REDSPOT Fresh Minds New Thoughts

About O LEVEL

COMPUTER SCIENCE
Paper 2 2210
(TOPICAL)

About **Teacher's Comments**

When solving problems, we first analyse the questions and then gather relevant information until we are able to determine the answers. But for presentation reason, we need to rearrange and then present ONLY the required workings and solutions.

Teacher's Comments reveal the extra but relevant information which is not required as part of the solutions.

period	**2014 to 2024**
contents	**June & November Exams, P2 Worked Solutions**
form	**Topic By Topic**
compiled for	**O Levels**
special features	**Teacher's Comments**

Appointed Agents & Wholesalers in PAKISTAN:

BOOK LAND
Urdu Bazaar, Lahore. Tel: 042-37124656

NATIONAL BOOK SERVICE
Urdu Bazaar, Lahore. Tel: 042-37247310.

LAROSH BOOKS
Urdu Bazaar Lahore. Tel: 042-37312126.

BURHANI BOOK CENTRE
New Urdu Bazar, Karachi, Tel: 021-32634718

MARYAM ACADEMY
Urdu Bazaar, Karachi, Tel: 0331-2425264

TARIQ BOOK TOWN
Samar Garden, Hydari North nazimzbad, Karachi
Tel: 021-34555918, 36649425

REHMAN BOOKS
College Road, Rawalpindi
Tel: 051-5770603, 0313-5618976

WELCOME BOOKS
Soneri Masjid Road, Peshawar Cantt.
Tel: 091-5274932, 0300-5860068

For Online Orders, visit pak.redspot.pk

Tel No: 042-35201010
Mobile No: 0300-8447654
E-Mail: info@redspot.com.pk
Website: www.redspot.pk
Address: P.O. Box 5041, Model Town, Lahore, Pakistan.

CONTENTS

Revision: June / November **2024** Paper 2

Topic 1 Algorithms, Programming and Logic

TOPIC 1.1
Algorithms

Question 1

A school has 1500 students. It is conducting a survey on their music preferences. Each student uses a computer and inputs their name and then chooses one of 5 options:

- rock (input value 1)
- soul (input value 2)
- pop (input value 3)
- jazz (input value 4)
- classical (input value 5)

Write an algorithm, using pseudocode **or** a flowchart, which:

- inputs the choice of all 1500 students (values 1 to 5)
- outputs all the names of the students who chose classical music
- outputs the percentage who chose **each** option.

.....

.....

.....

..... [5]

[J14/P1/Q18]

Question 2

A school has 3000 students sitting final examinations.
Each student sits eight examinations.

Write an algorithm, using pseudocode **or** a flowchart, which:

- inputs the marks for **all** 8 examinations for **each** student
- outputs for each student the average mark for their 8 examinations
- outputs the highest mark overall

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

[N14/P1/Q16]

Question 3

Pre-release Material

Write and test a program to complete the **three** tasks.

TASK 1

A data logger records the temperature on the roof of a school twice a day, at midday and midnight. Input and store the temperatures recorded for a month. You must store the temperatures in two onedimensional arrays, one for the midday temperatures and one for the midnight temperatures. All the temperatures must be validated on entry and any invalid temperatures rejected. You must decide your own validation rules. You may assume that there are 30 days in a month.

TASK 2

Calculate the average temperature for midday and the average temperature for midnight. Output these averages with a suitable message for each one.

TASK 3

Select the day with the highest midday temperature and the day with the lowest midnight temperature. Then output each of these temperatures, the corresponding day and a suitable message.

Your program must include appropriate prompts for the entry of data. Error messages and other outputs need to be set out clearly and understandably. All variables, constants and other identifiers must have meaningful names. Each task must be fully tested.

(a) All variables, constants and other identifiers should have meaningful names.

(i) In **Task 1**, you had to store the midday temperatures and midnight temperatures in arrays. Write suitable declarations for these **two** arrays.

.....

..... [2]

(ii) It has been decided to record the temperatures for one week rather than one month. Write the new array declarations that you would use.

.....

..... [1]

(iii) Declare **two** other variables that you have used and state what you used each one for.

Variable 1 .....

Use .....

Variable 2 .....

Use .....

