sum with i is greater than the value of array sum with index j using IF statement

                IF the condition is true store the value of array sum with index j in variable temp with data type integer

                 STORE the value of array sum with index i in array sum with index j

                STORE the value of variable temp in array sum with index i

                END IF statement

       END FOR statement

   END FOR statement

IF value of array sum with index 0 is equal to add1

      PRINT “     1      ”   value of array sum with index 0

ELSE IF value of array sum with index 0 is equal to add2

      PRINT “     2     ”   value of array sum with index 0
```

```
ELSE
          PRINT “    3     ”  value of array sum with index 0
END IF statement

 IF value of array sum with index 1 is equal to add3
          PRINT “    3     ”   value of array sum with index 1
ELSE IF value of array sum with index 1 is equal to add2
          PRINT “    2    ”   value of array sum with index 1
ELSE
         PRINT “    1    ”  value of array sum with index 1
END IF statement

IF value of array sum with index 2 is equal to add2
          PRINT “    2     ”   value of array sum with index 2
ELSE IF value of array sum with index 2 is equal to add1
          PRINT “    1   ”   value of array sum with index 2
ELSE
         PRINT “    3    ”  value of array sum with index 2
END IF statement
END the method addHours
END the class
```

Execution and Evaluation:

When the program runs each step is executed as such:-

First the program creates a 3-by- two-dimensional array of integer data-type called Hours

Then it takes random integers between 0 and 10 and stores in the array called Hours

Now it prints “Employees Data: ”

It then PRINT “   Mon   Tue   Wed   Thu   Fri   Sat   Sun”

It then PRINT “Employee1   ”

It then prints the hours worked by employee1

PRINT “Employee2   ”

It then prints the hours worked by employee2

PRINT “Employee3   ”

It then prints the hours worked by employee3

Now, it calls the method addHours

This method first add the hours worked by each employee and stores it in variable add1, add2 and add3

It then finds the day on which each employee worked maximum hours.

Finally, it prints the day on which employee worked the most hours

At last, it prints the total hours worked by each employee in ascending order.

Step 3: Complete the evaluation below:

Q1: Did the pseudocode exercise help you to understand the requirements and solve the problem faster?

- ○ Yes
- ✓ No
- ○ Other

Q2: Do you think that you could have fewer challenges with the assignments if this guide was introduced to you at the beginning of the semester?

- ○ Yes
- ✓ No
- ○ Other

Q3: What did you like about this pseudocode guide and the iterative method of solving problems?

Your Response: This guide is good. It helps us to understand the program properly.

Q4: What can be improved about this pseudocode guide?

Your response: I think this guide is just a waste of time. It do helps but it is too time consuming to do this sheet as well as do pseudocdes and source codes. We have a lot of other work too. But, specifically to improve this process, you would have reduced the topics to just 3 and removed re-execution of code. Wwe can write the whole code in initial plan.