**Solution of Pre release material june 2018**

Concept and Understanding of Tasks

All 3 tasks are a part of one big problem i.e. calculating the total and average volume (yield) of a herd of cows with the identification of most and least milk producing cows in the herd. The calculations need to done according to one-week data. The tasks are built incrementally and each task uses the code/algorithm of the previous task. Hence the general flow and explanation of each task is provided separately in the following diagram.

***Task 1***

Variables and constants declaration

Input and validate the 3 digit identity code of cow

Input the yield of milk from each cow(twice per day) for a week (7 days)

***Task 2***

*\* Task 1 code will be used again in this task with modifications*

Variables and constants declaration

Calculation of total yield of entire herd (of 7 days data)

Calculation of average yield per cow (of 7 days data)

Output both the total yield and average yield per cow to the nearest whole number

***Task 3***

*\* Task 2 code will be used again in this task with modifications*

Variables declaration

Calculate and output the identity code and yield of the cow which produced most milk in a week

Calculate and output the identity codes only, of the cows which produced less then 12 litres of milk on 4 or more days

Features of this Pre-Release

* Use of arrays data structure
* Use of nested loops
* Use of built-in function **INT**
* Overall Complexity: **Easy**

Approach to Solution

Like every algorithm, there can be many possible approaches to solve these tasks depending upon the understanding of person. We are listing down the key points that reflect our understanding and we’ll solve these tasks according to following **assumptions**.

* In Task1, the size of herd is a **constant**. Hence it will be declared as a constant value and will not be taken as an input from user.
* In Task1, the identity code will be validated using **length check**. It is assumed that the user is entering **unique identity codes** and no validation will be performed on it.
* In Task2, the calculation of total and average yield per cow will be calculated to **nearest whole liter** by using mathematical calculations. It should be noted that in formal programming languages like Python, VB or Pascal, there are built-in functions (ROUND functions) to do this. We will use the calculations which keep this solution compatible to pseudocode.
* In Task3, the highest weekly yield of cow will be output directly without converting it to nearest whole number as it is not asked in the pre release.

Explanation of Algorithm of Tasks

The explanation of the algorithms used in each task is listed below.

Task 1

In this task we will first input the identity codes of the cows using a FOR loop. The identity codes will be stored in an array and validation will be performed using a REPEAT loop to check that it should be of 3-digits (length check). A 3-digit numeric code means that it will start at 100 and ends at 999.

Once all identity codes are entered, we will then start a nested loop which will input daily yield of each cow for a week. The first FOR loop will run 7 times because we are taking input for 7 days, then we need another FOR loop will run according to the number of cows in the herd.

Since it is clearly given in the pre-release that each cow can be milked twice per day, so we will directly input the yield in 2 variables. It is to be noted that daily yield will not be stored in an array rather the total of each cow’s daily yield will be stored in an array. This is because we need this total value in task2 and task3. (see figure below for further clarification)

Task 2

In this task, we need to display the average yield of milk per cow which is easy to calculate because we already have the total yield of each cow already stored in an array. We will simply divide it by total number of cows in our herd to get the average of each cow.

For weekly total volume, we will once again use the same array and calculate its total using a FOR loop.

The only challenge is to display the output to the nearest whole number i.e. round figure. For this, we will use the built-in function “INT” which allows us to extract whole number value from a fractional number.

**For example:**

If we use INT function on the value 32.6 then it will give us the answer 32. It eliminates the factional part from any number. Remember that it **doesn’t** round off the number, it just trims the fractional part!

*(see past paper Q.3 of Oct/Nov 2016 variant 23)*

Task 3

In this task we have to find the identity code number of the cow which has the highest yield in a week and the identity code numbers of cows whose yield was less than 12 liters on 4 or more days in a week. To find the highest yield we will once again use the total yield array and find the highest value in it. Since this task doesn’t ask us to output the result in nearest whole number, we will simple print it as is.

To find the cows with low yield, we will modify Task1 and add an IF statement during input to check if the yield (sum of 2 yields) is less than 12 liters. We will use another array specially for this purpose and simply add 1 into it. In the end we will output all cow ids which have the value of 4 or more into it.

