Computing National 5

Revision Support

SDD Questions

21-22

Development methodologies

Describe and implement the phases of an iterative development process: analysis, design, implementation, testing, documentation, and evaluation, within general programming problem-solving.

SQP Q2	Explain why it may be necessary to return to the implementation stage of an iterative development process after the testing stage. **(1)**

Analysis

Identify the purpose and functional requirements of a problem that relates to the design and implementation at this level, in terms of:

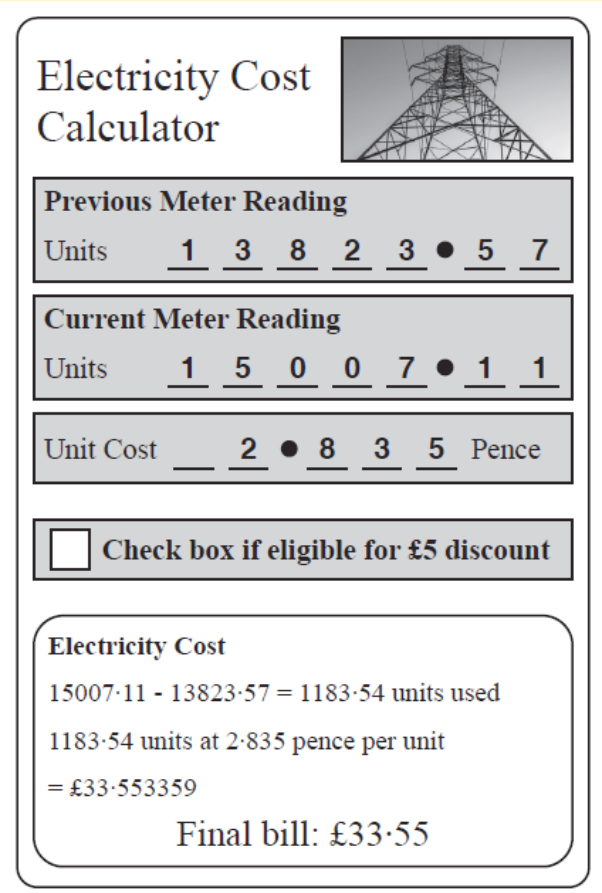
- inputs
- processes
- Outputs

<table>
<tr><td>SQP Q16(a)</td><td>Pam is creating an application that will find and display a person's tax rate based on their salary.

<table><tr><th>Salary</th><th>Tax rate</th></tr><tr><td>0–12000</td><td>0</td></tr><tr><td>12001–40000</td><td>20</td></tr><tr><td>40001 upwards</td><td>40</td></tr></table>
a) Analyse the problem and identify the input, the process and the output. (3)</td></tr>
<tr><td>2019 Q13(a)</td><td>A smart phone app is needed to calculate the cost of electricity. The following information will be entered by the user.

• Previous meter reading
• Current meter reading
• Unit cost
• Discount eligibility

A possible user interface for the app is shown below.</td></tr>
</table>



(a) Describe two processes that will be carried out by the program. **(2)**

2018 Q19(a)

A program is being designed that will allow pupils to add money to their lunch money account. The user enters their name, an 8 character password and the amount of money they want to add. A button is then clicked and the updated balance of the account is displayed.

(a) Analyse the problem and identify all inputs, processes and outputs. **(3)**

Design

Identify the data types and structures required for a problem that relates to the implementation at this level, as listed below:

Describe, identify, and be able to read and understand:

- structure diagrams
- flowcharts
- pseudocode

<table>
<tr>
<td>SQP
Q19 (a)

SQP
Q19 (b)</td>
<td>Read the following design for a solution to a problem.

Algorithm
1 Ask the user to enter their name
2 Ask the user to enter their flight details
3 Generate the holiday booking reference
4 Display the holiday booking reference

Refinements
1.1 Ask user to enter surname only

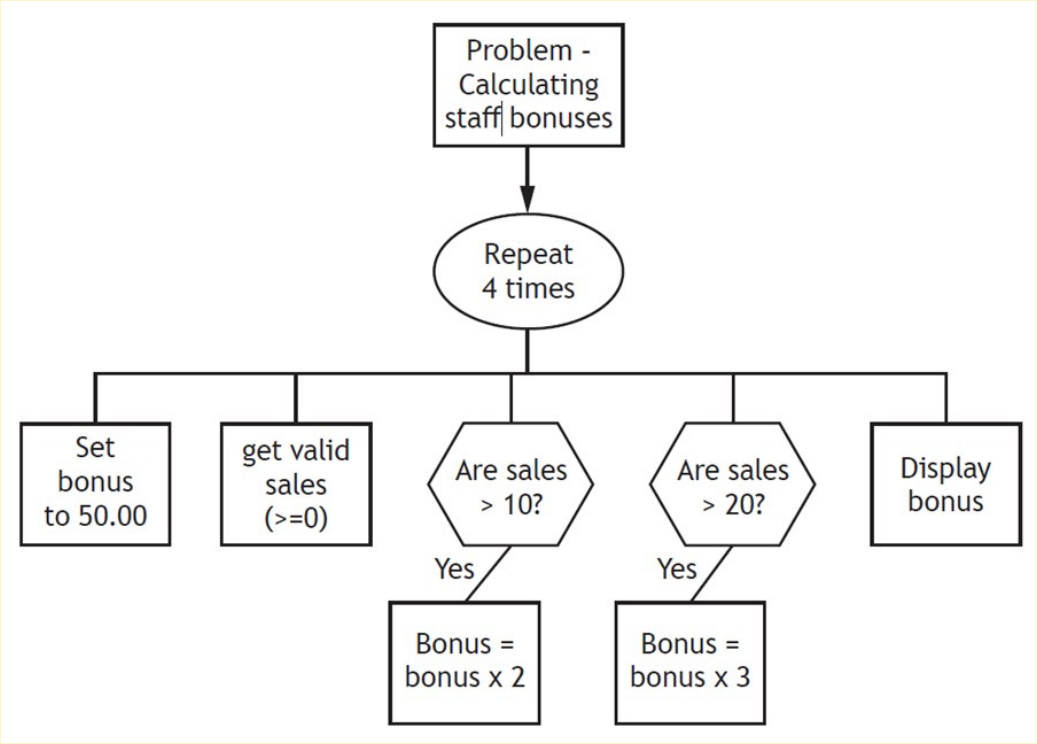
2.1 Ask user to enter first three letters of departure airport (for example: Edi for Edinburgh)
2.2 Ask user to enter first three letters of arrival airport

3.1 Store the booking reference as: arrival airport string + surname + departure airport string

(a) State which design technique has been used for the above solution. (1)

(b) State the output expected if the design is tested by Kate Bryant who is flying from Glasgow to Barcelona. (1)</td>
</tr>
<tr>
<td>SQP
Q21 (c)
(ii)</td>
<td>Arthur’s Antiques sells old furniture. All staff receive a monthly bonus of £50, which is increased if they sell over 10 items of furniture. The bonus is increased further if they sell over 20 items of furniture.

A design for the program used to calculate the bonus payment for each of the four members of staff is shown below.</td>
</tr>
</table>



(c) The program is further tested with normal test data. The results are shown below.

	Sales input	Expected output	Actual output
Staff 1	6	Bonus is 50	Bonus is 50
Staff 2	10	Bonus is 50	Bonus is 50
Staff 3	15	Bonus is 100	Bonus is 100
Staff 4	22	Bonus is 150	**Bonus is 300**

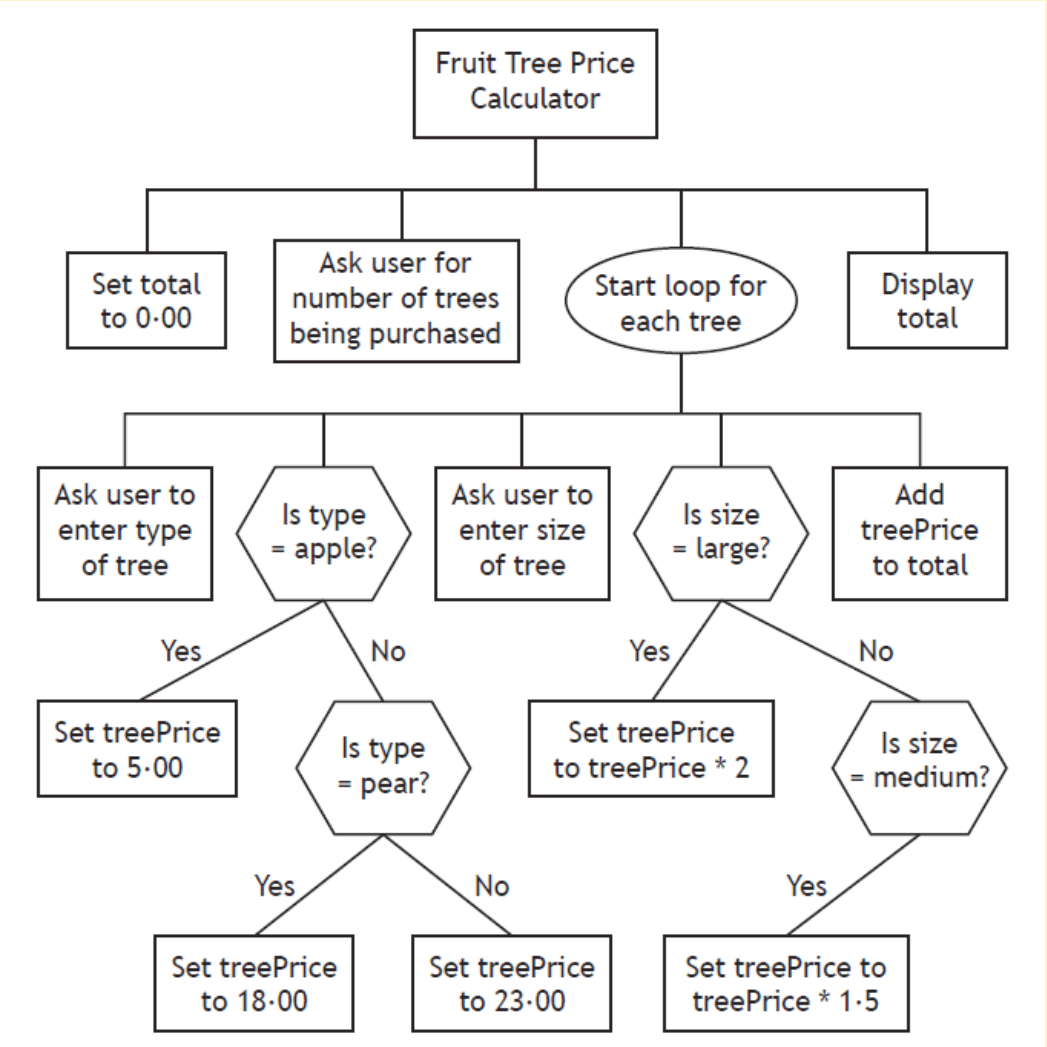
The test data for Staff 4 shows there is an error in the design.

(i) State the type of error. **(1)**

(ii) Describe how this design error could be corrected. You may wish to write a description or re-draw part of the design**. (2)**

2019 Q5

A garden centre requires a program to calculate the price of apple, pear and cherry trees being sold.
The design is shown below.



(a) State the type of loop shown in the design above. **(1)**

(b) The design is tested. For the following inputs state the total displayed. **(1)**

Inputs	Total displayed
Number of trees – 2 Type of tree – cherry Size of tree – small Type of tree – pear Size of tree – medium	

(c) The garden centre is considering selling orange trees for £23·00.