[4]

(b) Write an algorithm to complete **Task 2**, using **either** pseudocode, programming statements **or** a flowchart. Use temperatures for **one week** only. You should assume that Task 1 has already been completed.

..... [5]

(c) Give a set of midday temperature data, for a week, that could be used to check your validation rules for **Task 1**. Explain why you chose this data set.

Data set .....

.....

Reason for choice .....

.....

.....

[2]

(d) Explain how you select the day with the highest midday temperature (part of **Task 3**). You may include pseudocode or programming statements as part of your explanation.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

[J15/P21/Q1]

Question 4

Pre-release Material

Write and test a program to complete the **three** tasks.

TASK 1

A school keeps records of the weights of each pupil. The weight, in kilograms, of each pupil is recorded on the first day of term. Input and store the weights and names recorded for a class of 30 pupils. You must store the weights in a one-dimensional array and the names in another onedimensional array. All the weights must be validated on entry and any invalid weights rejected. You must decide your own validation rules. You may assume that the pupils' names are unique. Output the names and weights of the pupils in the class.

TASK 2

The weight, in kilograms, of each pupil is recorded again on the last day of term. Calculate and store the difference in weight for each pupil.

TASK 3

For those pupils who have a difference in weight of more than 2.5 kilograms, output, with a suitable message, the pupil's name, the difference in weight and whether this is a rise or a fall.

Your program must include appropriate prompts for the entry of data. Error messages and other outputs need to be set out clearly and understandably. All variables, constants and other identifiers must have meaningful names. Each task must be fully tested.

(a) All variables, constants and other identifiers should have meaningful names.

(i) Declare the array to store the pupils' names.

..... [1]

(ii) Declare the array to store the pupils' weights.

..... [1]

(iii) It has been decided to record the weights for the whole school of 600 pupils rather than one class.

Write suitable new declarations for these two arrays.

.....

..... [1]

(b) Write an algorithm to complete **Task 2**, using **either** pseudocode, programming statements **or** a flowchart. Use weights for **the whole school**. You should assume that Task 1 has already been completed.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(c) **(i)** Describe suitable validation rules for **Task 1**.

.....

.....

..... [2]

(ii) Give **two** pupil weights that you could use to check the validation used in **Task 1**. Explain why you chose each weight.

Weight 1 .....

Reason for choice .....

.....

ANSWERS

TOPIC - 1.1

Q1 - Solution

```
rock = 0: soul = 0: pop = 0: jazz = 0: classical = 0
for StudentNumber = 1 to 1500
    input StudentName, MusicChoice
        if MusicChoice = 1 then rock = rock + 1
        if MusicChoice = 2 then soul = soul + 1
        if MusicChoice = 3 then pop = pop + 1
        if MusicChoice = 4 then jazz = jazz + 1
        if MusicChoice = 5 then classical = classical + 1
        if MusicChoice = 5 then output StudentName
next StudentNumber
PercentRock = rock /15
PercentSoul = soul /15
PercentPop = pop /15
PercentJazz = jazz /15
PercentClassical = classical /15
output PercentRock, PercentSoul, PerecentPop, PercentJazz, PercentClassical
```

Teacher's Comments:

Instead of the "if…then.." statements, "case of… endcase" statement could also have been used to replace lines 4-9 as shown below:

```
case of MusicChoice:
1: rock = rock+1
2: soul = soul+1
3: pop = pop+1
4: jazz = jazz+1
5: classical = classical + 1
output   StudentName
endcase
```

Q2 - Solution

```
HighestMark = –1
for StudentNumber = 1 to 3000
    total = 0
    for ExamNumber = 1 to 8
        input mark
        total = total + mark
        if mark > HighestMark then HighestMark = mark
    next
    average = total / 8
    output average
next
output HighestMark
```

Q3 - Solution

(a) **(i)** TempMidday[1:30]
TempMidnight[1:30]

(ii) TempMidday[1:7]
TempMidnight[1:7]

(iii) Variable 1: LowMidNight (Real),
Reason: To Store the lowest midnight temperature.

Variable 2: MaxCounter (Integer),
Reason: To use as a for-loop counter when finding maximum Midday temperatures.

