

PRM – Oct/Nov 2020 – P23

1 (a) Programming concept 1: Conditional Statements

Programming concept 2: Iteration

Note: These concepts can be written in any order.

Explanation:

- A post-conditional loop would run based on a flag
- At the end of the loop, the user is prompted whether or not to order another baguette, cancel, etc.
- Response stored in flag variable
- Conditional statement determines the response and switches the flag value accordingly

[3]

(b) Data structure: Array

The following is an example solution. There can be other correct solutions:

OUTPUT "Enter your choice [1-3]: "

INPUT breadChoice

IF breadChoice < 1 OR breadChoice > 3 THEN

 OUTPUT "Please enter a valid choice of bread [1-3]."

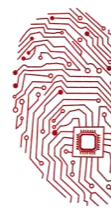
ELSE

 breadType[breadChoice] = breadType[breadChoice] + 1

END IF

[3]





PRM – Oct/Nov 2020 – P23

(c) Problem: The most popular and least popular choices could be assigned to either of the fillings // The algorithm wouldn't be able to decide which filling is most popular and which is least popular.

Code:

The following is an example solution. There can be other correct solutions:

```
IF mostFillings = leastFillings THEN
```

```
    OUTPUT "The " + mostilling[count] + "was ordered the same number of times as " +  
    leastFilling[count]"
```

```
ENDIF
```

[4]

(d) *The following is an example solution. There can be other correct solutions:*

```
moreOrder = TRUE
```

```
changeOrder = TRUE
```

```
WHILE moreOrder = TRUE
```

```
    WHILE changeOrder = TRUE
```

```
        OUTPUT "Baguette sizes:"
```

```
        OUTPUT "1. 15cm"
```

```
        OUTPUT "2. 30cm"
```

```
        sizeFlag = False
```

```
    WHILE sizeFlag = False
```

```
        OUTPUT "Enter your choice [1-2]: "
```

```
        INPUT sizeChoice
```

```
    IF sizeChoice < 1 OR sizeChoice > 2 THEN
```

```
        OUTPUT "Please enter a valid choice of size [1-2]."
```

```
    ELSE
```

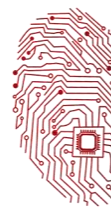
```
        baguetteSize[sizeChoice] = baguetteSize[sizeChoice] + 1
```

```
        sizeFlag = TRUE
```

```
    END IF
```

```
END WHILE
```





PRM – Oct/Nov 2020 – P23

```
// [Section start] Ask the user to input the type of bread
```

```
OUTPUT "Bread types:"
```

```
OUTPUT "1. White"
```

```
OUTPUT "2. Brown"
```

```
OUTPUT "3. Seeded"
```

```
// [Section end] Ask the user to input the type of bread
```

```
breadFlag = FALSE
```

```
WHILE breadFlag = FALSE
```

```
    OUTPUT "Enter your choice [1-3]: "
```

```
    INPUT breadChoice
```

```
    IF breadChoice < 1 OR breadChoice > 3 THEN
```

```
        OUTPUT "Please enter a valid choice of bread [1-3]."
```

```
    ELSE
```

```
        breadType[breadChoice] = breadType[breadChoice] + 1
```

```
        breadFlag = True
```

```
    END IF
```

```
END WHILE
```

```
OUTPUT "Baguette fillings:"
```

```
OUTPUT "1. Beef"
```

```
OUTPUT "2. Chicken"
```

```
OUTPUT "3. Cheese"
```

```
OUTPUT "4. Egg"
```

```
OUTPUT "5. Tuna"
```

```
OUTPUT "6. Turkey"
```

```
fillingFlag = FALSE
```

```
WHILE fillingFlag = FALSE
```

```
    OUTPUT "Enter your choice [1-6]: "
```

```
    INPUT fillingChoice
```

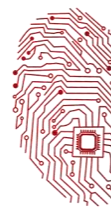
```
// [Section start] Validate choice of filling
```

```
IF fillingChoice < 1 OR fillingChoice > 6 THEN
```

```
    OUTPUT "Please enter a valid choice of filling [1-6]."
```

```
// [Section start] Validate choice of filling
```





PRM – Oct/Nov 2020 – P23

```
ELSE
    fillings[fillingChoice] = fillings[fillingChoice] + 1
    fillingFlag = TRUE
END IF
END WHILE

saladCount = 0

OUTPUT "Salad options:"
OUTPUT "1. Lettuce"
OUTPUT "2. Tomato"
OUTPUT "3. Sweetcorn"
OUTPUT "4. Cucumber"
OUTPUT "5. Peppers"

OUTPUT "Would you like to add salads to your baguette? (Y/N)"
INPUT addSalad

IF UPPER(addSalad) = "Y" THEN
    saladFlag = FALSE
    more = TRUE

    WHILE saladFlag = FALSE OR (saladCount < 3 AND more = TRUE)
        OUTPUT "You may choose up to" & (3 - saladCount) & " more
salad(s). Enter your next choice [1-5]: "

        INPUT saladChoice

        IF saladChoice < 1 OR saladChoice > 5 THEN

            OUTPUT "Please enter a valid choice of salad [1-5]."

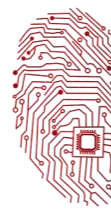
        ELSE

            salad[SaladChoice] = salad[SaladChoice] + 1
            saladFlag = TRUE
            saladCount = saladCount + 1
            saladChoiceList[saladCount] = saladChoice

            IF saladCount < 3 THEN
                OUTPUT "Would you like to add another salad?
(Y/N): "

                INPUT more
```