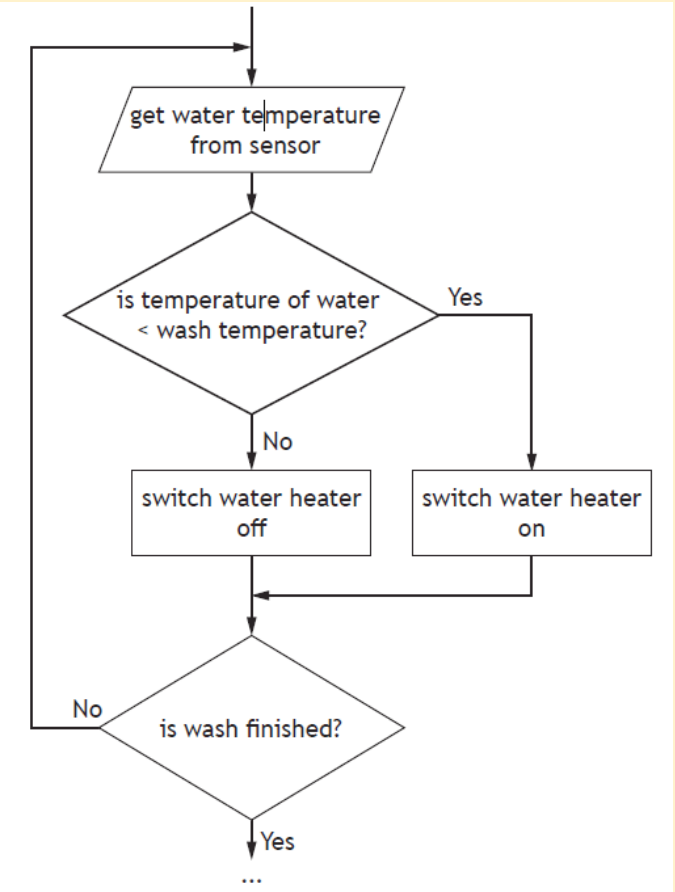
Explain why the design does not need to be changed. **(1)**

2019 Q16 (a)

2019 Q16 (b)

A program to control the water temperature inside a washing machine is being designed. The user will select a wash temperature using the control panel on the machine.

The program should ensure that the water stays heated at the correct temperature throughout the wash. The design for the part of the program that maintains the water temperature is shown below.



(a) State the design technique that has been used to design the solution. **(1)**

(b) To implement the program several programming constructs will be required.

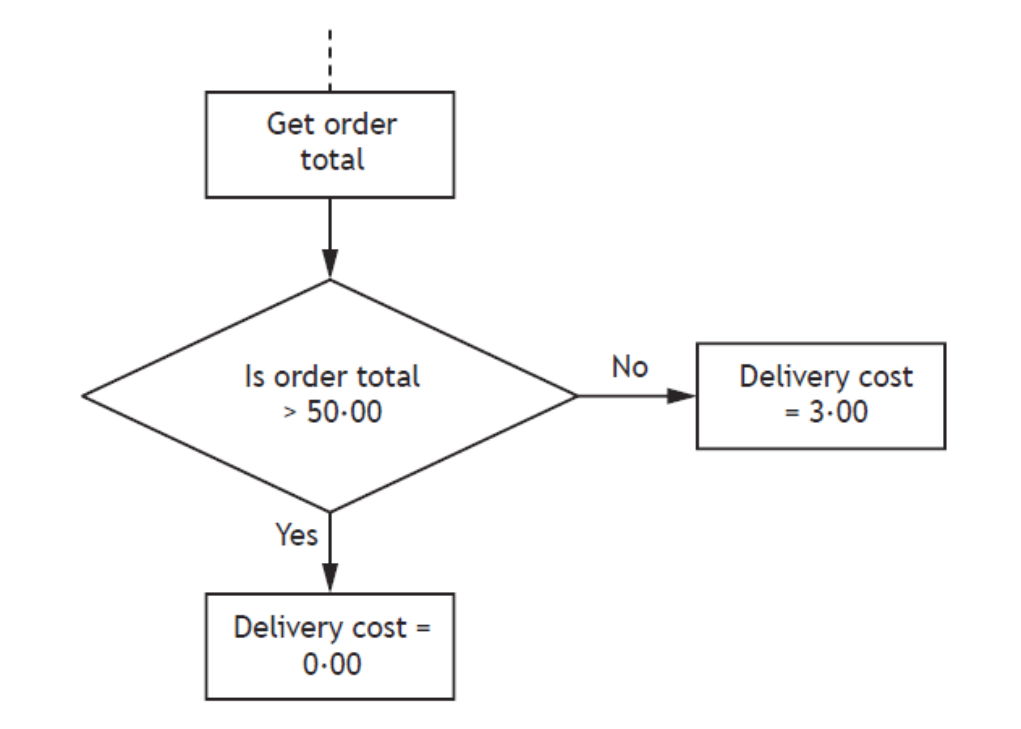
(i) State the condition used in the loop construct**. (1)**

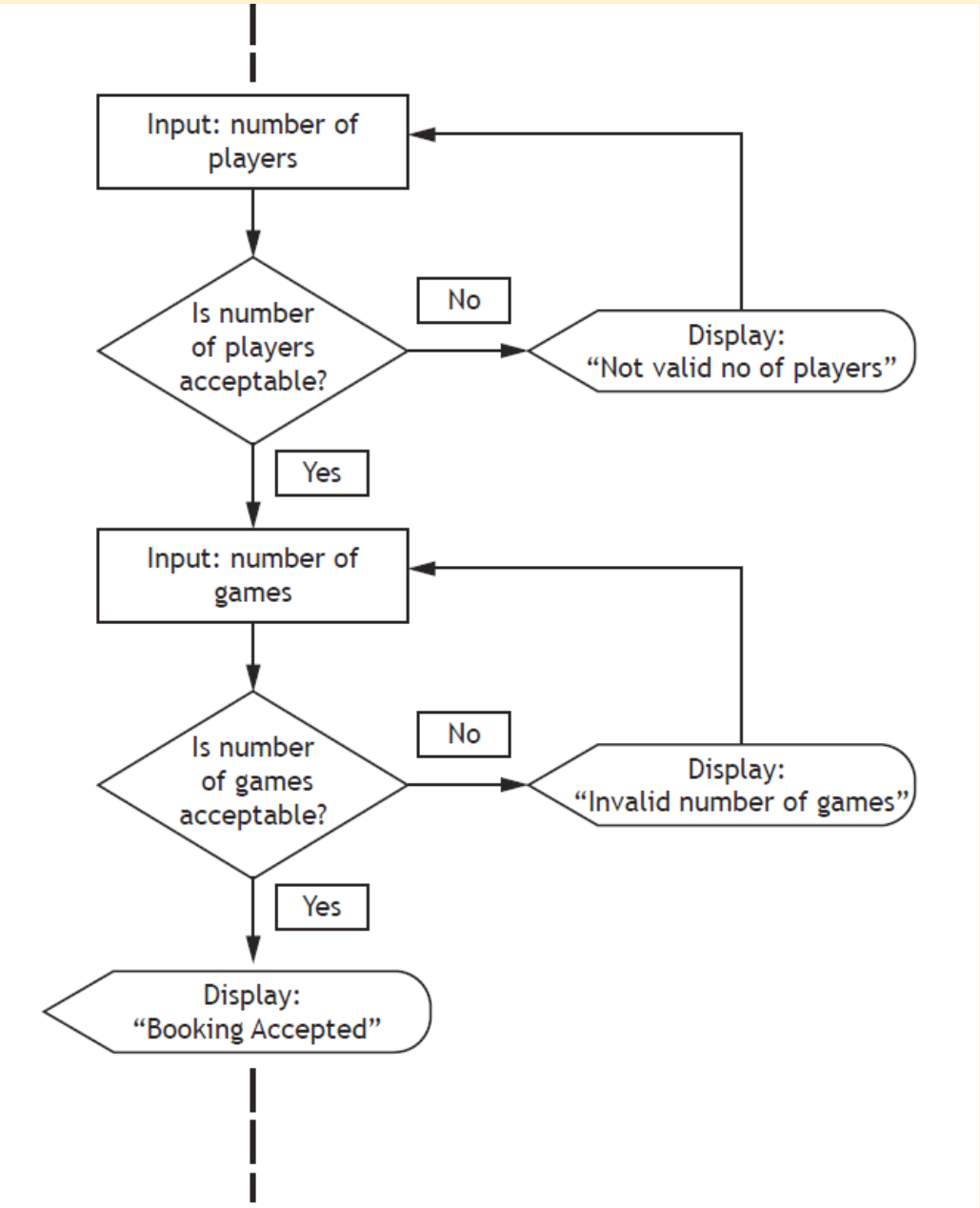
(ii) State one other construct that has been used in the design and describe how that construct has been used**. (2)**

Construct	
Description	

2017

Part of the design of a program is shown below.

Q13	 Identify the graphical design notation shown above. **(1)**
2016 Q21 (a)	A software developer is creating an online booking system for a bowling alley. Customers can book a bowling lane for a maximum of 4 people playing a maximum of 3 games. The developer has used a flow chart to produce the program design. Part of the design is shown below.



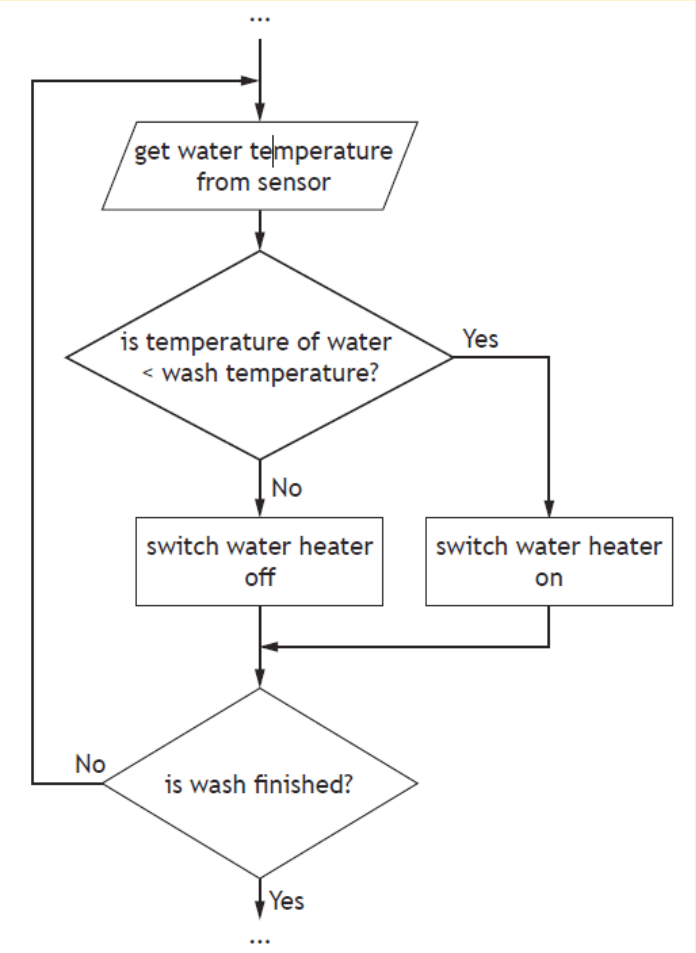
(a) (i) State one benefit of using the design notation shown above instead of pseudocode. **(1)**

(ii) Name the algorithm illustrated in the bowling alley program design**. (1)**

Exemplify and implement one of the above design techniques to design efficient solutions to a problem.

SQP Q16 (b)	Pam is creating an application that will find and display a person's tax rate based on their salary.

	<table><tr><th>Salary</th><th>Tax rate</th></tr><tr><td>0–12000</td><td>0</td></tr><tr><td>12001–40000</td><td>20</td></tr><tr><td>40001 upwards</td><td>40</td></tr></table> (b) Using a design technique of your choice, design an efficient solution to the problem of finding a person's tax rate. **(4)**
2019 Q16 (c)	A program to control the water temperature inside a washing machine is being designed. The user will select a wash temperature using the control panel on the machine. The program should ensure that the water stays heated at the correct temperature throughout the wash. The design for the part of the program that maintains the water temperature is shown below.



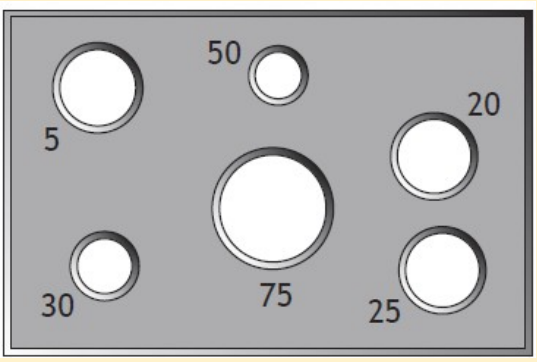
(c) When the wash is finished, the water will drain out. A sensor continuously detects the amount of water in the machine during the draining process.