(b)
```
SumMidday=0
SumMidnight=0
FOR AverageCounter = 1 TO 7
    SumMidday=SumMidday + TempMidday[AverageCounter]
    SumMidnight=SumMidnight + TempMidnight [AverageCounter]
NEXT AverageCounter
MiddayAverage = SumMidday/7
MidnightAverage = SumMidnight/7
PRINT 'The Average Midday Temperature is', MiddayAverage
PRINT 'The Average Midnight Temperature is', MidnightAverage
```

(c) Data Set: 24, 28, 25.5, 27.9, 21.4, 23, 24.8
Reason: Normal data that should be accepted by the validation rules of the program.

(d) Set a variable "HighestMidday" with a very low negative value. This variable will be used to hold the highest midday temperature value. Set a variable "DayofWeek". This variable will be used to hold the day the highest midday temperature occurred. Loop through all the 7 midday temperatures of the week, which are stored in the array "TempMidday". Compare each midday temperature against the value stored in "HighestMidday". If the midday temperature is greater than the value stored in "HighestMidday", replace the value of "HighestMidday" with the midday temperature value and store the array index of the array "TempMidday" in the variable "DayofWeek". Once the loop has finished, output the results of "HighestMidday" and "DayofWeek" with suitable PRINT statements.

Pseudo-code:

```
HighestMidday= -999
DayofWeek=0
FOR MaxCounter = 1 TO 7
    IF TempMidday[MaxCounter]>HighestMidday
        THEN
          HighestMidday = TempMidday[MaxCounter]
          DayofWeek=MaxCounter
        ENDIF
NEXT MaxCounter
PRINT 'The Highest Midday Temperature was: ', HighestMidday, 'and was recorded on day number', DayofWeek
```

Q4 - Solution

(a) **(i)** PupilNames[1:30]

(ii) InitialWeights[1:30]

(iii) PupilNames[1:600]

InitialWeights[1:600]

(b)

```
FOR Count = 1 TO 600
NewWeight = 0
    REPEAT
        PRINT 'Please enter the weight for the student named: ', PupilNames[Count]
        INPUT NewWeight
    UNTIL NewWeight < 120 AND NewWeight > 20
WeightDifference[Count] = NewWeight – InitialWeights[Count]
NEXT Count
```

(c) **(i)** Check that the weights being input are numeric.

Check that the weights are within a given range.

(ii) Weight 1: 56.7

Reason for choice: This data represents normal data that should be accepted by the validation checks.

Weight 2: Forty

Reason for choice: This data represents abnormal and incorrect data that should be rejected by the validation checks.

(d)

```
FOR Count = 1 TO 600
IF WeightDifference[Count] < –2.5
    AbsWeightDifference = –1 *WeightDifference
   THEN PRINT 'Weight Loss by ', PupilNames[Count], ' was ', AbsWeightDifference
    END IF
NEXT Count
```

The algorithm loops through all the 600 students and checks the difference in weight. If the difference in weight is lesser than –2.5, then the program outputs the Pupil's name, the absolute weight difference and an output message stating that there is a fall in weight.

Q5 - Solution

(a)
```
FOR Count = 1 TO 1000
    INPUT Array[Count]
NEXT Count
```

(b)
```
Count = 0
WHILE Count < 1000
    DO
    Count = Count + 1
    INPUT Array[Count]
ENDWHILE
```

Q6 - Solution

(a) **(i)** Variable 1: LoopCounter (: INTEGER)
Use: To keep track as a loop counter when entering temperatures.

Variable 2: HighestTemp (: REAL)
Use: To store the highest recorded temperature.

(ii) Constant 1: MinAppartmentTemp = 21.5
Use: To store the lowest temperature below which the Air-conditioning should be switched off.

Constant 2: MaxAppartmentTemp = 24.5
Use: To store the highest temperature above which the Air-conditioning should be switched on.