PRM – Oct/Nov 2020 – P23

```
IF UPPER(more) = "Y" THEN
    more = TRUE
ELSE
    more = FALSE
END IF
END IF
END IF
END WHILE
END IF
```

```
sizeDisplay = baguetteSizeName[sizeChoice]
bread = breadTypeName[breadChoice]
filling = fillingsName[fillingChoice]
```

```
OUTPUT "Your ordered baguette:"
OUTPUT ""
OUTPUT "Size:"
OUTPUT sizeDisplay
OUTPUT "Bread:"
OUTPUT bread
OUTPUT "Filling:"
OUTPUT filling
OUTPUT "Salad(s):"
```

```
FOR saladCount = 1 TO 3
    IF saladChoiceList[saladCount] <> "" THEN
        OUTPUT saladName[saladChoiceList[saladCount]]
    END IF
NEXT
```

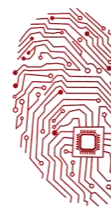
```
OUTPUT "Confirm order? (Y/N): "
INPUT confirmOrder
```

```
IF UPPER(confirmOrder) = "N" THEN
    confirmOrder = FALSE
    changeOrder = FALSE
    cancelOrder = FALSE
```

```
WHILE changeOrder = FALSE AND cancelOrder = FALSE
    OUTPUT "Would you like to change your order or cancel it?"
```

(change/cancel): "





PRM – Oct/Nov 2020 – P23

```
INPUT changeOrderChoice
IF UPPER(changeOrderChoice) = "CHANGE" THEN
    changeOrder = TRUE
ELSEIF UPPER(changeOrderChoice) = "CANCEL" THEN
    cancelOrder = TRUE
ELSE
    OUTPUT "Please enter a valid choice: "
END IF
END WHILE

ELSEIF UPPER(confirmOrder) = "Y" THEN
    changeOrder = FALSE
    cancelOrder = FALSE

    // [Section start] Calculate and display order number
    orderNo = orderNo + 1
    OUTPUT "Your order number is: " & orderNo
    // [Section start] Calculate and display order number

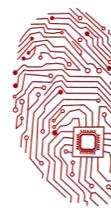
END IF
END WHILE

OUTPUT "Would you like to order another baguette? (Y/N): "
INPUT moreOrderChoice

IF UPPER(moreOrderChoice) = "Y" THEN
    moreOrder = TRUE
ELSE
    moreOrder = FALSE
END IF
END WHILE
```

[6]





PRM – Oct/Nov 2020 – P23

(e) and (f)

NOTE: (e) and (f) have the same example solutions. There are other correct solutions to these questions individually.

Pseudocode:

DECLARE totalSize15 : INTEGER

DECLARE totalSize30 : INTEGER

DECLARE totalBreadWhite : INTEGER

DECLARE totalBreadBrown : INTEGER

DECLARE totalBreadSeeded : INTEGER

DECLARE totalFillingBeef : INTEGER

DECLARE totalFillingChicken : INTEGER

DECLARE totalFillingCheese : INTEGER

DECLARE totalFillingEgg : INTEGER

DECLARE totalFillingTuna : INTEGER

DECLARE totalFillingTurkey : INTEGER

DECLARE totalBaguettesSold : INTEGER

totalBaguettesSold = orderNo

totalSize15 = baguetteSize[1]

totalSize30 = baguetteSize[2]

totalBreadWhite = breadType[1]

totalBreadBrown = breadType[2]

totalBreadSeeded = breadType[3]

totalFillingBeef = fillings[1]

totalFillingChicken = fillings[2]

totalFillingCheese = fillings[3]

totalFillingEgg = fillings[4]

totalFillingTuna = fillings[5]

totalFillingTurkey = fillings[6]