When there is no more water in the machine the door will automatically open.

Using a design technique of your choice, design a solution to this problem. **(3)**

2019 Q19 (b) (ii)

A fairground game involves throwing balls through holes in a large wooden board. Each hole scores different points. The game is played using the following four rules.



1. A player starts with 3 balls and throws them one at a time.
2. If a ball is successfully thrown through a hole the points are added onto the player's score.
3. The game ends immediately if a player's score is greater than or equal to 50.
4. If the score reaches exactly 50 points the player is told they have won a prize.

A program is written to keep the score for a player.

```
...
Line 3    DECLARE total INITIALLY 0
Line 4    DECLARE balls INITIALLY 3
Line 5    WHILE total < 50 AND balls > 0 DO
Line 6       RECEIVE ballScoreOne FROM KEYBOARD
Line 7       SET total TO total + ballScoreOne
Line 8       SET balls TO balls - 1
Line 9       RECEIVE ballScoreTwo FROM KEYBOARD
Line 10      SET total TO total + ballScoreTwo
Line 11      SET balls TO balls - 1
Line 12      RECEIVE ballScoreThree FROM KEYBOARD
Line 13      SET total TO total + ballScoreThree
Line 14      SET balls TO balls - 1
Line 15   END WHILE
Line 16   SEND "Well done! You have won a prize." TO DISPLAY
```

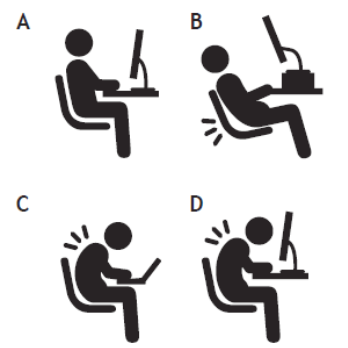
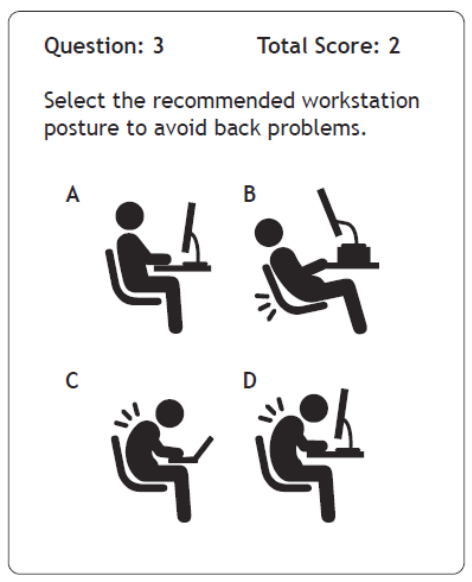
(b) The program runs but does not meet the functional requirements stated in the rules.

(ii) The program has been edited as shown, but still breaks rule 3 and rule 4 of the game.

```
...
Line 3    DECLARE total INITIALLY 0
Line 4    DECLARE balls INITIALLY 3
Line 5    WHILE total < 50 AND balls > 0 DO
Line 6       RECEIVE ballScore FROM KEYBOARD
Line 7       SET total TO total + ballScore
Line 8       SET balls TO balls - 1
Line 9    END WHILE
Line 10   SEND "Well done! You have won a prize." TO
          DISPLAY
```

Using a design technique of your choice, design a solution that meets the requirements of all four game rules. **(4)**

2018 Q19 (d)	A program is being designed that will allow pupils to add money to their lunch money account. The user enters their name, an 8 character password and the amount of money they want to add. A button is then clicked and the updated balance of the account is displayed. (d) Using a design technique of your choice, design an efficient solution to ensure that a password of only 8 characters can be entered. An error message should be displayed if the incorrect number of characters is entered, and the user asked to re-enter the password. **(4)**
2018 Q21 (a)	A program will calculate the total cost when customers purchase tickets to a theme park. Adults pay £25 per ticket; children pay £10. If there are two or more adults with more than two children a discount of £5 is subtracted from the total cost. Algorithm 1. Store cost of adult and child ticket 2. Get name of person making booking 3. Get quantity of tickets 4. Calculate total cost 5. Display food voucher message Refinement 2.1 Get first name 2.2 Get second name 3.1 Get quantity of adult tickets 3.2 Get quantity of child tickets (a) Using a design technique of your choice, refine step 4**. (6)**
2016 Q12	A running group has 16 members. They are taking part in a marathon. Using pseudocode or a programming language of your choice, write the code which will take in each runner's time for the marathon. **(2)**
2016 Q18 (d) (ii)	A software development company decides to review staff knowledge of computer related legislation. Mikal is asked to create an app covering a range of legal issues.

	(d) In line with Health and Safety legislation, the company provides adjustable seating and guidelines on maintaining good posture. Mikal finds graphics on a website that he can use to illustrate his next quiz question.  (ii) Mikal uses the graphics to create question 3 for the app.  Using pseudocode or a programming language of your choice, write the code to show how the total score is calculated when the user answers question 3 correctly**. (2)**

Describe, exemplify, and implement user-interface design, in terms of input and output, using a wireframe.

2019 Q3	A bank requires a program for loan applications. The user will enter how much money they want to borrow and the number of monthly repayments. The user will then be informed how much they must repay each month.

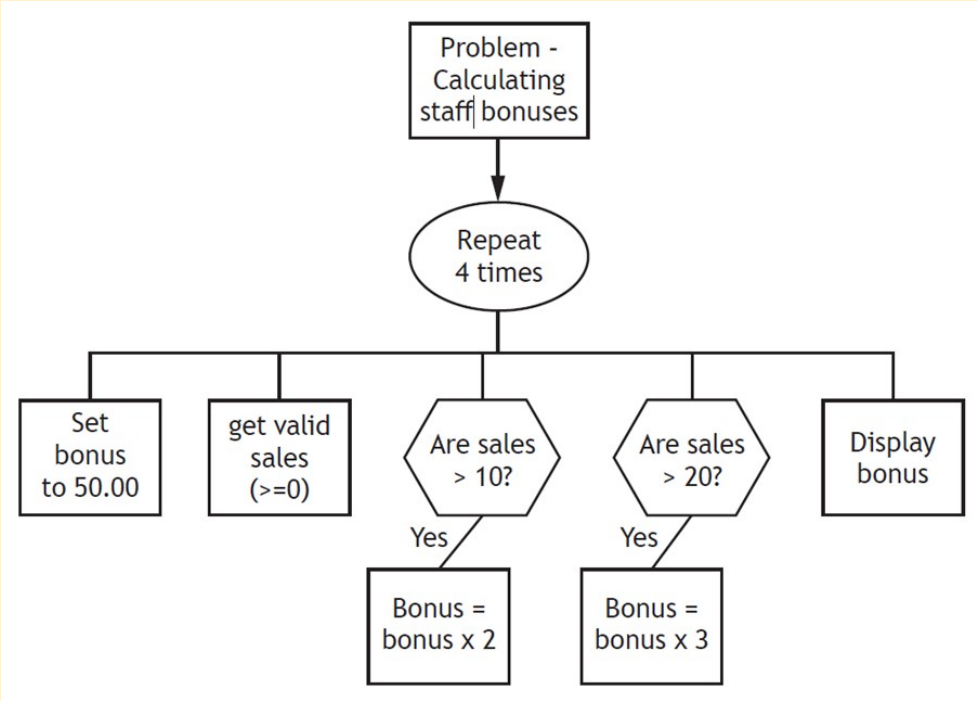
	Using the information above, design a user interface for the program. **(3)**
2018 Q19 (b)	A program is being designed that will allow pupils to add money to their lunch money account. The user enters their name, an 8 character password and the amount of money they want to add. A button is then clicked and the updated balance of the account is displayed. Design a user interface for this program**. (3)**

Implementation (data types and structures)

Describe, exemplify, and implement appropriately the following data types and structures:

- character
- string
- numeric (integer and real)
- Boolean
- 1D arrays

SQP Q14 (b)	Mark writes a program to calculate a worker's average weekly wage. The first part of the program asks the user to log in. They are given three attempts to enter the correct password which is 'Bingo'. ... Line 6 SET attempts TO 0 Line 7 REPEAT Line 8 RECEIVE password FROM KEYBOARD Line 9 SET attempts TO attempts +1 Line 10 UNTIL _____ ... (b) State the data type of the variable password. **(1)**
SQP Q21 (a)	Arthur's Antiques sells old furniture. All staff receive a monthly bonus of £50, which is increased if they sell over 10 items of furniture. The bonus is increased further if they sell over 20 items of furniture. A design for the program used to calculate the bonus payment for each of the four members of staff is shown below.



(a) List the variables and data types that would be required to implement the design.

The first one has been completed for you**. (2)**

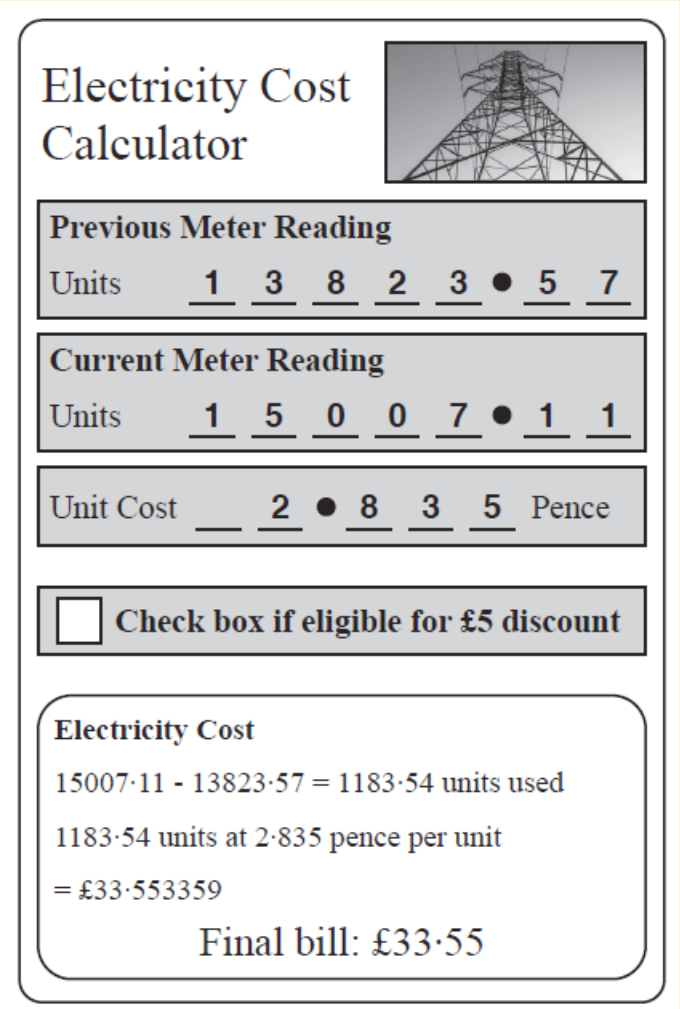
Variable Name	**Date Type**
Loop	Integer

2019 Q13 (c)

A smart phone app is needed to calculate the cost of electricity. The following information will be entered by the user.

- Previous meter reading
- Current meter reading
- Unit cost
- Discount eligibility

A possible user interface for the app is shown below.