(b) Assume that the Array in which the temperatures are stored for the five hour period is denoted by the variable name 'TempArray'. The pseudocode is then as follows:

```
HighestTemp = 0
LowestTemp = 100
FOR LoopCounter = 1 TO 60
    IF TempArray[LoopCounter] > HighestTemp
        THEN HighestTemp = TempArray[LoopCounter]
    ENDIF
    IF TempArray[LoopCounter] < LowestTemp
        THEN LowestTemp = TempArray[LoopCounter]
    ENDIF
NEXT LoopCounter
TempRange = HighestTemp – LowestTemp
PRINT 'The Highest recorded temperature in the Five hour period was: ', HighestTemp
PRINT 'The Lowest recorded temperature in the Five hour period was: ', LowestTemp
PRINT 'Difference between Highest and Lowest Recorded Temperatures is: ', TempRange
```

(c) **(i)** Check to see if the Highest and Lowest recorded temperatures are above or below the acceptable range of temperatures respectively. If this is not the case, then all the temperatures are within the acceptable normal range. Otherwise, If the Highest recorded temperature is above the acceptable range of 24.5 degrees celsius, then loop through all the recorded temperatures and check how many times the temperature goes above 24.5 degrees celsius.
Output a message stating the number of times the temperature goes above the maximum limit.
Similarly, If the Lowest recorded temperature is below the acceptable range of 21.5 degrees celsius, then loop through all the recorded temperatures and check how many times the temperature goes below 21.5 degrees celsius. Output a message stating the number of times the temperature goes below the minimum limit.

The pseudocode is as follows:

```
IF HighestTemp <= MaxAppartmentTemp AND LowestTemp >= MinAppartmentTemp THEN
    PRINT 'All Temperatures are within the nominal acceptable range'
    EXIT
ENDIF

IF HighestTemp > MaxAppartmentTemp THEN
CounterMax = 0
    FOR LoopCounter = 1 TO 60
      IF TempArray[LoopCounter] > MaxAppartmentTemp THEN
        CounterMax = CounterMax + 1
      ENDIF
    NEXT LoopCounter
PRINT 'Number of times the Temperature has exceeded 24.5 degrees celsius is: ', CounterMax
ENDIF

IF LowestTemp < MinAppartmentTemp THEN
CounterMin = 0
    FOR LoopCounter = 1 TO 60
      IF TempArray[LoopCounter] < MinAppartmentTemp THEN
        CounterMin = CounterMin + 1
      ENDIF
    NEXT LoopCounter
PRINT 'Number of times the Temperature has dropped below 21.5 degrees celsius is: ',
CounterMin
ENDIF
```

Teacher's Comments:

> Note that the constants MinAppartmentTemp and MaxAppartmentTemp correspond to the values 21.5 and 24.5 respectively

(c) (ii) The code is efficient in the sense that it only looks at all the values if any abnormal values exist in the first place. This can be seen from the first 'IF' Condition for Task 3. Therefore, it only checks for necessary conditions. However, it should be noted that Task 3 depends upon the results from Task 2 for proper execution.

Q7 - Solution

(a) (i) Variable 1: LoopCounter (: INTEGER)

Use: To keep track as a loop counter when entering temperatures of the baby.

Variable 2: TempBaby (: REAL)

Use: To store the temperature of the baby

(ii) Constant 1: MinTempBaby = 36.0

Use: To store the lowest acceptable temperature of the baby.

Constant 2: MaxTempBaby = 37.5

Use: To store the highest acceptable temperature of the baby.

TOPIC 3
Databases

Question 1

A database, PROPERTY, was set up to show the prices of properties for sale and the features of each property. Part of the database is shown below.

Property Type	Brochure No	Number of Bedrooms	Number of Bathrooms	Garden	Garage	Price in $
Bungalow	B17	7	4	Yes	Yes	750,000
Apartment	A09	2	1	No	No	100,000
House	H10	4	2	Yes	No	450,000
House	H13	3	2	Yes	No	399,000
Apartment	A01	2	2	No	Yes	95,000
Apartment	A16	1	1	No	No	150,000
House	H23	3	1	No	Yes	250,000
House	H46	2	1	Yes	Yes	175,000

(a) Give the number of fields that are in each record.

..... [1]

(b) State which field you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

..... [2]

(c) State the data type you would choose for each of the following fields.

Garage .....

Number of Bedrooms .....

Price in $ .....

[3]

(d) The query-by-example grid below selects all houses with more than 1 bathroom and more than 2 bedrooms.

Field:	Property Type	Number of Bedrooms	Number of Bathrooms	Price in $	Brochure No
Table:	PROPERTY	PROPERTY	PROPERTY	PROPERTY	PROPERTY
Sort:				Ascending	
Show:	☐	☐	☐	☑	☑
Criteria:	= 'House'	>2	>1		
or:					

Show what would be output.

.....

