COMPUTATIONAL THINKING

Mock Qualifier Solution

Total Marks: 25 Duration: 45 mins

1. The following pseudocode is executed using the "Scores" table. What will the value of **A** represent at the end of the execution? It is a Multiple Select Question (MSQ). [4 Marks]

```
A = 0
while (Table 1 has more rows) {
    Read the first row X in Table 1
    if (X.Gender == "M" or X.Total > 80) {
        A = A + 1
    }
    Move X to Table 2
}
```

a. Number of male students with total marks more than 80

b. Number of male students + Number of female students with total marks more than 80

c. Number of male students

d. Number of male students with total marks more than 80 + Number of female students with total marks at most 80

e. Number of students with total marks more than 80 + Number of male students with total marks at most 80

<u>Solution</u>: The pseudocode has an iteration. We use the variable **X** to iterate the rows of Table 1. In the iteration process, we have a filter $\mathbf{X}.Gender ==$ "M" or $\mathbf{X}.Total > 80$. The filter has two conditional statements that are merged by an `or` operator. The first condition $\mathbf{X}.Gender ==$ "M" (refer it as C1) filters only the male students, and the second condition $\mathbf{X}.Total > 80$ (refer it as C2) filters only the students with total marks more than 80. Since the conditions are merged with `or` operator, for any row, either one of the condition is `True` then the variable **A** will be incremented.

Evaluation of conditional statement with `or` operator: Let C1 `or` C2 be the conditional statement. The condition is evaluated as follows:

- If C1 is evaluated `True` then execute the `if-block`.
- If C1 is evaluated `False` then evaluate C2. If C2 is evaluated `True` then execute the `if-block`. Otherwise, do not execute the `if-block`.

Let **Y** be the set of students who are filtered by C1, and let **Z** be the set of students who are filtered by C2. Note that the sets **Y** and **Z** is need not to be disjoint set. There can be some students presented in both sets. So we have to exclude the students who are presented in the set **Y** from **Z**. Therefore, the set **Z** can be redefined such that

Z- Set of all students who are filtered by C2 but not filtered by C1

By the above definition,

- The size of the set **Y** is the number of male students, and
- the size of the set **Z** is the number of female students with total marks more than 80.

Therefore, at the end of the execution, the variable **A** captures the following:

Number of male students + Number of female students with total marks more than 80

Similarly, C1 or C2 is equivalent to C2 or C1. By applying first filter as C2, let **Y** be the set of students who are filtered by C2, and let **Z** be the set of students who are filtered by C2 but not filtered by C1. By the above definition,

- The size of the set **Y** is the number of students with total marks more than 80, and
- the size of the set **Z** is the number of male students with total marks at most 80.

Therefore, at the end of the execution, the variable **A** captures the following:

Number of students with total marks more than 80 + Number of male students with total marks at most 80

2. Consider the procedure **doCounting** for the following questions. [12 Marks]

```
Procedure doCounting ()
    A = 0, B = 0, C = 0
    P = False, Q = False
    while (Table 1 has more rows) {
        Read the first row X in Table 1
        if (X.PartOfSpeech == "Pronoun" and Q) {
            B = B + X.LetterCount
        }
        if (X.PartOfSpeech == "Noun") {
            if (P) {
                C = C + X.LetterCount
            }
            P = True
        }
        else {
            P = False
        }
        Q = True
        if (X.Word ends with a full stop) {
            A = A + X.LetterCount
            Q = False
        }
        Move X to Table 2
    }
    return ([A, B, C])
End doCounting
```

(i) What will the value of **A** represent at the end of the procedure?

a. Sum of letter count of words which are the end of each sentence and the part of speech is "Pronoun"

b. Sum of letter count of words which are the end of each sentence except for the first sentence.

c. Sum of letter count of words which are the end of each sentence

d. Sum of letter count of words which are the end of each sentence but not the end of the paragraph

Solution: The variable **A** is initialized with value 0. It is getting updated in one place during the course of execution. At Line 20, the variable **A** is added with the value of the letter count of the current word which is filtered by the condition

X.*Word* ends with a full stop.