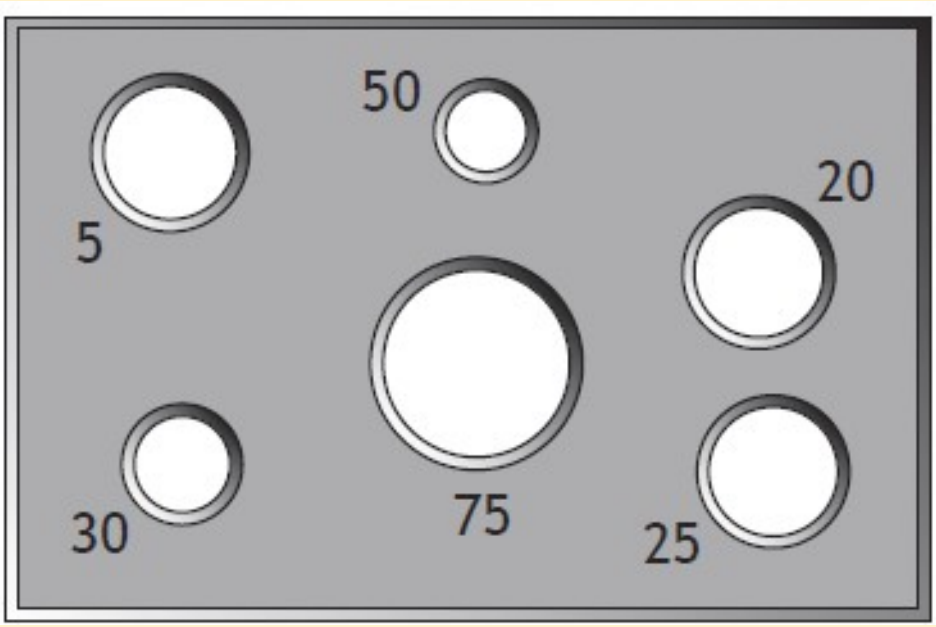
(c) State the data types that will be required to store the values of the following inputs. **(2)**

The current meter reading	
Check box if eligible for £5 discount	

2019 Q19 (c) (ii)

A fairground game involves throwing balls through holes in a large wooden board. Each hole scores different points.

The game is played using the following four rules.



1. A player starts with 3 balls and throws them one at a time.
2. If a ball is successfully thrown through a hole the points are added onto the player's score.
3. The game ends immediately if a player's score is greater than or equal to 50.
4. If the score reaches exactly 50 points the player is told they have won a prize.

A program is written to keep the score for a player.

```
...
Line 3    DECLARE total INITIALLY 0
Line 4    DECLARE balls INITIALLY 3
Line 5    WHILE total < 50 AND balls > 0 DO
Line 6       RECEIVE ballScoreOne FROM KEYBOARD
Line 7       SET total TO total + ballScoreOne
Line 8       SET balls TO balls - 1
Line 9       RECEIVE ballScoreTwo FROM KEYBOARD
Line 10      SET total TO total + ballScoreTwo
Line 11      SET balls TO balls - 1
Line 12      RECEIVE ballScoreThree FROM KEYBOARD
Line 13      SET total TO total + ballScoreThree
Line 14      SET balls TO balls - 1
Line 15   END WHILE
Line 16   SEND "Well done! You have won a prize." TO DISPLAY
```

(c) A single ball can achieve a variety of different possible scores.

Two versions of input validation were coded and tested to check that only valid scores are entered.

Version A

```
...
Line 6    RECEIVE ballScore FROM KEYBOARD
Line 7    WHILE ballScore < 0 OR ballScore > 75 DO
Line 8       RECEIVE ballScore FROM KEYBOARD
Line 9    END WHILE
```

Version B

```
Line 1    DECLARE possScore INITIALLY
          [0,5,20,25,30,50,75]
...
Line 6    DECLARE found AS BOOLEAN INITIALLY false
Line 7    REPEAT
Line 8       RECEIVE ballScore FROM KEYBOARD
Line 9       FOR check FROM 0 TO length(possScore)-1 DO
Line 10         IF possScore[check] = ballScore THEN
Line 11            SET found TO true
Line 12         END IF
Line 13      END FOR
Line 14   UNTIL found
```

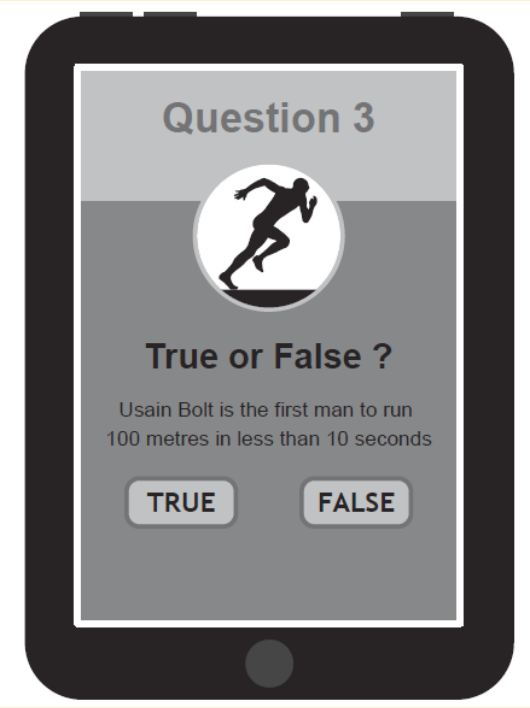
(ii) Name the data structure used in line 1 of Version B and state the data type that it is used to store. **(2)**

Name of data structure	
Data type stored	

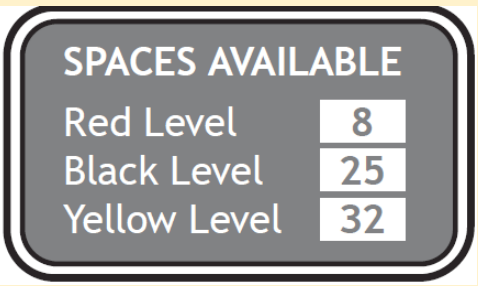
2018 Q17 (a) (i)

Scott is developing an online quiz with ten true or false questions. At the end of the quiz, the user's final score will be calculated.

(a) The user interface is shown below.

	 (i) Explain why a 1-D array of Boolean values is a suitable data structure to store the user's responses. (2)
2017 Q3	The validity of a password is checked as part of a program. . . . Line 8 SET passValid TO false Line 9 RECEIVE userPassword FROM (STRING) KEYBOARD Line10 IF userPassword = storedPassword THEN Line 11 SET passValid TO true Line 12 END IF . . . State the data type used to store the variable "passValid**". (1)**
2017 Q15 (a)	A program is being developed to monitor the availability of parking spaces in a multi-level car park. The car park has three levels, each with 50 numbered spaces and a digital display board that shows the number of spaces available on each level.

Level	Numbered Spaces
Red	1–50
Black	51–100
Yellow	101–150



Part of the program is shown below:

```
Line 1      DECLARE redAvailable AS INTEGER INITIALLY 50
Line 2      DECLARE blackAvailable AS INTEGER INITIALLY 50
Line 3      DECLARE yellowAvailable AS INTEGER INITIALLY 50
. . .
. . .
                < vehicle is detected occupying a space>
. . .
. . .

Line 22     IF spaceNumber ≥1 AND spaceNumber ≤50 THEN
Line 23         redAvailable = redAvailable – 1
Line 24     END IF

. . .
. . .
```

(a) Explain why integer data types are used in Lines 1 to 3. **(1)**

2016 Q16 (b)

A Maths game is designed for primary school pupils to test number ordering. In the game the pupil is asked to enter two integer numbers. A third integer number is then randomly generated and shown to the pupil.

The pupil must then state if the random number is:

lower (l) than the two entered numbers
higher (h) than the two entered numbers
in the middle (m) of the two entered numbers.

A design for the code is shown below.

```
Line 1    <enter the first number and assign to numOne>
Line 2    <enter the second number and assign to numTwo>
Line 3    <generate random number and assign to randNum>
Line 4    SEND randNum TO DISPLAY
Line 5    RECEIVE guess FROM (CHARACTER) KEYBOARD
Line 6    IF guess = "l" AND randNum < numOne THEN
Line 7        SEND "Correct it is lower" TO DISPLAY
Line 8        SET score TO score + 1
Line 9    END IF
Line 10   IF guess = "m" AND randNum >= numOne AND randNum <= numTwo
Line 11       SEND "Correct it is in the middle" TO DISPLAY
Line 12       SET score TO score + 1
Line 13   END IF
Line 14   IF guess = "h" AND randNum > numTwo
Line 15       SEND "Correct it is higher" TO DISPLAY
Line 16       SET score TO score + 1
Line 17   END IF
Line 18   <display incorrect message>
```

(b) When the pupil enters the answer it is stored in a variable called "guess".

State the data type stored by the variable "guess". **(1)**

2016 Q19 (a)

Gillian designs a program to calculate how much it costs to get her dog Penny groomed. The design is shown below.

```
Line 1    SET total = 0
Line 2    DECLARE all costs INITIALLY [35.00, 36.00, 40.00, 35.00, 42.50]
Line 3    FOR EACH cost FROM all costs DUE
Line 4        SET total=total+cost
Line 5    END FOR EACH
Line 6    SEND "The total cost = £"&total TO DISPLAY
```

(a) Describe the data structure that has been used to store the individual costs. **(2)**

Implementation (computational constructs)

Describe, exemplify, and implement the appropriate constructs in a high-level (textual) language:

- expressions to assign values
- expressions to return values using arithmetic operations (addition, subtraction, multiplication, division, and exponentiation)
- expressions to concatenate strings
- selection constructs using simple conditional statements with <, >, ≤, ≥, =, ≠ operators
- selection constructs using complex conditional statements
- logical operators (AND, OR, NOT)
- iteration and repetition using fixed and conditional loops
- predefined functions (with parameters):
  — random
  — round
  — length

Read and explain code that makes use of the above constructs.

SQP Q4	The code below monitors the speed of a vehicle: ... Line 5 RECEIVE speed FROM <sensor> Line 6 WHILE speed <= 70 DO Line 7 RECEIVE speed FROM <sensor> Line 8 END WHILE Line 9 SEND signal TO <alarm> Describe what happens in lines 6 to 9 above if the sensor detects a value of 83 at line 5. **(3)**
SQP Q14 (a) SQP Q14 (c)	Mark writes a program to calculate a worker's average weekly wage. The first part of the program asks the user to log in. They are given three attempts to enter the correct password which is 'Bingo'.

```
...
Line 6    SET attempts TO 0
Line 7    REPEAT
Line 8      RECEIVE password FROM KEYBOARD
Line 9      SET attempts TO attempts +1
Line 10   UNTIL _____
...
```

(a) Complete line 10 of the code above**. (3)**

Line 10	

The following section of code calculates the average weekly wage:

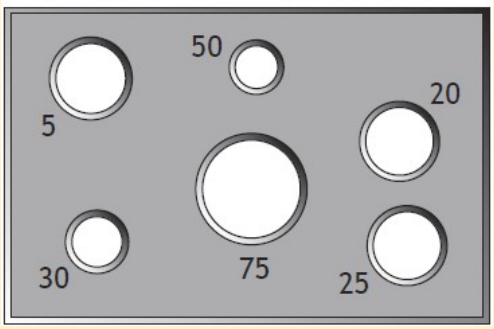
```
Line 11   RECEIVE day1 FROM KEYBOARD
Line 12   RECEIVE day2 FROM KEYBOARD
Line 13   RECEIVE day3 FROM KEYBOARD
Line 14   RECEIVE day4 FROM KEYBOARD
Line 15   RECEIVE day5 FROM KEYBOARD
Line 16   RECEIVE day6 FROM KEYBOARD
Line 17   RECEIVE day7 FROM KEYBOARD
Line 18   SET weeklyAverage TO (day1 + day2 + day3 + day4 +
          day5 + day6 + day7)/7
Line 19   <display the seven days wages and average>
```

(c) When evaluating this code, it is found to be inefficient.

Using a programming language of your choice, rewrite lines 11 to 18 of the code using more efficient constructs**. (5)**

SQP Q19 (c)

Read the following design for a solution to a problem.