Task 1 Solution (Pseudocode)

*\* The herd size is assumed to be 5.*

**BEGIN**

**CONST** *herdsize* **←** *5* **AS INTEGER**

**DECLARE** *idcode [1:5], cntdays* **←** *0, cntcows* **←** *0* **AS INTEGER**

**DECLARE** *yield1* **←** *0.0, yield2* **←** *0.0, cowtotalyield [1:5]* **AS REAL**

**FOR** *cntcows* = *1* **TO** *herdsize*

REPEAT

**PRINT** "Enter ID code for cow number ", cntcows

**INPUT** *idcode[cntcows]*

**IF** *idcode[cntcows] < 100* **OR** *idcode[cntcows] > 999* **THEN**

**OUTPUT** "ID code should be of 3 digits"

**ENDIF**

**UNTIL** *(idcode[cntcows] >= 100)* **AND** *(idcode[cntcows] <= 999)*

**NEXT** *cntcows*

FOR cntdays = 1 TO 7

**PRINT** "Enter herd yields for weekday", *cntdays*

**FOR** *cntcows* = *1* **TO** *herdsize*

**PRINT** “Enter first yield of the day for cow ID”, *idcode[cntcows]*

**INPUT** *yield1*

**PRINT** “Enter second yield of the day for cow ID”, *idcode[cntcows]*

**INPUT** *yield2*

*cowtotalyield[cntcows]* **←***cowtotalyield[cntcows] + yield1 + yield2*

**NEXT** *cntcows*

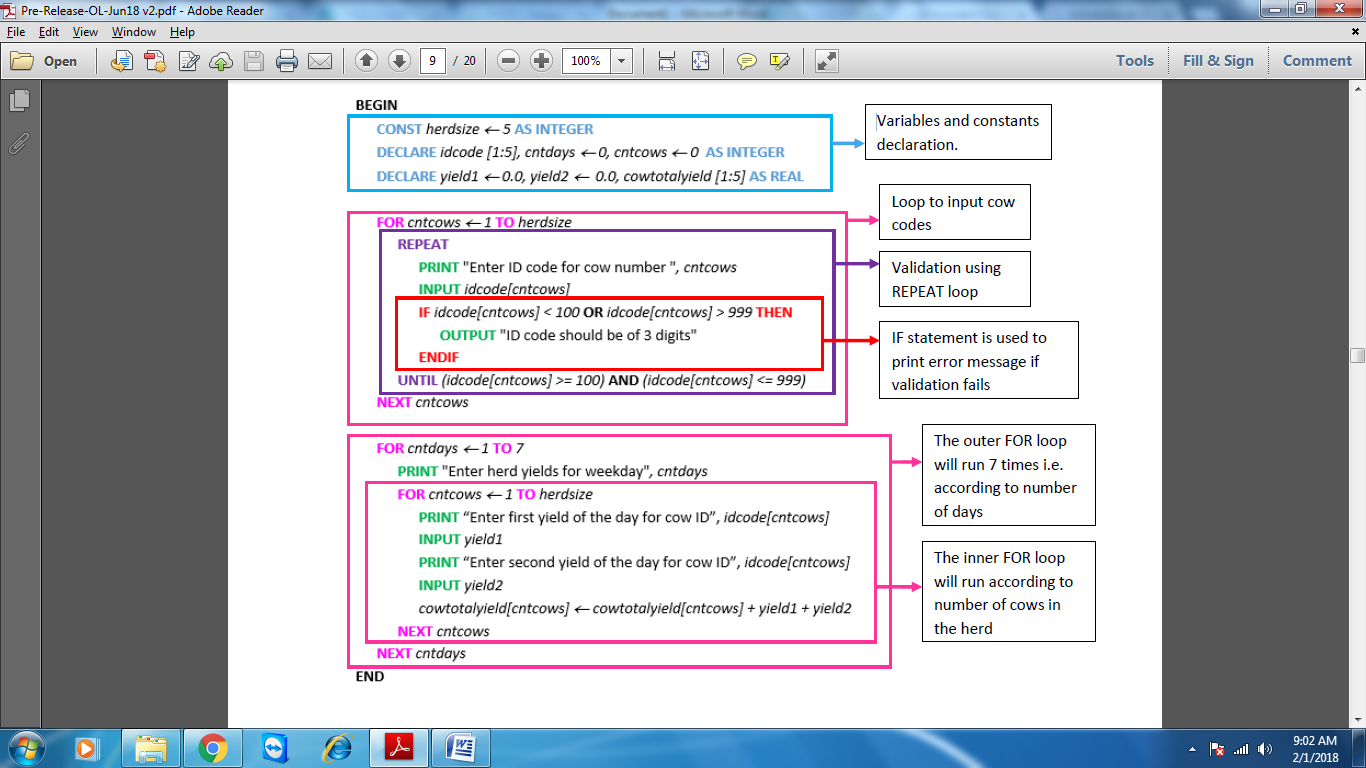
**NEXT** *cntdays*

**END**

**Efficiency of Algorithm**

* Use of **CONSTANT** to hold fixed value of herd size
* Use of ARRAY to store identity code of cows
* Uses **IF** statement to print appropriate error message when validation fails
* Uses **nested FOR** statements to input milk yield for all cows for a week

Explanation of Task 1



Task 2 Solution (Pseudocode)