That is, the variable **A** sums up the letter counts of the words which are the end of each sentence. Therefore, the variable **A** captures the following:

Sum of letter count of words which are the end of each sentence

(ii) What will the value of **B** represent at the end of the procedure?

a. Sum of letter count of all pronouns which appear at the end of a sentence
b. Sum of letter count of all pronouns except the pronouns which appear at the end of a sentence
c. Sum of letter count of all pronouns except the pronouns which appear at the beginning of a sentence
d. Sum of letter count of all pronouns which appear at the beginning of a sentence

Solution: The variable **B** is initialized with value 0. It is getting updated in one place during the course of execution. At Line 7, the variable **B** is added with the value of the letter count of the current word which is filtered by the condition

$$\mathbf{X}.\mathit{PartOfSpeech} == \text{“Pronoun” and } \mathbf{Q}.$$

The filter is based on two conditions merged with an `and` operator. The first condition is to check the part of speech is "Pronoun" and the second condition is to check the variable **Q** is evaluated to `True`. We have to find what is the role of the variable **Q** in this procedure. The variable **Q** is initially set to `False`. During the course of execution, **Q** is set to `True` in line 18 and set to `False` in Line 21. For each row in the table, after reading the row, **Q** is set to `True`. But, while reading the end of a sentence, **Q** is set to `False`. That is, when we read the first word of each sentence, **Q** is set to `False`. Therefore, the **Q** is set to `True`when the current word is not a beginning of a sentence.

By combining the above two conditions, the filter in Line 6 captures the pronouns which are not the beginning of a sentence. Therefore, the variable **B** captures the following:

Sum of letter count of all pronouns except the pronouns which appear at the beginning of a sentence.

(iii) What will the value of **C** represent at the end of the procedure?

a. Sum of letter count of all nouns except the first in each sentence
b. Sum of letter count of alternative nouns
c. Sum of letter count of alternative nouns in each consecutive noun sequence
d. Sum of letter count of all consecutive nouns except the first in each such sequence

Solution: The variable **C** is initialized with value 0. It is getting updated in one place during the course of execution. At Line 11, the variable **C** is added with the value of the letter count of the current word which is filtered by the condition

$$\mathbf{X}.\mathit{PartOfSpeech} == \text{“Noun” and } \mathbf{P}.$$

Note that the nested if-conditions are merged using the `and` operator. The first condition is to check the part of speech is “Noun” and the second condition is to check the variable **P** is evaluated to `True`. We have to find what is the role of the variable **P** in this procedure. The variable **P** is initially set to `False`. During the course of execution, **P** is set to `True` in line 13 and set to `False` in Line 16. For each row in the table, after reading a noun, **P** is set to `True`. But, while reading a word which is not a noun, **P** is set to `False`. That is, when we read a noun immediate to a next noun in the same sentence, **P** is set to `True`. Therefore, the **P** is set to `True` for all nouns in a sequence except for the first noun in the sequence.

By combining the above two conditions, the filters in Line 9 and 10 together capture the nouns which are not the first in a sequence of nouns. Therefore, the variable **C** captures the following:

Sum of letter count of all consecutive nouns except the first in each such sequence

(iv) If the `and` operator in Line 6 is replaced by the `or` operator, then what will the value of **B** represent at the end of the procedure?

a. Sum of letter count of all pronouns except the first in each sentence
b. Sum of letter count of all pronouns or the first word of each sentence
c. Sum of letter count of all words except the pronouns which are at the beginning of a sentence
d. Sum of letter count of all words except the words which are the first word of a sentence and the part of speech is not "Pronoun"