	Algorithm 1 Ask the user to enter their name 2 Ask the user to enter their flight details 3 Generate the holiday booking reference 4 Display the holiday booking reference Refinements 1.1 Ask user to enter surname only 2.1 Ask user to enter first three letters of departure airport (for example: Edi for Edinburgh) 2.2 Ask user to enter first three letters of arrival airport 3.1 Store the booking reference as: arrival airport string + surname + departure airport string (c) Refinement 3.1 stores the holiday booking reference. State two programming constructs that would be required to implement this refinement. **(2)**
2019 Q19 (a) 2019 Q19 (c) (iii)	A fairground game involves throwing balls through holes in a large wooden board. Each hole scores different points. The game is played using the following four rules.  1. A player starts with 3 balls and throws them one at a time. 2. If a ball is successfully thrown through a hole the points are added onto the player's score. 3. The game ends immediately if a player's score is greater than or equal to 50. 4. If the score reaches exactly 50 points the player is told they have won a prize. A program is written to keep the score for a player.

```
...
Line 3    DECLARE total INITIALLY 0
Line 4    DECLARE balls INITIALLY 3
Line 5    WHILE total < 50 AND balls > 0 DO
Line 6       RECEIVE ballScoreOne FROM KEYBOARD
Line 7       SET total TO total + ballScoreOne
Line 8       SET balls TO balls - 1
Line 9       RECEIVE ballScoreTwo FROM KEYBOARD
Line 10      SET total TO total + ballScoreTwo
Line 11      SET balls TO balls - 1
Line 12      RECEIVE ballScoreThree FROM KEYBOARD
Line 13      SET total TO total + ballScoreThree
Line 14      SET balls TO balls - 1
Line 15   END WHILE
Line 16   SEND "Well done! You have won a prize." TO DISPLAY
```

(a) Identify one logical operator in the above code**. (1)**

(c) A single ball can achieve a variety of different possible scores.

Two versions of input validation were coded and tested to check that only valid scores are entered.

Version A

```
...
Line 6    RECEIVE ballScore FROM KEYBOARD
Line 7    WHILE ballScore < 0 OR ballScore > 75 DO
Line 8       RECEIVE ballScore FROM KEYBOARD
Line 9    END WHILE
```

Version B

```
Line 1    DECLARE possScore INITIALLY
          [0,5,20,25,30,50,75]
...
Line 6    DECLARE found AS BOOLEAN INITIALLY false
Line 7    REPEAT
Line 8       RECEIVE ballScore FROM KEYBOARD
Line 9       FOR check FROM 0 TO length(possScore)-1 DO
Line 10         IF possScore[check] = ballScore THEN
Line 11            SET found TO true
Line 12         END IF
Line 13      END FOR
Line 14   UNTIL found
```

(iii) Describe how the found variable is used in Version B. **(2)**

2018 Q7

The code for part of a program is shown below.

```
...
Line 41   SET runnerTime TO firstRaceTime +
          secondRaceTime + thirdRaceTime +
          fourthRaceTime + fifthRaceTime
Line 42   SET runnerAverage TO runnerTime / 5
Line 43   <display average to 2 decimal places>
...
```

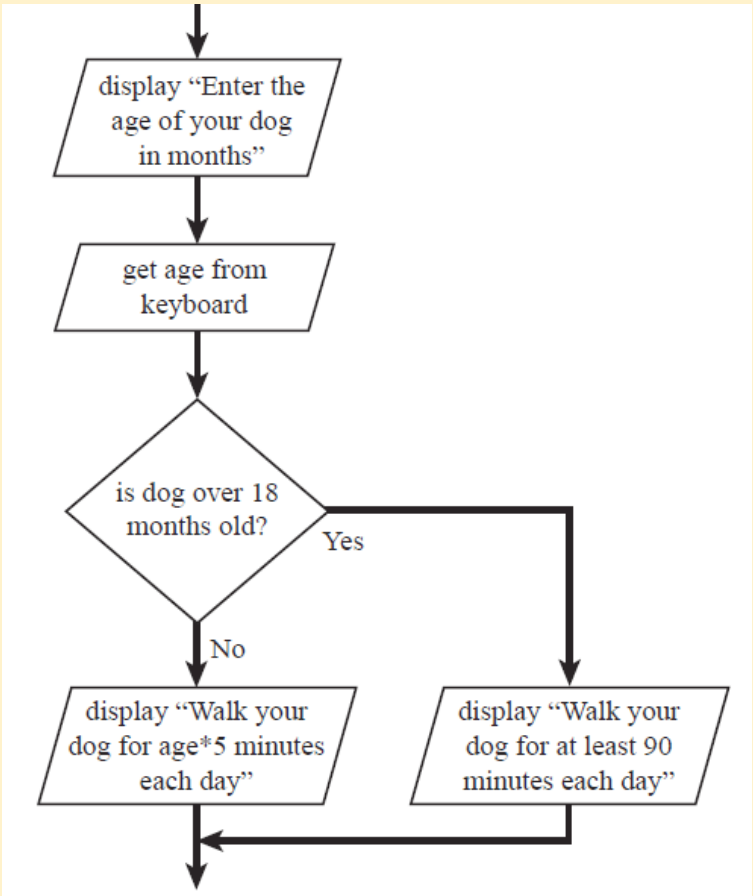
State the pre-defined function and a parameter that could be used in Line 43**. (2)**

Pre-defined function	
Parameter	

2018 Q11

The design shown below asks a user to enter the age of their dog. It then displays advice on how many minutes the dog should be walked each day.

Circle the condition in the design below**. (1)**



2018 Q14 (c)	The program code below calculates the delivery cost of orders. ... Line 13 IF orderTotal < 50.00 AND NOT(cardType = "Platinum") THEN Line 14 SET deliveryCost TO 5.00 Line 15 ELSE Line 16 SET delivery TO 1.50 Line 17 END IF Line 18 SEND deliveryCost TO DISPLAY ... (c) State the delivery cost for the following order. **(1)** Card Type: Gold Order Total: 43.00
2018 Q15	Explain why a conditional loop would be used when writing code**. (1)**
2018 Q19 (c) (i) 2018 Q19 (c) (ii)	A program is being designed that will allow pupils to add money to their lunch money account. The user enters their name, an 8 character password and the amount of money they want to add. A button is then clicked and the updated balance of the account is displayed. (c) The password must contain 8 characters. (i) State a suitable pre-defined function to check that the password contains 8 characters. **(1)** (ii) Explain why a pre-defined function would be used. **(1)**
2018 Q21 (b) (i)	A program will calculate the total cost when customers purchase tickets to a theme park. Adults pay £25 per ticket; children pay £10. If there are two or more adults with more than two children a discount of £5 is subtracted from the total cost.

Algorithm

1. Store cost of adult and child ticket
2. Get name of person making booking
3. Get quantity of tickets
4. Calculate total cost
5. Display food voucher message

Refinement

2.1 Get first name
2.2 Get second name

3.1 Get quantity of adult tickets
3.2 Get quantity of child tickets

(b) Customers who spend £50 or more on tickets qualify for a number of food vouchers.

Step 5 of the algorithm has been implemented below.

```
...
Line 23   IF totalCost < 50 THEN
Line 24       SEND "Sorry, no food voucher" TO DISPLAY
Line 25   ELSE
Line 26       IF totalCost >100 THEN
Line 27           SEND "You have been awarded 2 food
                  vouchers" TO DISPLAY
Line 28       ELSE
Line 29           SEND "You have been awarded 1 food
                  voucher" TO DISPLAY
Line 30       END IF
Line 31   END IF
...
```

(i) State the output if:

(A) the total cost is 104; **(1)**

(B) the total cost is 50. **(1)**

2017	Part of a program is shown below.

Q7

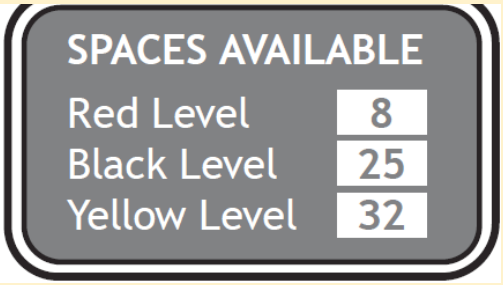
```
Line 1:     DECLARE score AS REAL INITIALLY 0·0
Line 2:     RECEIVE score FROM KEYBOARD
Line 3:     IF score > 2·0 THEN
Line 4:         SEND "Congratulations. You are in the final" TO DISPLAY
Line 5:     ELSE
Line 6:         SEND "You have failed to qualify" TO DISPLAY
Line 7:     END IF
```

Describe what happens in Lines 3 to 6 when the value 1·4 is entered at Line 2. **(2)**

2017 Q15 (c) (i)

A program is being developed to monitor the availability of parking spaces in a multi-level car park. The car park has three levels, each with 50 numbered spaces and a digital display board that shows the number of spaces available on each level.

Level	Numbered Spaces
Red	1–50
Black	51–100
Yellow	101–150



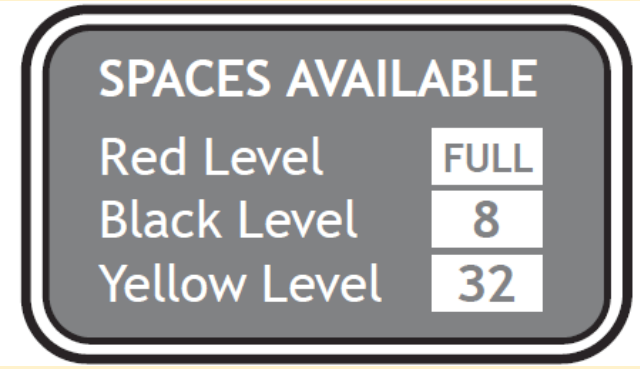
Part of the program is shown below:

```
Line 1      DECLARE redAvailable AS INTEGER INITIALLY 50
Line 2      DECLARE blackAvailable AS INTEGER INITIALLY 50
Line 3      DECLARE yellowAvailable AS INTEGER INITIALLY 50
. . .
. . .
                < vehicle is detected occupying a space>
. . .
. . .

Line 22     IF spaceNumber ≥1 AND spaceNumber ≤50 THEN
Line 23         redAvailable = redAvailable – 1
Line 24     END IF

. . .
. . .
```

When a vehicle parks, the digital display board will be updated to show the number of available spaces on each level.



(c) (i) Complete the condition below, that will display the message "FULL" when all the spaces on the Red Level are occupied. **(1)**

```
IF_____THEN
    SEND "FULL" TO DISPLAY
END IF
```

2017 Q19 (b)

Louise is conducting a survey at her school to find out how many hours per week her class mates spend playing computer games. Louise will survey 100 pupils.

The program assigns 100 names to a 1-D array as shown below.

```
Line 1      DECLARE name AS ARRAY OF STRING INITIALLY []
Line 2      RECEIVE name[0] FROM KEYBOARD
Line 3      RECEIVE name[1] FROM KEYBOARD
Line 4      RECEIVE name[2] FROM KEYBOARD
. . .
. . .
Line 101    RECEIVE name[99] FROM KEYBOARD
```

(b) Another section of the program is shown below.

```
. . .
Line 119    SET averageHours = totalHours / 7
Line 120    <use a pre-defined function to store averageHours to the
            nearest whole number>
Line 121    SEND "An average of " & averageHours & " hours" TO
            DISPLAY
```

(i) Identify the operator used to concatenate in the program above. (1)

(ii) Explain why averageHours should be stored as a real data type. (1)

	(iii) The program is executed. At Line 119 the value 4·26 is assigned to averageHours. Write the message that will be displayed when Line 121 is executed**. (2)** (iv) State the pre-defined function that could be used when Line 120 is coded. **(1)**
2016 Q9	This code design monitors the temperature of food as it is reheated. Line 1 RECEIVE temperature FROM (REAL) <*temperature sensor*> Line 2 WHILE temperature < 82 DO Line 3 SEND "temperature too low: continue to reheat" TO DISPLAY Line 4 RECEIVE temperature FROM (REAL) <*temperature sensor*> Line 5 END WHILE Explain what will happen in lines 2 to 5 if the sensor detects 63°. **(2)**
2016 Q12	A running group has 16 members. They are taking part in a marathon. Using pseudocode or a programming language of your choice, write the code which will take in each runner's time for the marathon. **(2)**
2016 Q16 (d) 2016 Q16 (e)	A Maths game is designed for primary school pupils to test number ordering. In the game the pupil is asked to enter two integer numbers. A third integer number is then randomly generated and shown to the pupil. The pupil must then state if the random number is: lower (l) than the two entered numbers higher (h) than the two entered numbers in the middle (m) of the two entered numbers. A design for the code is shown below.

```
Line 1    <enter the first number and assign to numOne>
Line 2    <enter the second number and assign to numTwo>
Line 3    <generate random number and assign to randNum>
Line 4    SEND randNum TO DISPLAY
Line 5    RECEIVE guess FROM (CHARACTER) KEYBOARD
Line 6    IF guess = "l" AND randNum < numOne THEN
Line 7        SEND "Correct it is lower" TO DISPLAY
Line 8        SET score TO score + 1
Line 9    END IF
Line 10   IF guess = "m" AND randNum >= numOne AND randNum <= numTwo
Line 11       SEND "Correct it is in the middle" TO DISPLAY
Line 12       SET score TO score + 1
Line 13   END IF
Line 14   IF guess = "h" AND randNum > numTwo
Line 15       SEND "Correct it is higher" TO DISPLAY
Line 16       SET score TO score + 1
Line 17   END IF
Line 18   <display incorrect message>
```

(d) The program will have to make use of a pre-defined function.