..... [2]

(e) Complete the query-by-example grid below to select and show the brochure number, property type and price of all properties with a garage below $200,000.

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

[4]

[J15/P21/Q7]

Question 2

A database, MARKS, was set up to record the test results for a class of students. Part of the database is shown below.

Student Name	Class ID	Maths	English	Science	History	Geography
Paul Smith	0017	70	55	65	62	59
Ravi Gupta	0009	29	34	38	41	44
Chin Hwee	0010	43	47	50	45	52
John Jones	0013	37	67	21	28	35
Diana Abur	0001	92	88	95	89	78
Rosanna King	0016	21	13	11	27	15

(a) Give the number of fields that are in each record.

..... [1]

(b) State which field you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

..... [2]

(c) The query-by-example grid below selects all students with more than 60 marks in History or more than 60 marks in Geography.

Field:	Student Name	History	Geography
Table:	MARKS	MARKS	MARKS
Sort:	Ascending		
Show:	☑	☐	☐
Criteria:		>60	
or:			>60

Show what would be output.

.....

..... [2]

(d) Complete the query-by-example grid below to select and show the student names only of all students with less than 40 marks in both Maths and English.

Field:			
Table:			
Sort:			
Show:	☐	☐	☐
Criteria:			
or:			

[3]

[J15 / P22 / Q6]

Question 3

A motor boat hire company decides to set up a database to keep information about boats that are available for hire. The database table, BOAT, will contain the following fields:

Boat Name; Model; Engine Power (in hp); Number of Seats; Life Raft (whether there is a life raft kept on the boat); Day Price (price for a day's hire).

(a) Give the data type you would choose for each field.

Boat Name .....

Model .....

Engine Power .....

Number of Seats .....

Life Raft .....

Day Price .....

[3]

(b) State a validation check that you can perform on each of these fields. Each validation check must be different.

Boat Name .....

Model .....

Number of Seats .....

Day Price .....

[4]

(c) Complete the query-by-example grid below to select and show the Boat Name, Model and Day Price of a day's hire for all boats with 4 seats and an Engine Power of more than 100 hp.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

[N15/P21/Q5]

Question 4

A picture gallery owner has decided to set up a database to keep information about the pictures he has for sale. The database table, PICTURE, will contain the following fields:

Title; Artist; Description; Catalogue Number; Size (area in square centimetres); Price; Arrived (date picture arrived at gallery); Sold (whether picture is already sold)

(a) **(i)** State what data type you would choose for each field.

Title .....

Artist .....

Description .....

Catalogue Number .....

Size .....

Price .....

Arrived .....

Sold .....

[4]

(ii) State which field you would choose for the primary key.

..... [1]

(b) Give a validation check that you can perform on each of these fields. Each validation check must be different.

Catalogue Number .....

Size .....

Price .....

Arrived .....

[4]

(c) Complete the query-by-example grid below to select and show the Catalogue Number, Title and Price of all unsold pictures by the artist 'Twister'.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

[N15 / P22 / Q6]

Question 5

A database, SOFASELECT, was set up to show the prices of suites, sofas and chairs for sale from an online furniture warehouse. Part of the database is shown below.

Description	Brochure Number	Number of Seats	Number of Pieces	Material	Colour	Price in $
Sofa	SF17	2	1	Leather	Red	950
Sofa	SF19	3	1	Vinyl	Black	1,000
Suite	SU10	4	3	Velvet	Green	1,500
Suite	SU23	5	3	Leather	Brown	950
Recliner chair	RC01	1	1	Leather	Cream	600
Chair	CH16	1	1	Vinyl	Red	250
Recliner sofa	RS23	4	1	Leather	Cream	1,200
Chair	CH10	1	1	Velvet	Red	175

(a) How many fields are in each record?

..... [1]

(b) State which field you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

..... [2]

(c) State the data type you would choose for each of the following fields.

Number of Seats .....

Price in $ .....

[2]

(d) The query-by-example grid below selects all the furniture in cream leather.

Field:	Description	Material	Colour	Price in $	Brochure Number
Table:	SOFASELECT	SOFASELECT	SOFASELECT	SOFASELECT	SOFASELECT
Sort:				Descending	
Show:	☑	☐	☐	☑	☑
Criteria:		= ‘Leather’	= ‘Cream’		
or:					

ANSWERS

TOPIC - 3

Q1 - Solution

(a) 7

(b) Brochure No.