Solution: The variable **B** is initialized with value 0. It is getting updated in one place during the course of execution. At Line 7, the variable **B** is added with the value of the letter count of the current word which is filtered by the condition

$$\mathbf{X}.\mathit{PartOfSpeech} == \text{"Pronoun" or } \mathbf{Q}.$$

The filter is based on two conditions merged with an `or` operator. The first condition is to check the part of speech is "Pronoun" and the second condition is to check the variable **Q** is evaluated to `True`. We have to find what is the role of the variable **Q** in this procedure. The variable **Q** is initially set to `False`. During the course of execution, **Q** is set to `True` in line 18 and set to `False` in Line 21. For each row in the table, after reading the row, **Q** is set to `True`. But, while reading the end of a sentence, **Q** is set to `False`. That is, when we read the first word of each sentence, **Q** is set to `False`. Therefore, the **Q** is set to `True`when the current word is not a beginning of a sentence.

By combining the above two conditions, the filter in Line 6 captures all the pronouns and the all the words which are not the beginning of a sentence. But pronouns in the beginning of a sentence were filtered by the condition. Therefore, the variable **B** captures the following:

Sum of letter count of all words except the words which are the first word of a sentence and the part of speech is not "Pronoun".

3. The following pseudocode is executed using the "Scores" table. At the end of the execution, **Z** captures the number of pairs of students with the sum of difference of each subject marks is the same as the difference of the total marks. Choose the correct choice to complete the pseudocode. [4 marks]

```
1   Z = 1
2   while (Table 1 has more rows) {
3       Read the first row X in Table 1
4       Move X to Table 2
5       while (Table 1 has more rows) {
6           Read the first row Y in Table 1
7           value = Proc2(X.Physics, Y.Physics)
8           value = value + Proc2(X.Chemistry, Y.Chemistry)
9           value = value + Proc2(X.Mathematics, Y.Mathematics)
10          value = value + Proc2(X.Total, Y.Total)
11          if (value == 0) {
12              Z = Z + 1
13          }
14          Move Y to Table 3
15      }
16      Move all rows from Table 3 to Table 1
17  }
18
19  Procedure Proc2(C, D)
20      if (C < D) {
21          return (C - D)
22      }
23      else {
24          return (D - C)
25      }
26  End Proc2
```

a. Error in Line 1
b. Error in Line 7
c. Error in Line 8
d. Error in Line 9
e. Error in Line 10
f. Error in Line 14
g. Error in Line 20
h. No error

Solution: **X** and **Y** are two students in the dataset. The difference of the marks in each subject is as follows:

- | **X**.*Mathematics* - **Y**.*Mathematics* |
- | **X**.*Physics* - **Y**.*Physics* |
- | **X**.*Chemistry* - **Y**.*Chemistry* |

The sum of the differences is given by the following expression:

| **X**.*Mathematics* - **Y**.*Mathematics* | + | **X**.*Physics* - **Y**.*Physics* | + | **X**.*Chemistry* - **Y**.*Chemistry* |

According to the condition given in the question, we wish to count all pairs of students (**X**, **Y**) such that the sum of the differences given above is equal to the difference in the total marks. In other words, we want to count those pairs for which the following condition is satisfied:

| **X**.*Mathematics* - **Y**.*Mathematics* | + | **X**.*Physics* - **Y**.*Physics* | + | **X**.*Chemistry* - **Y**.*Chemistry* | =| **X**.*Total* - **Y**.*Total* |

A little bit of rearranging gives:

| **X**.*Mathematics* - **Y**.*Mathematics* | + | **X**.*Physics* - **Y**.*Physics* | + | **X**.*Chemistry* - **Y**.*Chemistry* | - | **X**.*Total* - **Y**.*Total* | = 0

If we wish to write a program to achieve this, we can use a variable called **value** that accumulates the differences of the individual subjects.