State the pre-defined function used and describe its purpose. **(2)**

(e) Using line numbers, describe how the code could be adapted, allowing the pupil to play the game 10 times using the same values for numOne and numTwo but a different random number each time. **(2)**

2016 Q18 (d) (ii)

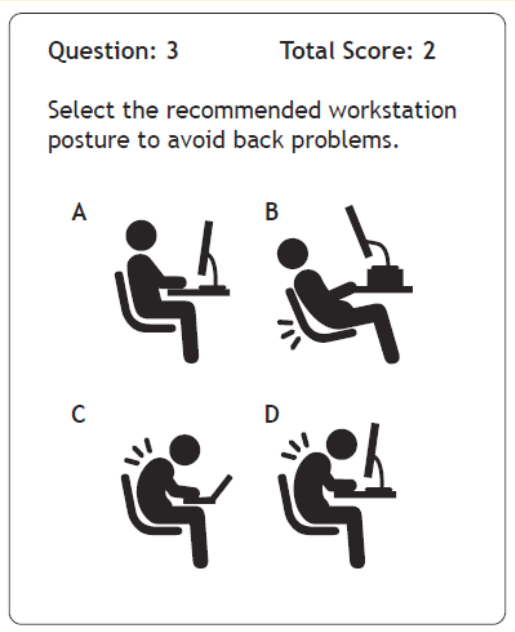
A software development company decides to review staff knowledge of computer related legislation.

Mikal is asked to create an app covering a range of legal issues.

(d) In line with Health and Safety legislation, the company provides adjustable seating and guidelines on maintaining good posture.

Mikal finds graphics on a website that he can use to illustrate his next quiz question.

(ii) Mikal uses the graphics to create question 3 for the app.



Using pseudocode or a programming language of your choice, write the code to show how the total score is calculated when the user answers question 3 correctly. **(2)**

2016 Q19 (b) (i)

2016 Q19 (b) (iii)

2016 Q19 (c)

Gillian designs a program to calculate how much it costs to get her dog Penny groomed. The design is shown below.

```
Line 1      SET total = 0
Line 2      DECLARE all costs INITIALLY [35.00, 36.00, 40.00, 35.00, 42.50]
Line 3      FOR EACH cost FROM all costs DUE
Line 4          SET total=total+cost
Line 5      END FOR EACH
Line 6      SEND "The total cost = £"&total TO DISPLAY
```

(b) Gillian writes and tests her program. It works perfectly calculating a correct total of 188.50.

	(i) With reference to line numbers, explain how the program calculates the final total. (iii) Gillian edits the program with the following data: [35.00, 36.00, 40.00, 35.00, 42.50, 45.00] The output is still 188.50. A Explain why the output is still 188.50. **(1)** B State how this error could be corrected**. (1)** (c) Concatenation has been used in line 6. State the purpose of concatenation**. (1)**
2016 Q20 (b)	Sue uses a website called "Check your Defences!" to learn more about keeping her computer and data safe  (a) Explain the purpose of a firewall**. (1)** (b) Explain how encryption can help keep data safe**. (2)**

Implementation (algorithm specification)

Describe, exemplify, and implement standard algorithms:

- input validation
- running total within loop
- traversing a 1D array

<table>
<tr>
<td>SQP
Q19 (e)</td>
<td>Read the following design for a solution to a problem.

Algorithm
1 Ask the user to enter their name
2 Ask the user to enter their flight details
3 Generate the holiday booking reference
4 Display the holiday booking reference

Refinements
1.1 Ask user to enter surname only

2.1 Ask user to enter first three letters of departure airport (for example: Edi for Edinburgh)
2.2 Ask user to enter first three letters of arrival airport

3.1 Store the booking reference as: arrival airport string + surname + departure airport string

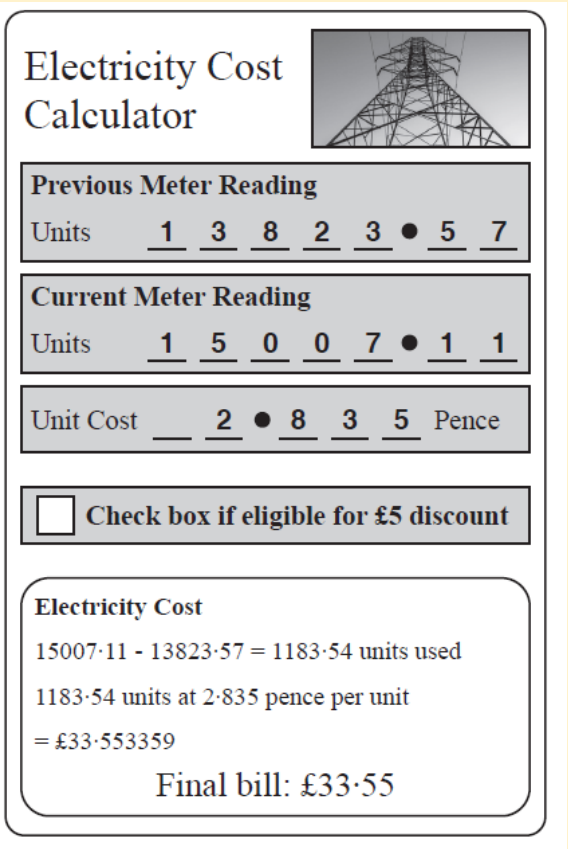
(e) Using a design technique of your choice, add input validation to refinement 2.1 to ensure that the user only enters a 3 character string.

An error message should inform the user when their input is not valid. (4)</td>
</tr>
<tr>
<td>2019
Q7 (b)</td>
<td>Part of a program requires a user to input the total score achieved when they roll a pair of six-sided dice.

For example, if the user rolled a 4 and a 1 they would input 5.

(b) The code below shows part of the program.</td>
</tr>
</table>

	…
	FOR loop FROM 1 TO 100 DO
	RECEIVE diceScore FROM KEYBOARD
	SET total TO total + diceScore
	END FOR
	…
	
	State the standard algorithm shown above. **(1)**

2019 Q13 (e)	A smart phone app is needed to calculate the cost of electricity. The following information will be entered by the user. • Previous meter reading • Current meter reading • Unit cost • Discount eligibility A possible user interface for the app is shown below.  (e) The program uses input validation.

```
...
Line 13  REPEAT
Line 14    RECEIVE currentReading FROM <the touch
           screen keyboard>
Line 15    IF currentReading < previousReading THEN
Line 16      SEND "Reading too low. Please re-enter"
             TO DISPLAY
Line 17    END IF
Line 18  _____________________________________________
```

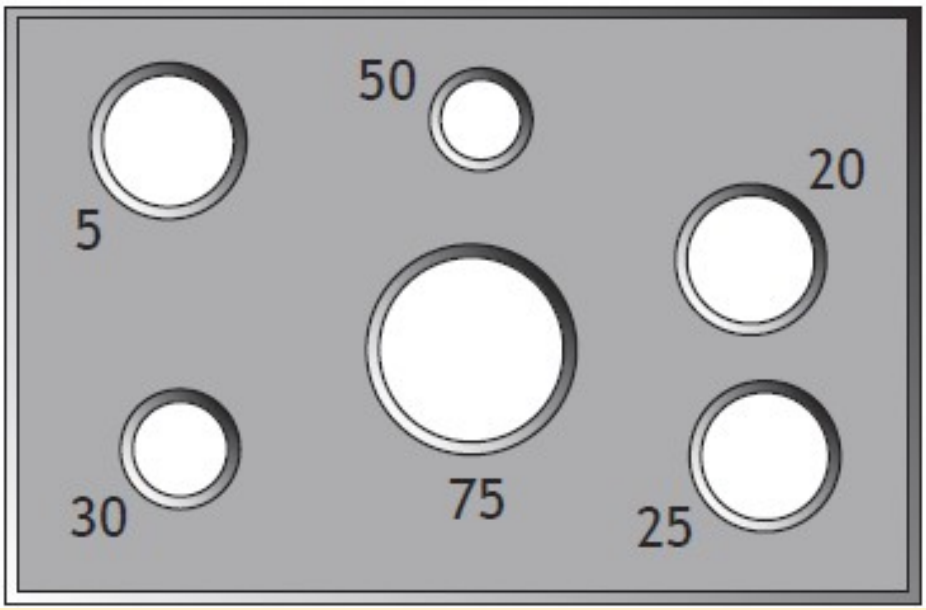
Using a programming language of your choice, complete Line 18.

Ensure that only acceptable values can be entered for the current meter reading. **(2)**

2019 Q19 (c) (i)

A fairground game involves throwing balls through holes in a large wooden board. Each hole scores different points.

The game is played using the following four rules.



1. A player starts with 3 balls and throws them one at a time.
2. If a ball is successfully thrown through a hole the points are added onto the player's score.
3. The game ends immediately if a player's score is greater than or equal to 50.
4. If the score reaches exactly 50 points the player is told they have won a prize.

A program is written to keep the score for a player.

```
...
Line 3    DECLARE total INITIALLY 0
Line 4    DECLARE balls INITIALLY 3
Line 5    WHILE total < 50 AND balls > 0 DO
Line 6       RECEIVE ballScoreOne FROM KEYBOARD
Line 7       SET total TO total + ballScoreOne
Line 8       SET balls TO balls - 1
Line 9       RECEIVE ballScoreTwo FROM KEYBOARD
Line 10      SET total TO total + ballScoreTwo
Line 11      SET balls TO balls - 1
Line 12      RECEIVE ballScoreThree FROM KEYBOARD
Line 13      SET total TO total + ballScoreThree
Line 14      SET balls TO balls - 1
Line 15   END WHILE
Line 16   SEND "Well done! You have won a prize." TO DISPLAY
```

(c) A single ball can achieve a variety of different possible scores.

Two versions of input validation were coded and tested to check that only valid scores are entered.

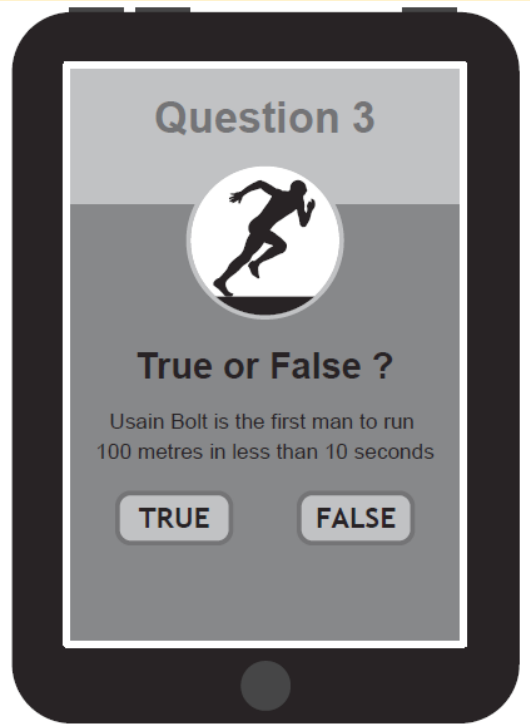
Version A

```
...
Line 6    RECEIVE ballScore FROM KEYBOARD
Line 7    WHILE ballScore < 0 OR ballScore > 75 DO
Line 8       RECEIVE ballScore FROM KEYBOARD
Line 9    END WHILE
```

Version B

```
Line 1    DECLARE possScore INITIALLY
          [0,5,20,25,30,50,75]
...
Line 6    DECLARE found AS BOOLEAN INITIALLY false
Line 7    REPEAT
Line 8       RECEIVE ballScore FROM KEYBOARD
Line 9       FOR check FROM 0 TO length(possScore)-1 DO
Line 10         IF possScore[check] = ballScore THEN
Line 11            SET found TO true
Line 12         END IF
Line 13      END FOR
Line 14   UNTIL found
```

(i) Explain why it would not be appropriate to use the input validation shown in Version A. **(1)**

2018 Q17 (a) (ii)	Scott is developing an online quiz with ten true or false questions. At the end of the quiz, the user's final score will be calculated. (a) The user interface is shown below.  (ii) For each correct response, 5 points are added to the user's score. Using a programming language of your choice, write efficient code to calculate the user's final score. Your code should use a running total within a loop. **(4)**
2018 Q21 (b) (ii)	A program will calculate the total cost when customers purchase tickets to a theme park. Adults pay £25 per ticket; children pay £10. If there are two or more adults with more than two children a discount of £5 is subtracted from the total cost.