*\* This task is the extension of Task1 and includes entire pseudocode of Task1*

**BEGIN**

**CONST** *herdsize* **←** *5* **AS INTEGER**

**DECLARE** *idcode [1:5], cntdays* **←** *0, cntcows* **←** *0* **AS INTEGER**

**DECLARE** *yield1* **←** *0.0, yield2* **←** *0.0, cowtotalyield [1:5], cowavgyield* **←** *0.0,*

*weeklyvol* **←** *0.0, temp* **←** *0.0* **AS REAL**

**FOR** *cntcows* = *1* **TO** *herdsize*

REPEAT

**PRINT** "Enter ID code for cow number "*, cntcows*

**INPUT** *idcode[cntcows]*

**IF** *idcode[cntcows] < 100* **OR** *idcode[cntcows] > 999* **THEN**

**OUTPUT** "ID code should be of 3 digits"

**ENDIF**

**UNTIL** *(idcode[cntcows] >= 100)* **AND** *(idcode[cntcows] <= 999)*

**NEXT** *cntcows*

FOR cntdays = 1 TO 7

**PRINT** "Enter herd yields for Weekday:", *cntdays*

**FOR** *cntcows* =*1* **TO** *herdsize*

**PRINT** “Enter first yield of the day for cow ID”, *idcode[cntcows]*

**INPUT** *yield1*

**PRINT** “Enter second yield of the day for cow ID”, *idcode[cntcows]*

**INPUT** *yield2*

*cowtotalyield[cntcows]* **←***cowtotalyield[cntcows] + yield1 + yield2*

**NEXT** *cntcows*

**NEXT** *cntdays*

**FOR** *cntco*ws **=** 1 **TO** *herdsize*

*cowavgyield* **←** *cowtotalyield[cntcows] / herdsize*

*temp* **←** *cowavgyield – INT(cowavgyield)*

**IF** *temp <= 0.4* **THEN**

*cowavgyield* **←** *INT(cowavgyield)*