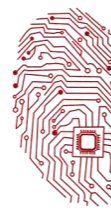
DECLARE mostFilling : INTEGER

DECLARE leastFilling : INTEGER

DECLARE mostPopularPerc : REAL

DECLARE leastPopularPerc : REAL





PRM – Oct/Nov 2020 – P23

```
DECLARE mostPopularName : STRING = ""
```

```
DECLARE leastPopularName : STRING = ""
```

```
mostFilling = -10000
```

```
leastFilling = 10000
```

```
FOR Count = 1 TO 6
```

```
    IF fillings[Count] > mostFilling THEN
```

```
        mostFilling = fillings[Count]
```

```
        mostPopularName = fillingsName[Count]
```

```
    ELSEIF fillings[Count] < leastFilling THEN
```

```
        leastFilling = fillings[Count]
```

```
        leastPopularName = fillingsName[Count]
```

```
    END IF
```

```
NEXT
```

```
mostPopularPerc = (mostFilling / orderNo) * 100
```

```
leastPopularPerc = (leastFilling / orderNo) * 100
```

```
OUTPUT ""
```

```
OUTPUT "The most popular baguette filling was " & mostPopularName & " with an order of " &  
mostPopularPerc & "%"
```

```
OUTPUT "The least popular baguette filling was " & leastPopularName & " with an order of " &  
leastPopularPerc & "%"
```

Programming Language: VB

```
Dim totalSize15 As Integer
```

```
Dim totalSize30 As Integer
```

```
Dim totalBreadWhite As Integer
```

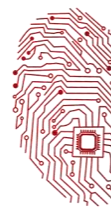
```
Dim totalBreadBrown As Integer
```

```
Dim totalBreadSeeded As Integer
```

```
Dim totalFillingBeef As Integer
```

```
Dim totalFillingChicken As Integer
```





PRM – Oct/Nov 2020 – P23

```
Dim totalFillingCheese As Integer

Dim totalFillingEgg As Integer

Dim totalFillingTuna As Integer

Dim totalFillingTurkey As Integer

Dim totalBaguettesSold As Integer

totalBaguettesSold = orderNo

totalSize15 = baguetteSize(1)

totalSize30 = baguetteSize(2)

totalBreadWhite = breadType(1)

totalBreadBrown = breadType(2)

totalBreadSeeded = breadType(3)

totalFillingBeef = fillings(1)

totalFillingChicken = fillings(2)

totalFillingCheese = fillings(3)

totalFillingEgg = fillings(4)

totalFillingTuna = fillings(5)

totalFillingTurkey = fillings(6)

Dim mostFilling As Integer

Dim leastFilling As Integer

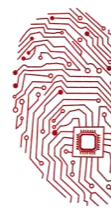
Dim mostPopularPerc As Single

Dim leastPopularPerc As Single

Dim mostPopularName As String = ""

Dim leastPopularName As String = ""
```





PRM – Oct/Nov 2020 – P23

```
mostFilling = -10000
```

```
leastFilling = 10000
```

```
For count = 1 To 6
```

```
    If fillings(count) > mostFilling Then
```

```
        mostFilling = fillings(count)
```

```
        mostPopularName = fillingsName(count)
```

```
    ElseIf fillings(count) < leastFilling Then
```

```
        leastFilling = fillings(count)
```

```
        leastPopularName = fillingsName(count)
```

```
    End If
```

```
Next
```

```
mostPopularPerc = (mostFilling / orderNo) * 100
```

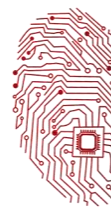
```
leastPopularPerc = (leastFilling / orderNo) * 100
```

```
Console.WriteLine("")
```

```
Console.WriteLine("The most popular baguette filling was " & mostPopularName & " with an  
order of " & mostPopularPerc & "%")
```

```
Console.WriteLine("The least popular baguette filling was " & leastPopularName & " with an  
order of " & leastPopularPerc & "%")
```





PRM – Oct/Nov 2020 – P23

Programming Language: Python

```
totalBaguettesSold = orderNo

totalSize15 = baguetteSize[0]
totalSize30 = baguetteSize[1]

totalBreadWhite = breadType[0]
totalBreadBrown = breadType[1]
totalBreadSeeded = breadType[2]

totalFillingBeef = fillings[0]
totalFillingChicken = fillings[1]
totalFillingCheese = fillings[2]
totalFillingEgg = fillings[3]
totalFillingTuna = fillings[4]
totalFillingTurkey = fillings[5]

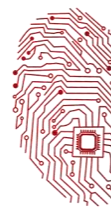
mostFilling = -10000
leastFilling = 10000

for i in range(6):
    if fillings[i] > mostFilling:
        mostFilling = fillings[i]
        mostPopularName = fillingsName[i]
    elif fillings[i] < leastFilling:
        leastFilling = fillings[i]
        leastPopularName = fillingsName[i]

mostPopularPerc = (mostFilling / totalBaguettesSold) * 100
leastPopularPerc = (leastFilling / totalBaguettesSold) * 100

print("")
print("The most popular baguette filling was {} with an order of
```





PRM – Oct/Nov 2020 – P23

```
{}}".format(mostPopularName, mostPopularPerc))  
    print("The least popular baguette filling was {} with an order of  
{}}".format(leastPopularName, leastPopularPerc))
```

Additional answer for (f)

"The most popular baguette filling was tuna with an order of 40%"

"The least popular baguette filling was cheese/egg with an order of 0%"

e: [5]

f: [5]