Algorithm

1. Store cost of adult and child ticket
2. Get name of person making booking
3. Get quantity of tickets
4. Calculate total cost
5. Display food voucher message

Refinement

2.1 Get first name
2.2 Get second name

3.1 Get quantity of adult tickets
3.2 Get quantity of child tickets

(b) Customers who spend £50 or more on tickets qualify for a number of food vouchers.

Step 5 of the algorithm has been implemented below.

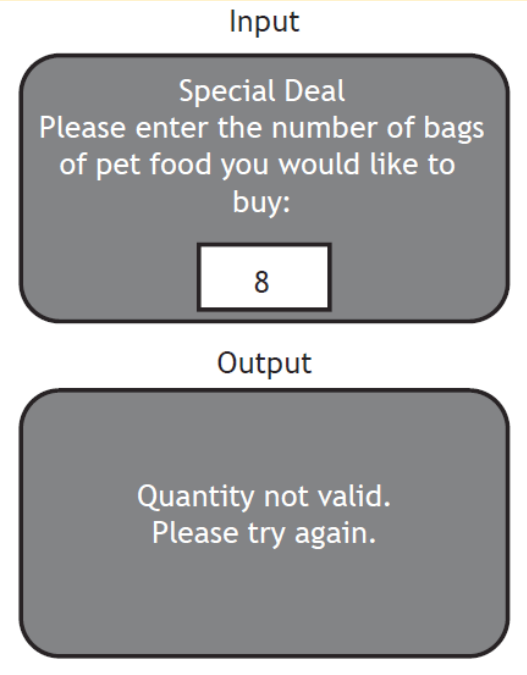
```
...
Line 23   IF totalCost < 50 THEN
Line 24       SEND "Sorry, no food voucher" TO DISPLAY
Line 25   ELSE
Line 26       IF totalCost >100 THEN
Line 27           SEND "You have been awarded 2 food
                  vouchers" TO DISPLAY
Line 28       ELSE
Line 29           SEND "You have been awarded 1 food
                  voucher" TO DISPLAY
Line 30       END IF
Line 31   END IF
...
```

(ii) When the completed code is tested, a user enters 2.5 for the number of adult tickets.

The program continues to run and calculates the total cost.

Explain how the program could be made fit for purpose. **(1)**

2017	An online pet supply retailer is offering a special deal to customers buying at

Q17 (a)	least two, but not more than six, bags of pet food. If customers try to buy any other quantity, a message is displayed. For example:  (a) Show, using pseudocode or a programming language of your choice, how input validation could be used to ensure an acceptable number of bags is entered. **(4)**
2017 Q19 (a)	Louise is conducting a survey at her school to find out how many hours per week her class mates spend playing computer games. Louise will survey 100 pupils. The program assigns 100 names to a 1-D array as shown below. Line 1 DECLARE name AS ARRAY OF STRING INITIALLY [] Line 2 RECEIVE name[0] FROM KEYBOARD Line 3 RECEIVE name[1] FROM KEYBOARD Line 4 RECEIVE name[2] FROM KEYBOARD . . . . . . Line 101 RECEIVE name[99] FROM KEYBOARD (a) Louise realises that writing the code to read the data into the array like this is time consuming and not good practice.

	Write, using pseudocode or a programming language of your choice, the code to show how the data can be entered into the 1-D array using repetition. **(3)**
2016 Q16 (a)	A Maths game is designed for primary school pupils to test number ordering. In the game the pupil is asked to enter two integer numbers. A third integer number is then randomly generated and shown to the pupil. The pupil must then state if the random number is: lower (l) than the two entered numbers higher (h) than the two entered numbers in the middle (m) of the two entered numbers. A design for the code is shown below. Line 1 <enter the first number and assign to numOne> Line 2 <enter the second number and assign to numTwo> Line 3 <generate random number and assign to randNum> Line 4 SEND randNum TO DISPLAY Line 5 RECEIVE guess FROM (CHARACTER) KEYBOARD Line 6 IF guess = "l" AND randNum < numOne THEN Line 7 SEND "Correct it is lower" TO DISPLAY Line 8 SET score TO score + 1 Line 9 END IF Line 10 IF guess = "m" AND randNum >= numOne AND randNum <= numTwo Line 11 SEND "Correct it is in the middle" TO DISPLAY Line 12 SET score TO score + 1 Line 13 END IF Line 14 IF guess = "h" AND randNum > numTwo Line 15 SEND "Correct it is higher" TO DISPLAY Line 16 SET score TO score + 1 Line 17 END IF Line 18 <display incorrect message> (a) When the two numbers are entered the program should ensure that numTwo is always a higher number than numOne. Using pseudocode or a programming language of your choice, write several lines to represent this input validation for line 2. **(4)**

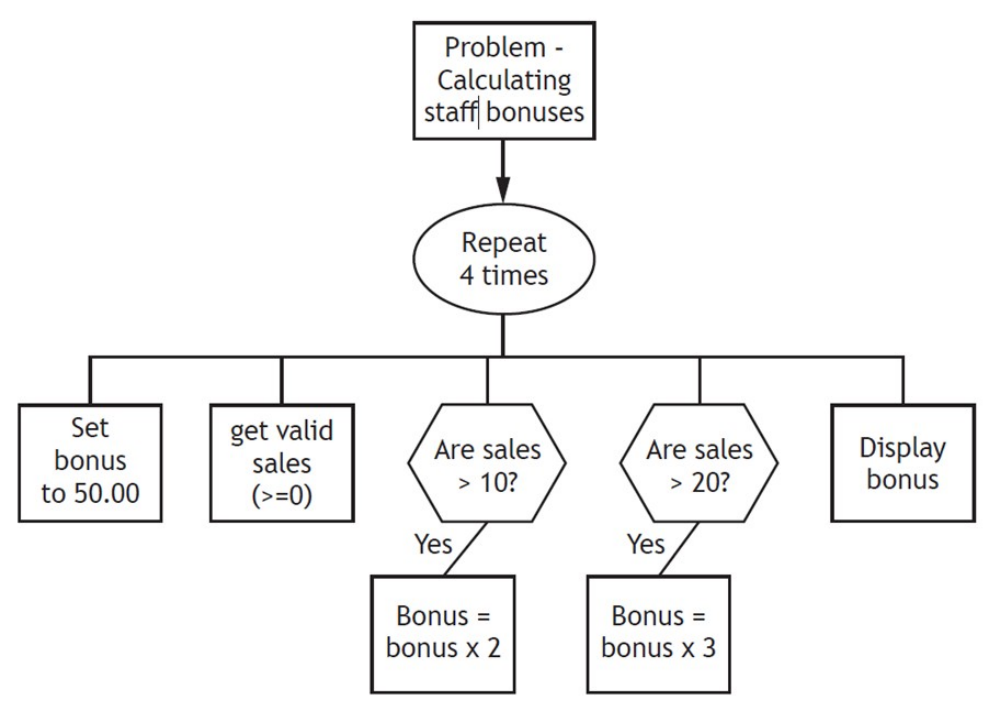
Testing

Describe, identify, exemplify, and implement normal, extreme, and exceptional test data for a specific problem, using a test table.

SQP Q21(c) (i)

Arthur's Antiques sells old furniture. All staff receive a monthly bonus of £50, which is increased if they sell over 10 items of furniture. The bonus is increased further if they sell over 20 items of furniture.

A design for the program used to calculate the bonus payment for each of the four members of staff is shown below.

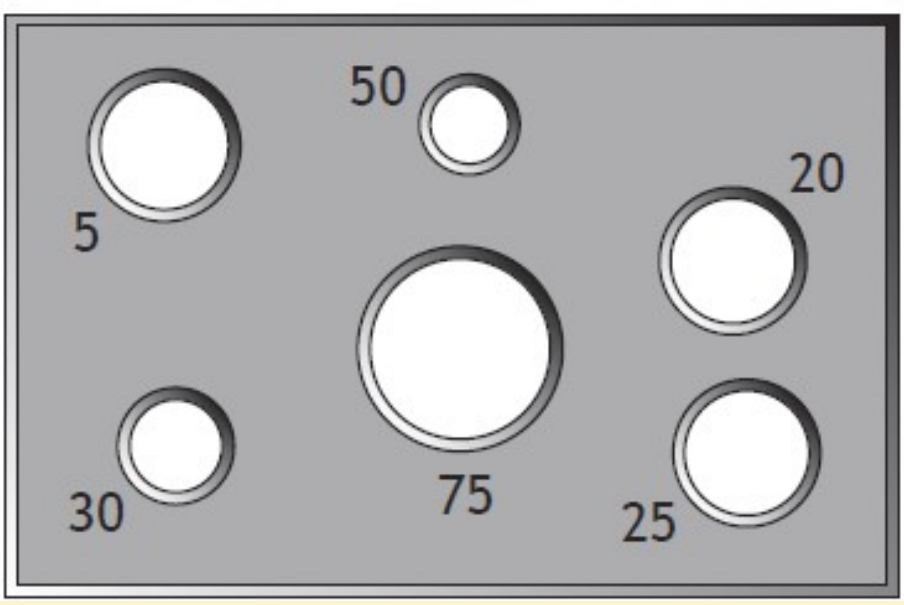


(c) The program is further tested with normal test data. The results are shown below.

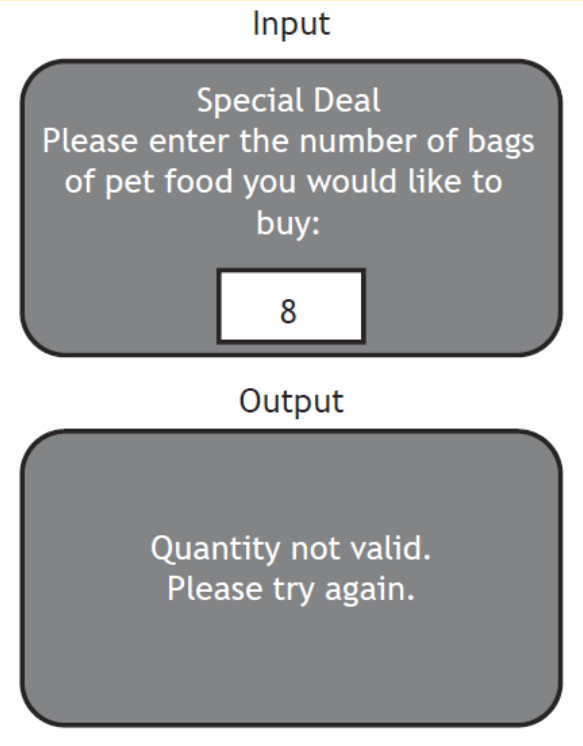
	Sales input	Expected output	Actual output
Staff 1	6	Bonus is 50	Bonus is 50
Staff 2	10	Bonus is 50	Bonus is 50
Staff 3	15	Bonus is 100	Bonus is 100
Staff 4	22	Bonus is 150	**Bonus is 300**

The test data for Staff 4 shows there is an error in the design.