**ELSE** *cowavgyield* **←** *INT(cowavgyield) + 1*

ENDIF

**PRINT** "Avg yield of cow ID"*, code[cntcows],* " is "*, cowavgyield,* " liters"

*weeklyvol* **←** *weeklyvol + cowtotalyield[cntcows]*

**NEXT** *cntcows*

*temp* **←** *weeklyvol – INT(weeklyvol)*

**IF** *temp <= 0.4* **THEN**

*weeklyvol* **←***INT(weeklyvol)*

**ELSE** *weeklyvol* **←** *INT(weeklyvol) + 1*

END

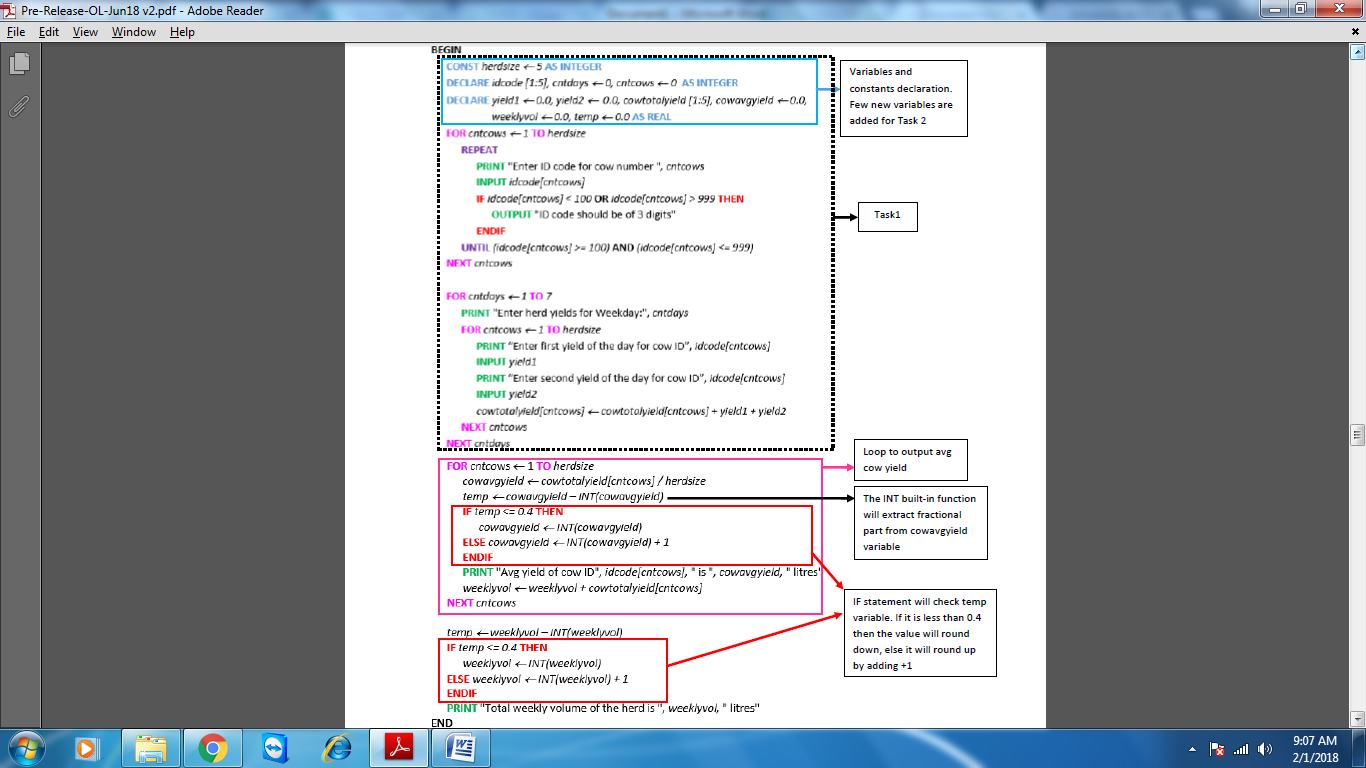
**PRINT** "Total weekly volume of the herd is "*, weeklyvol,* " liters"

**END**

**Efficiency of Algorithm**

* Use of **CONSTANT** to hold fixed value of herd size
* Use of ARRAY to store identity code of cows
* Uses **IF** statement to print appropriate error message when validation fails
* Uses **nested FOR** statements to input milk yield for all cows for a week
* Use of INT built-in function to perform round up and round down of fractional value

Pseudocode Explanation of Task 2



Task 3 Solution (Pseudocode)

*\* This task includes both Task1 and Task2 code hence it is the complete pseudocode of entire pre-release*

**BEGIN**

**CONST** *herdsize* **←** *5* **AS INTEGER**

**DECLARE** *idcode [1:5], cntdays* **←** *0, cntcows* **←** *0, lessyield[1:5], highidcode* **←** *0* **AS INTEGER**

**DECLARE** *yield1* **←** *0.0, yield2* **←** *0.0, cowtotalyield [1:5], cowavgyield* **←** *0.0,*

*weeklyvol* **←** *0.0, temp* **←** *0.0, highestyield* **←** *-99.0* **AS REAL**