This is exactly what is happening in lines 7, 8 and 9 of the pseudocode. So, there is nothing wrong in any of these lines. After adding the subject wise differences, we can subtract the difference in the total marks. According to the condition given above, this should result in 0. In line 10, instead of subtracting the difference in the total marks, we are adding it. This is an error. So there is an error in line 10. The correct statement should be:

value = **value** - **Proc2**(**X**.*Total* , **Y**.*Total*)

Now, in line 11 we are checking if **value** is 0. This is correct. Line 12 is incrementing **Z**. This is also correct. But this brings us back to the very first line. In line 1, **Z**is initialized to zero. This is an error. So far, lines 1 and 10 have errors in them.

That leaves us with the procedure **Proc2**. Notice that the procedure is returning **C** - **D** if **C** < **D**. That is, it is returning the negative difference between **C** and **D**. At the outset, this seems like an error. But if we look carefully, we are consistently following this throughout. While calling the procedure, **C** always takes the marks of **X** and **D** always takes the marks of **Y**.

To summarize, the pseudocode has two errors, one in line 1 and the other in line 10.

4. Let **X** and **Y** be two rows in the "Library" table. We call **X** and **Y** compatible if either published in the same year or the same genre but not both. Let **CompatiblePair**(**X**, **Y**) be a procedure to find whether **X** and **Y** are compatible. Choose the correct implementation of the procedure **CompatiblePair**. [5 marks]

a.

```
Procedure CompatiblePair (X, Y)
    A = False, B = False, C = False
    if (X.Year == Y.Year) {
        A = True
    }
    if (X.Genre == Y.Genre) {
        B = True
    }
    if (A or B) {
        return (True)
    }
    return (False)
End CompatiblePair
```

b.

```
Procedure CompatiblePair (X, Y)
    A = False, B = False, C = False
    if (X.Year == Y.Year or X.Genre == Y.Genre ) {
        return (True)
    }
    return (False)
End CompatiblePair
```

c.

```
Procedure CompatiblePair (X, Y)
    A = False, B = False, C = False
    if (X.Year == Y.Year) {
        A = True
    }
    if (X.Genre == Y.Genre) {
        B = True
    }
    if ((A and B) == False or (A or B) == True) {
        return (True)
    }
    return False
End CompatiblePair
```

d.

```
Procedure CompatiblePair (X, Y)
    A = False, B = False, C = False
    if (X.Year == Y.Year) {
        A = True
    }
    if (X.Genre == Y.Genre) {
        B = True
    }
    if ((A and B) == False and (A or B) == True) {
        return (True)
    }
    return False
End CompatiblePair
```

Solution: We have to use two boolean variables to check that the published year and the genre of the boos are the same. In the procedure, we are using **A** to check the publishing year and **B** to check the genre. This can be done by the following piece of codes.

```
if (X.Year == Y.Year) {
    A = True
}
```

and

```
if (X.Genre == Y.Genre) {
    B = True
}
```

Finally, we have to check the condition that either **A** or **B** must be `True` but not both. This is possible only when one of the variable is set to `True` and the other one is set to `False`. Therefore, the following filter captures the condition:

- One of the variable is set to `True`:

$$(\mathbf{A} \text{ or } \mathbf{B}) == \texttt{True}$$

- One of the variable is set to `False`:

$$(\mathbf{A} \text{ and } \mathbf{B}) == \texttt{False}$$

- by combining above two conditions, we capture the condition that either **A** or **B** is `True` but not both:

$$(\mathbf{A} \text{ or } \mathbf{B}) == \texttt{True} \text{ and } (\mathbf{A} \text{ and } \mathbf{B}) == \texttt{False}$$

For the above condition, the procedure **CompatiblePair** returns `True`. That is,

```
if ((A or B) == True and (A and B) == False) {
    return True
}
```

Otherwise, the procedure returns `False`. By combine all the above piece of codes, we get the following procedure implementation.

```
Procedure CompatiblePair (X, Y)
    A = False, B = False, C = False
    if (X.Year == Y.Year) {
        A = True
    }
    if (X.Genre == Y.Genre) {
        B = True
    }
    if ((A and B) == False and (A or B) == True) {
        return (True)
    }
    return False
End CompatiblePair
```