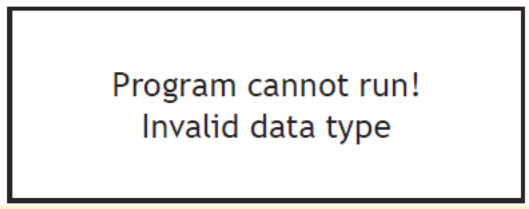
(i) State the type of error. **(1)**

2019 Q11	The programming language below uses & to concatenate two strings. `SET message TO "hello" & "world"` When coding, a programmer types £ instead of & leading to an error. State the type of programming error and describe its effect. **(2)** Type Effect
2019 Q19 (b) (i)	A fairground game involves throwing balls through holes in a large wooden board. Each hole scores different points. The game is played using the following four rules.  1. A player starts with 3 balls and throws them one at a time. 2. If a ball is successfully thrown through a hole the points are added onto the player's score. 3. The game ends immediately if a player's score is greater than or equal to 50. 4. If the score reaches exactly 50 points the player is told they have won a prize. A program is written to keep the score for a player.

	... Line 3 DECLARE total INITIALLY 0 Line 4 DECLARE balls INITIALLY 3 Line 5 WHILE total < 50 AND balls > 0 DO Line 6 RECEIVE ballScoreOne FROM KEYBOARD Line 7 SET total TO total + ballScoreOne Line 8 SET balls TO balls - 1 Line 9 RECEIVE ballScoreTwo FROM KEYBOARD Line 10 SET total TO total + ballScoreTwo Line 11 SET balls TO balls - 1 Line 12 RECEIVE ballScoreThree FROM KEYBOARD Line 13 SET total TO total + ballScoreThree Line 14 SET balls TO balls - 1 Line 15 END WHILE Line 16 SEND "Well done! You have won a prize." TO DISPLAY (b) The program runs but does not meet the functional requirements stated in the rules. (i) State the type of error that has occurred. **(1)**
2018 Q14 (a) 2018 Q14 (b)	The program code below calculates the delivery cost of orders. ... Line 13 IF orderTotal < 50.00 AND NOT(cardType = "Platinum") THEN Line 14 SET deliveryCost TO 5.00 Line 15 ELSE Line 16 SET delivery TO 1.50 Line 17 END IF Line 18 SEND deliveryCost TO DISPLAY ... (a) Explain why the program may not display the expected output at line 18**. (1)** (b) Identify one logical operator in the above code**. (1)**
2017 Q17 (c) 2017 Q17 (d) (i)	An online pet supply retailer is offering a special deal to customers buying at least two, but not more than six, bags of pet food. If customers try to buy any other quantity, a message is displayed. For example:



(c) When testing the program using the data from the table, “Three” is entered. As expected, an error message appears.



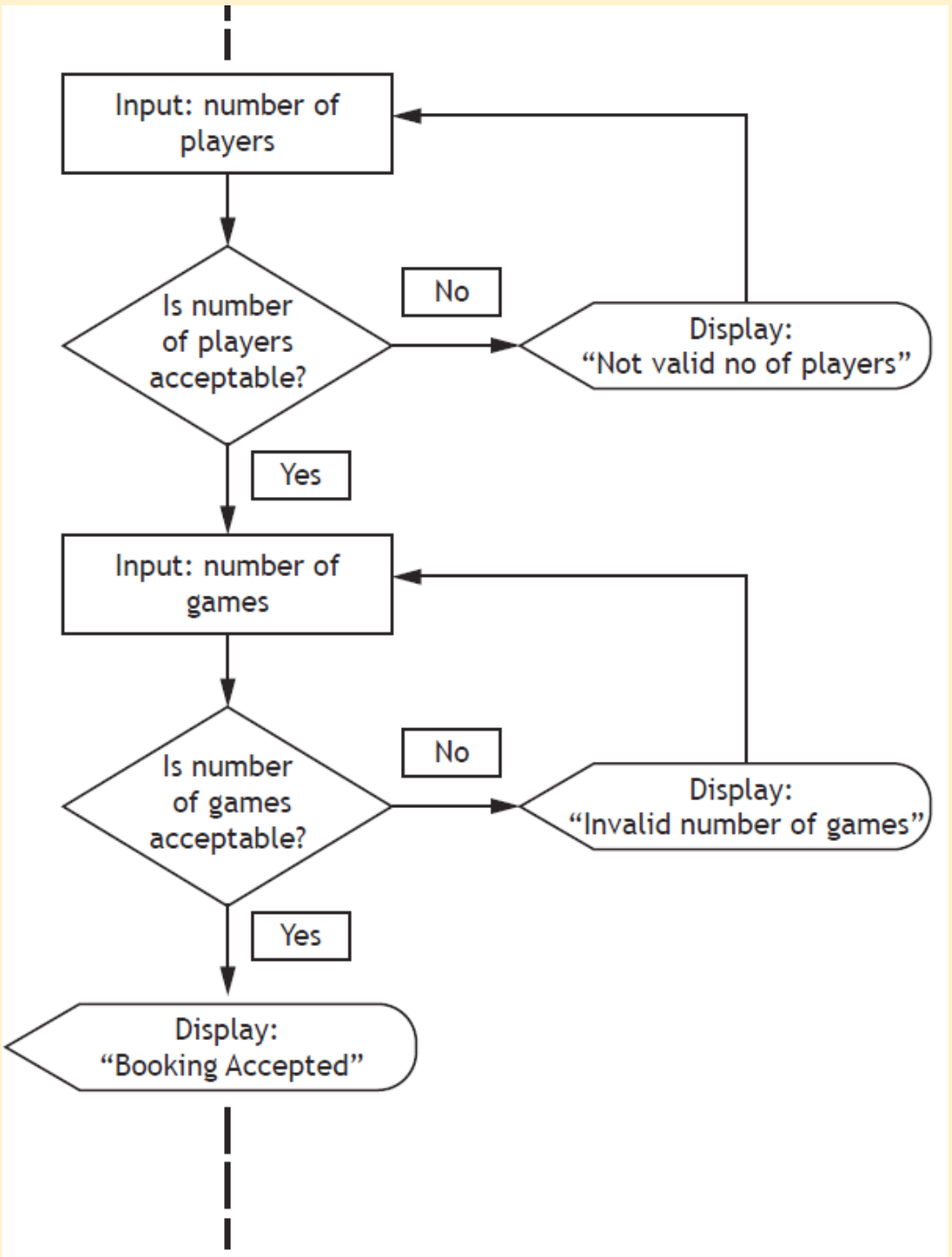
(i) Name this type of error. **(1)**

(ii) Explain why this error occurred when testing the program. **(1)**

(d) A syntax error can occur when writing code.

(i) Explain what is meant by a syntax error**. (1)**

2016 Q21 (c) (ii)	A software developer is creating an online booking system for a bowling alley. Customers can book a bowling lane for a maximum of 4 people playing a maximum of 3 games. The developer has used a flow chart to produce the program design. Part of the design is shown below.



(c) The program is tested using a set of test data.

(ii) The character "£" is entered as a test value for the number of players. This causes the program to crash.

State the type of error that would cause this crash**. (1)**

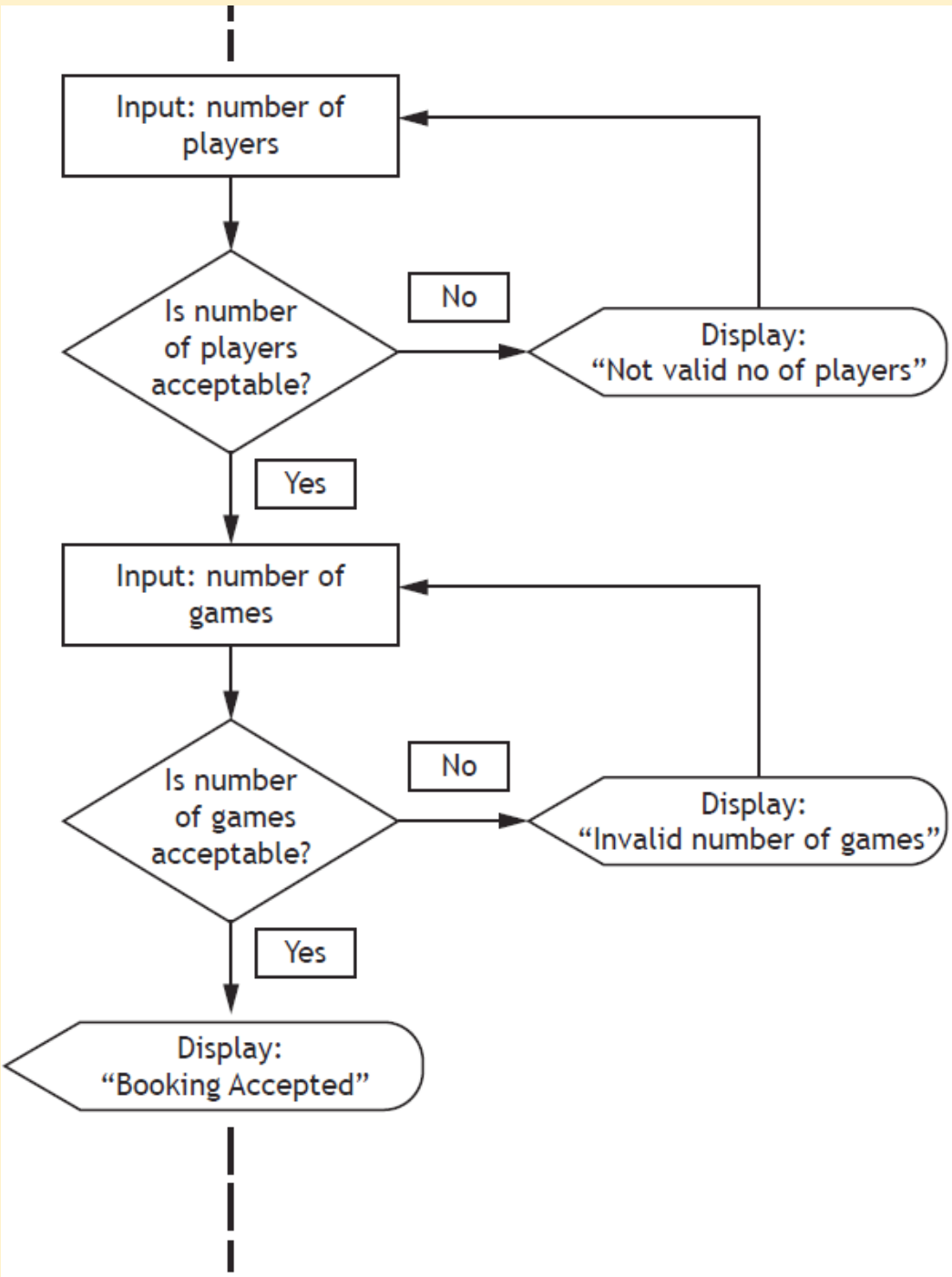
Evaluation

Describe, identify, and exemplify the evaluation of a solution in terms of:

- fitness for purpose
- efficient use of coding constructs
- robustness
- readability:
  — internal commentary
  — meaningful identifiers
  — indentation
  — white space

2017
Q19 (c)
	Louise is conducting a survey at her school to find out how many hours per week her class mates spend playing computer games. Louise will survey 100 pupils.
	
	The program assigns 100 names to a 1-D array as shown below.
	
	`Line 1 DECLARE name AS ARRAY OF STRING INITIALLY []`
	`Line 2 RECEIVE name[0] FROM KEYBOARD`
	`Line 3 RECEIVE name[1] FROM KEYBOARD`
	`Line 4 RECEIVE name[2] FROM KEYBOARD`
	`. . .`
	`. . .`
	`Line 101 RECEIVE name[99] FROM KEYBOARD`
	
	(c) Louise gives a copy of her finished program to her friend who tells her that the program code is difficult to read.
	
	(i) Explain how indentation can help readability in the program**. (1)**
	
	(ii) State one other programming technique used to improve readability of programs. (1)

2016 Q2	Explain why it is important that program code is readable**. (1)**
2016 Q21 (d)	A software developer is creating an online booking system for a bowling alley. Customers can book a bowling lane for a maximum of 4 people playing a maximum of 3 games. The developer has used a flow chart to produce the program design. Part of the design is shown below.



(d) Error detection and correction in a program is easier if the code is readable.

State one technique that can be used to ensure readability of code. **(1)**