**Validation:**

1. **Days of week**

\*In the Function TakeRecording under the line TakeRecording.day = ConvertDay(Console.ReadLine) add:

Do While TakeRecording.day = 7

Console.WriteLine("That is not a valid day. Please enter the day: mon, tue, etc")

TakeRecording.day = ConvertDay (Console.ReadLine)

Loop

\*In the Function ConvertDay change Case Else Return 1 to:

Case Else

Return 7

1. **Yield is to one decimal place**

\*In the Function TakeRecording create a variable:

Dim newVal As Single

\*In the Function TakeRecording under the line TakeRecording.yield = Console.ReadLine add:

newVal = (((TakeRecording.yield \* 10) \ 1) / 10)

Do While newVal <> TakeRecording.yield

Console.WriteLine("Please enter a yield to one decimal place")

TakeRecording.yield = Console.ReadLine

Loop

1. **Yield entered is not a negative number**

\*In the Function TakeRecording under the line TakeRecording.yield = Console.ReadLine add:

Do While TakeRecording.yield < 0

Console.WriteLine("Please enter a yield greater than or equal to 0. Please try again.")

TakeRecording.yield = Console.ReadLine

Loop

**Allow all inputs to be valid:**

1. **Days of week**

\*In the Function TakeRecording add an LCase:

Console.WriteLine("Please enter Day mon, tue, etc (give the first 3 letters only)") TakeRecording.day = ConvertDay(LCase(Console.ReadLine))

\*In the Function TakeRecording in the while loop add an LCase:

Do While TakeRecording.day = 7

Console.WriteLine("That is not a valid day. Please enter the day: mon, tue, etc")

TakeRecording.day = ConvertDay(LCase(Console.ReadLine))

Loop

1. **Cow Id**

\*In the Function TakeRecording create two variables:

Dim input As Integer

Dim correct As Integer = 0

\*In the Function TakeRecording copy this:

Console.WriteLine("Please Enter the cow ID")

input = Console.ReadLine

For x = 1 To herdSize

If input = cows(x) Then

correct = 1

End If

Next

Do While correct <> 1

For x = 0 To herdSize

If input = cows(x) Then

correct = correct + 1

End If

Next x

If correct <> 1 Then

Console.WriteLine("the cow ID entered was invalid. Please re enter")

input = Console.ReadLine()

End If

Loop

TakeRecording.cowID = input

1. **Practice Program**

\*In the Sub main add this do loop

Do

Console.WriteLine("Do you want to run the practice program? 1 for yes and 0 for no")

userinput = Console.ReadLine()

Loop Until userinput = 0 Or userinput = 1

1. Herd Size

\*In the Sub main copy this

If input = 0 Or input = 1 Then

Console.WriteLine("Input herd size again. 0 or 1 is not a valid herd size. ")

Do

input = Console.ReadLine()

If input = 1 Or input = 0 Then

Console.WriteLine("Input herd size again. 0 or 1 is not a valid herd size. ")

End If

Loop Until input <> 0 And input <> 1

End If

**Kill the cow**

1. **Add case in sub and menu**

\*In the Sub main create 2 variables

Dim count As Integer = -1

Dim dead(0) As Integer

\*In the Sub main in the case section add this:

Case = "f”

count = count + 1

ReDim dead(count)

dead(count) = KillCow(cows)

\*In the Sub menu add this:

Console.WriteLine("f. Remove Cow")

1. **Write function**

\*Create a whole new function and copy this:

Function KillCow(ByVal cows() As Integer)

Dim death As Integer

Console.WriteLine("Which cow do you want to kill?")

death = Console.ReadLine()

Return death

End Function

1. **Edit displayCows Sub**

\*In the Sub display cows, add byVal Rip() as integer and add the y loop

Sub displayCows(ByVal cows() As Integer, ByVal rip() As Integer) ]

For x = 0 To cows.Length - 1 '

For y = 0 To rip.Length - 1

x = x + 1

End If

Next

Console.Write(cows(x) & " ")

Next

End Sub

\*In the Sub main in case a make sure you change it to:

displayCows(cows, dead)

**Clear data in weekly table**

\*In the Function End Week and this for loop at the end before returning year Data

For x = 0 To NumCows

For y = 0 To 13

milkings(x, y) = 0

Next

Next

**Make sure cow only milked twice a day**

\*In the Function Update Yield add this after the for loop

If milkings(location, newData.day \* 2) = 0 Then

milkings(location, (newData.day \* 2)) = newData.yield

ElseIf milkings(location, (newData.day \* 2) + 1) = 0 Then

milkings(location, (newData.day \* 2) + 1) = newData.yield

Else

Console.WriteLine("This cow has already been milked twice today. ")

End If

**Flip the table around for the yearly output so that cow Id is on top and week number is down the side**

\*In the Sub OutputYearlyData copy this:

Console.WriteLine()

Console.WriteLine("YEARLY WEEKLY TOTAL DATA")

Console.Write(" ")

For x = 0 To cows.Length - 1

Console.Write(cows(x) & " ")

Next

Console.WriteLine()

For x = 0 To 51

If x < 9 Then

Console.Write(" " & x + 1 & " ")

Else

Console.Write(" " & x + 1 & " ")

End If

For y = 0 To cows.Length - 1

Console.Write(milkingData(y, x) & " ")

For z = 0 To 2 - Len(CStr(milkingData(y, x))) Console.Write(" ")

Next

Next

Console.WriteLine()

Next

Console.WriteLine()

**Makes sure a unique cow ID has been input**

\*In the function input Cow IDs: add the word Randomize()at the beginning

Randomize()

\*In the function input Cow IDs: create two variables:

Dim uniqueID As Boolean = True

Dim currentCowId As Integer

\*In the function input Cow IDs: copy these loops:

For x = 0 To cows

Do

uniqueID = True

currentCowId = ((Rnd() \* 899) + 100)

For count = 1 To cows

If currentCowId = newCowIds(count) Then

uniqueID = False

End If

Next

Loop Until uniqueID

newCowIds(x) = currentCowId

Next

**Add data output to nearest whole number:**

\*In the sub main copy find these lines in case c and add math.round:

Console.WriteLine("Cow " & cows(x) & " average = " & Math.Round(Average(herdData, x)))

Console.WriteLine("Weekly total so far " & Math.Round(WeeklyTotal(herdData, herdSize)))

Console.WriteLine("Average so far this week " & Math.Round(total / (herdSize + 1)))

\*In the function end week find this line and add math.round:

Console.WriteLine("Cow " & herd(x) & " average = " & Math.Round((Average(milkings, x))))

\*In the function weekly total find this line and add math.round:

Return Math.Round(WeeklyTotal)