**FOR** *cntcows* = *1* **TO** *herdsize*

REPEAT

**PRINT** "Enter ID code for cow number ", cntcows

**INPUT** *idcode[cntcows]*

**IF** *idcode[cntcows] < 100* **OR** *idcode[cntcows] > 999* **THEN**

**OUTPUT** "ID code should be of 3 digits"

**ENDIF**

**UNTIL** *(idcode[cntcows] >= 100)* **AND** *(idcode[cntcows] <= 999)*

**NEXT** *cntcows*

FOR cntdays = 1 TO 7

**PRINT** "Enter herd yields for Weekday:", *cntdays*

**FOR** *cntcows* = *1* **TO** *herdsize*

**PRINT** “Enter first yield of the day for cow ID”, *idcode[cntcows]*

**INPUT** *yield1*

**PRINT** “Enter second yield of the day for cow ID”, *idcode[cntcows]*

**INPUT** *yield2*

*cowtotalyield[cntcows]* **←** *cowtotalyield[cntcows] + yield1 + yield2*

**IF** *(yield1 + yield2) < 12.0* **THEN**

*lessyield[cntcows]* **←** *lessyield[cntcows] + 1*

ENDIF

**NEXT** *cntcows*

**NEXT** *cntdays*

**FOR** *cntco*ws **=** 1 **TO** *herdsize*

*cowavgyield* **←** *cowtotalyield[cntcows] / herdsize*

*temp* **←** *cowavgyield – INT(cowavgyield)*