Reason: The Brochure No. field is unique for each record.

(c) **Garage:** Boolean
Number of Bedrooms: Integer
Price in $: Integer

(d) 399000 H13
450000 H10

(e)

Field:	Property Type	Garage	Price in $	Brochure No.
Table:	PROPERTY	PROPERTY	PROPERTY	PROPERTY
Sort:				
Show:	☑	☐	☑	☑
Criteria:		=Yes	< 200000	
or:				

Q2 - Solution

(a) 7

(b) Field: Class ID

Reason: The Class ID field is unique for each student.

(c) Paul Smith, Diana Abur.

(d)

Field:	Student Name	Maths	English
Table:	MARKS	MARKS	MARKS
Sort:			
Show:	☑	☐	☐
Criteria:		<40	<40
or:			

Q3 - Solution

(a) Boat Name: Text
Model: Text
Engine Power: Number
Number of Seats: Number
Life Raft: Boolean
Day Price: Currency

(b) Boat Name: Presence Check
Model: Length Check
Number of Seats: Range Check
Day Price: Presence Check

(c)

Field:	Boat Name	Model	Day Price	Number of Seats	Engine Power
Table:	BOAT	BOAT	BOAT	BOAT	BOAT
Sort:					
Show:	☑	☑	☑	☐	☐
Criteria:				= 4	> 100
or:					

Q4 - Solution

(a) **(i)** Title: Text
Artist: Text
Description: Text
Catalogue Number: Text
Size: Number
Price: Currency
Arrived: Date
Sold: Boolean

(ii) Catalogue Number.

(b) Catalogue Number: Check Digit
Size: Range Check
Price: Presence Check
Arrived: Format Check

(c)

Field:	Catalogue No.	Title	Price	Artist	Sold
Table:	PICTURE	PICTURE	PICTURE	PICTURE	PICTURE
Sort:					
Show:	☑	☑	☑	☐	☐
Criteria:				='Twister'	False
or:					

Q5 - Solution

(a) 7

(b) Brochure Number

It uniquely identifies all records as it has no duplicate.

(c) **Number of Seats:** Integer

Price in $: Currency

(d) Recliner sofa 1200 RS23

Recliner chair 600 RC01

(e)

Field:	Brochure Number	Material	Colour	Price in $	Number of Seats
Table:	SOFASELECT	SOFASELECT	SOFASELECT	SOFASELECT	SOFASELECT
Sort:					
Show:	☑	☑	☑	☑	☐
Criteria:					>2
or:					

Q6 - Solution

(a) 6

(b) **Play:** Text

Number Seats Stalls: Number

Price Stalls Seats $: Currency

(c) As You Like It 01/07/2016

Julius Caesar 22/07/2016

Macbeth 14/07/2016

(d)

Field:	Play	Performance Date	Number Seats Circle	Price Circle Seats $
Table:	PLAYPRODUCTION	PLAYPRODUCTION	PLAYPRODUCTION	PLAYPRODUCTION
Sort:		Ascending/Descending		
Show:	☑	☑	☐	☑
Criteria:			>=6	
or:				

Q7 - Solution

(a) Field 1 name: EarTag
Data type: Text
Data sample: EAR9201

Field 2 name: DOB
Data type: Date
Data sample: 9/8/2017

Field 3 name: Gender
Data type: Text
Data sample: F

Field 4 name: Weight
Data type: Floating Point Number
Data sample: 12.3

(b) EarTag

(c)

Field:	EarTag	Gender	Weight	
Table:	SHEEP	SHEEP	SHEEP	
Sort:				
Show:	☑	☐	☐	☐
Criteria:		='M'	> 10	
or:				

Q8 - Solution

(a) **Field 1 Name:** Engine Number
Data type: Text
Data sample: 10432

Field 2 Name: Engine Class
Data type: Text
Data sample: F7

Field 3 Name: Service Date
Data type: Date
Data sample: 8/7/2017

(b) Engine Number

(c)

Field:	Engine Number	Engine Class	Service Date	
Table:	TRAIN	TRAIN	TRAIN	
Sort:				
Show:	☑	☐	☐	☐
Criteria:		Like 'P*'	<10/11/2016	
or:				