**IF** *temp <= 0.4* **THEN**

*cowavgyield* **←** *INT(cowavgyield)*

**ELSE** *cowavgyield* **←** *INT(cowavgyield) + 1*

ENDIF

**PRINT** "Avg yield of cow ID"*, idcode[cntcows],* " is "*, cowavgyield,* " liters"

*weeklyvol* **←** *weeklyvol + cowtotalyield[cntcows]*

**NEXT** *cntcows*

*temp* **←***weeklyvol – INT(weeklyvol)*

**IF** *temp <= 0.4* **THEN**

*weeklyvol* **←***INT(weeklyvol)*

**ELSE** *weeklyvol* **←** *INT(weeklyvol) + 1*

ENDIF

**PRINT** "Total weekly volume of the herd is "*, weeklyvol,* " liters"

**FOR** *cntcows* **=** *1* **TO** *herdsize*

**IF** *lessyield[cntcows] >= 4* **THEN**

**PRINT** "Cow code ", *idcode[cntcows],* " produced less than 12 liters"

**ENDIF**

**IF** *cowtotalyield[cntcows] > highestyield* **THEN**

*highestyield* **←** *cowtotalyield[cntcows]*

*highidcode* **←** *idcode[cntcows]*

**ENDIF**

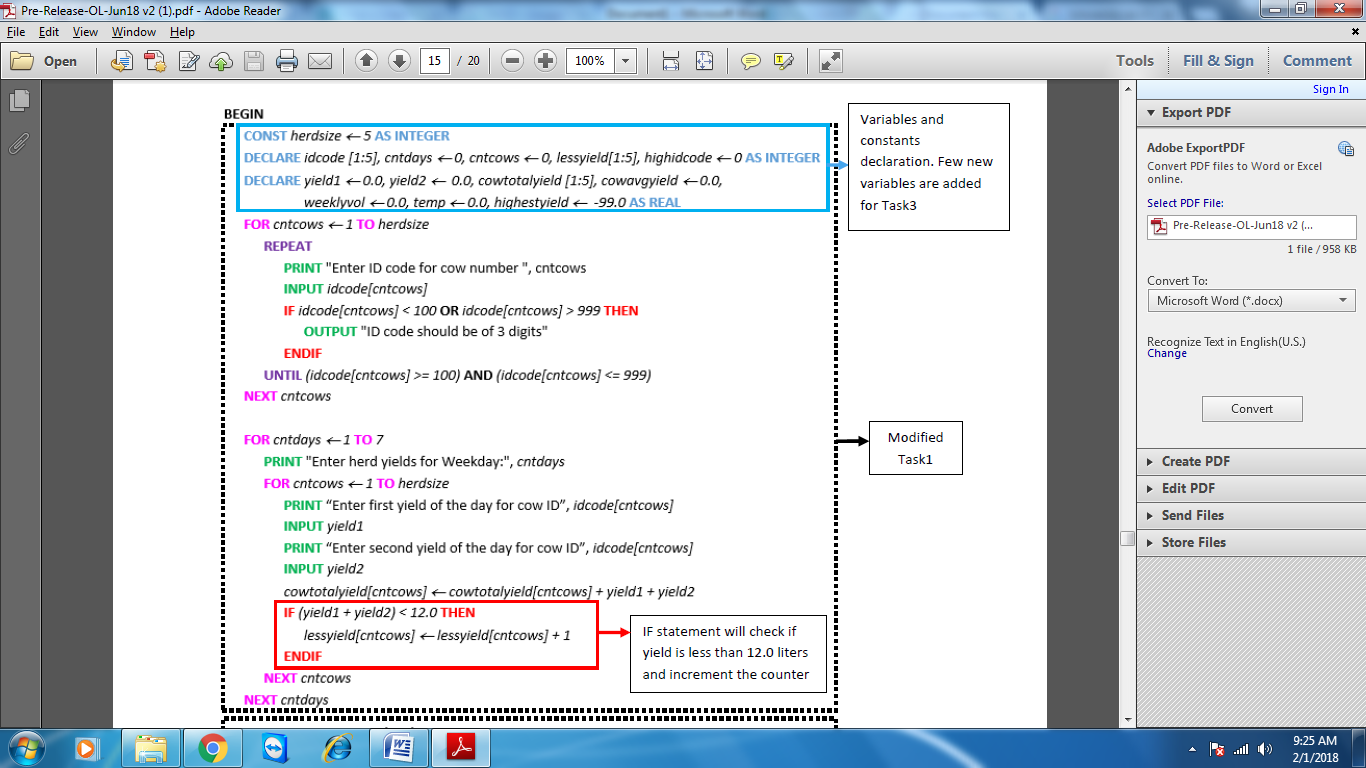
**NEXT** *cntcows*

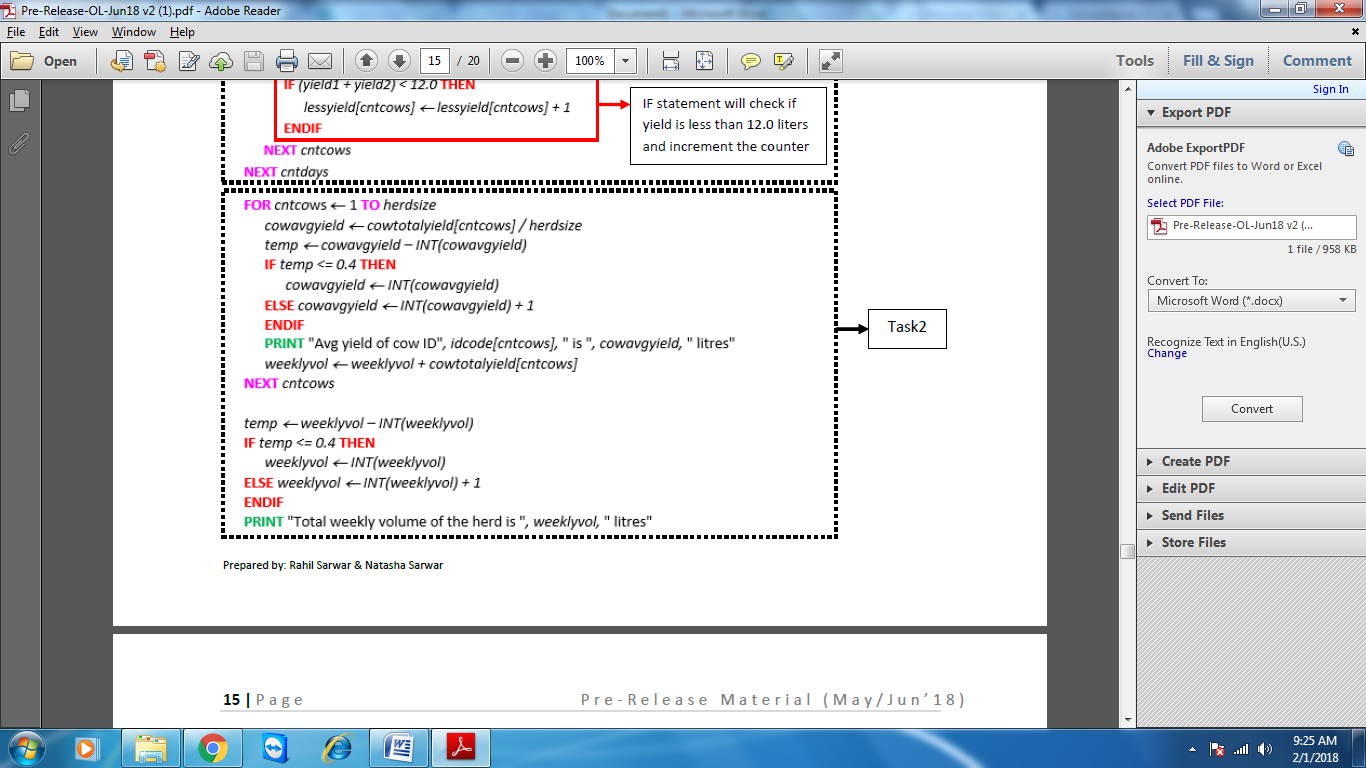
**PRINT** "The highest yield is produced by cow code", *highidcode*

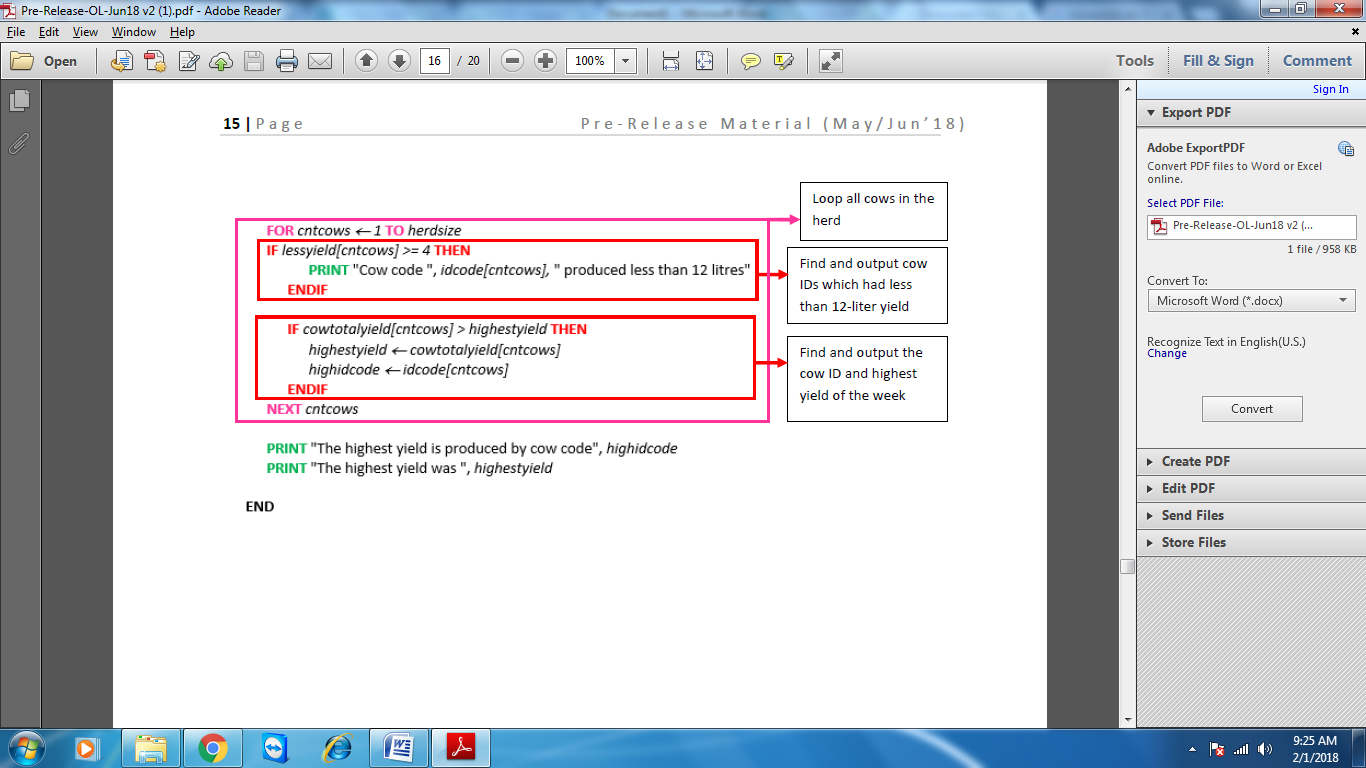
**PRINT** "The highest yield was ", *highestyield*

**END**

Pseudocode Explanation of Task 3







VB .NET Program Code of all Tasks

*\* In VB.NET, array indexes start at 0 not 1. That’s why the FOR loop counter starts with 0 in the following code.*

Module Module1

Sub Main()

Const herdsize As Integer = 5

Dim idcode(4), cntdays, cntcows, lessyield(4), highidcode As Integer

Dim yield1, yield2, cowtotalyield(4), cowavgyield, weeklyvol, temp, highestyield As Double

cntdays = 0

cntcows = 0

highidcode = 0

yield1 = 0.0

yield2 = 0.0

cowavgyield = 0.0

weeklyvol = 0.0

temp = 0.0

highestyield = -99.0

For cntcows = 0 To herdsize - 1

Do

Console.WriteLine("Enter ID code for cow number " & cntcows)

idcode(cntcows) = Console.ReadLine()

If idcode(cntcows) < 100 Or idcode(cntcows) > 999 Then

Console.WriteLine("ID code should be of 3 digits")

End If

Loop Until (idcode(cntcows) >= 100) And (idcode(cntcows) <= 999)

Next cntcows

For cntdays = 1 To 7

Console.WriteLine("Enter herd yields for Weekday:" & cntdays)

For cntcows = 0 To herdsize - 1

Console.WriteLine("Enter first yield of the day for cow ID " & idcode(cntcows))

yield1 = Console.ReadLine()

Console.WriteLine("Enter second yield of the day for cow ID " & idcode(cntcows))

yield2 = Console.ReadLine()

cowtotalyield(cntcows) = cowtotalyield(cntcows) + yield1 + yield2

If (yield1 + yield2) < 12.0 Then

lessyield(cntcows) = lessyield(cntcows) + 1

End If

Next cntcows

Next cntdays

For cntcows = 0 To herdsize - 1

cowavgyield = cowtotalyield(cntcows) / herdsize

temp = cowavgyield - Math.Truncate(cowavgyield)

If temp <= 0.4 Then

cowavgyield = Math.Truncate(cowavgyield)

Else

cowavgyield = Math.Truncate(cowavgyield) + 1

End If

Console.WriteLine("Avg yield of cow ID" & idcode(cntcows) & " is " & cowavgyield & " liters")

weeklyvol = weeklyvol + cowtotalyield(cntcows)

Next cntcows

temp = weeklyvol - Math.Truncate(weeklyvol)

If temp <= 0.4 Then

weeklyvol = Math.Truncate(weeklyvol)

Else

weeklyvol = Math.Truncate(weeklyvol) + 1

End If

Console.WriteLine("Total weekly volume of the herd is " & weeklyvol & " liters")

For cntcows = 0 To herdsize - 1

If lessyield(cntcows) >= 4 Then

Console.WriteLine("Cow code " & idcode(cntcows) & " produced less than 12 liters")

End If

If cowtotalyield(cntcows) > highestyield Then

highestyield = cowtotalyield(cntcows)

highidcode = idcode(cntcows)

End If

Next cntcows

Console.WriteLine("The highest yield is produced by cow code : " & highidcode)

Console.WriteLine("The highest yield was : " & highestyield)

End Sub

End Module