

### Example 1.1 – Pseudocode

iGCSE Computer Science, SAM Paper 1, Q7(a)

- 7 Algorithms can be designed using pseudocode or flowcharts. Then, they need to be translated into code that a computing device can execute.

Figure 2 shows the pseudocode for an algorithm.

```

1  # This is the pseudocode for an algorithm
2  SET inNum TO 0
3  SET result TO 1
4  SET i TO 0
5
6  SEND "Enter a number: " TO DISPLAY
7  RECEIVE inNum FROM (INTEGER) KEYBOARD
8
9  IF (inNum < 0) THEN
10     SEND "Invalid input" TO DISPLAY
11 ELSE
12     IF (inNum = 0) THEN
13         SEND "Answer is 1" TO DISPLAY
14     ELSE
15         FOR i FROM 1 TO inNum DO
16             SET result TO result * i
17         END FOR
18         SEND "The answer is " & result TO DISPLAY
19     ENDIF
20 ENDIF
    
```

**Figure 2**

- (a) Use the information in Figure 2 to answer these questions.

- (i) Complete the table to show the output for the given input.

(3)

Input	Output message
0	
-12	
5	

- (ii) State the purpose of this algorithm.

(1)

.....

.....

.....

### Example 1.2 – Flowcharts

GCSE Computer Science (2016), SAM Paper 2, Q10(a)

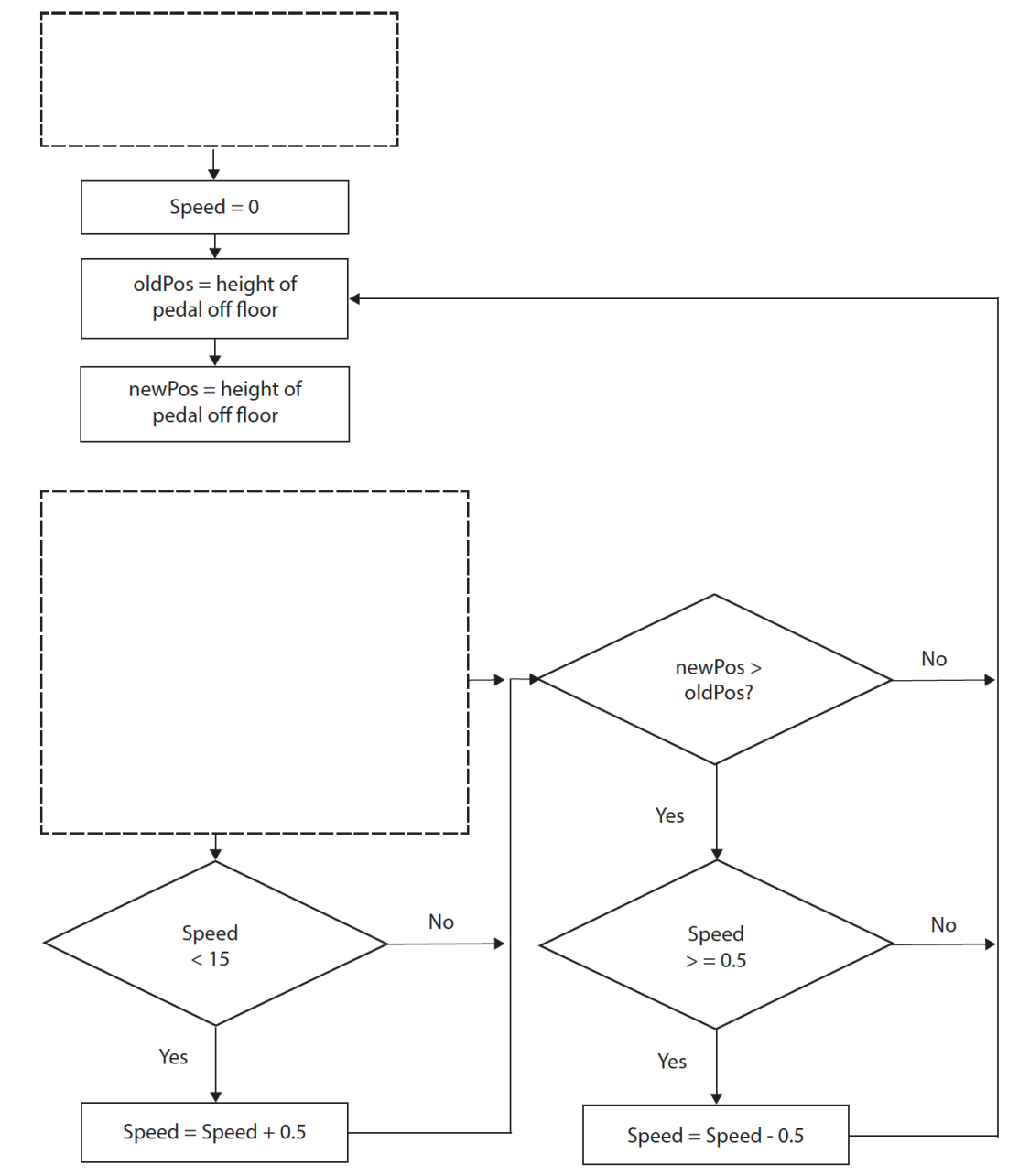
10 Each car at Sparky Autos has a pedal to make the car go forward.

(a) The drivers can make the car go forward by using the pedal.

- The closer the pedal is to the floor, the faster the car goes.
- The further the pedal is from the floor, the slower the car goes.
- Each car has its speed limited to 15 kilometres per hour.

Complete the flow chart to show this process.

(6)



### **Example 1.3 – Flowcharts**

*iGCSE Computer Science, SAM Paper 1, Q7(b)*

(b) A bus company sets fares for different groups of passengers.

The fares are:

- a child fare for passengers 15 years old and younger
- a senior fare for passengers 65 years old and older
- a full fare for all other passengers.

Construct a flowchart of an algorithm that will determine the fare for one passenger when an age is input.

No validation of input is required.

(5)

### Example 1.4 - Written descriptions

GCSE Computer Science (2016), Specimen Paper 2, Q6(b)

- 6 The HappyPetBox Company uses a software application to calculate staff wages and produce payslips. Sample input data for this system is shown.

National insurance (NI) number	Standard format LL123456L
Full time	Y or N Full-time = minimum of 40 hours per week Part-time = maximum of 20 hours per week
Hours worked	Integer Hours worked in current week Maximum of 10 hours overtime in one week for full-time only
Pay rate	Real Hourly pay rate

- (b) Create a **written description** of a function for the wages system. The function should receive hours worked and pay rate from the main program and return total pay due. Overtime rate is 1.5 x pay rate.

Do not use pseudo-code, program code, or a flowchart.

(4)

check if hours worked is > 40  
 If no overtime → total = hours x pay rate  
 If overtime worked → total = (hours - 40) x  
 (pay rate x 1.5) + (40 x pay rate)



**Example 1.5 - Program code**  
*iGCSE Computer Science, SAM Paper 2, Q5(a)*

- 5 Data, stored as numbers, is very easily processed using computer algorithms.
- (a) Open the file **Q05a** in the code editor.

Complete the trace table to show the execution of the code.

You may not need to fill in all the rows in the table.

(5)

target	rs	rm	r

```
target = 4
r = 1
rs = 0
rm = 0

while (r <= target):
    rs = r ** 2
    print(rs)
    rm = r % 4
    print(rm)
    r = r + 1
```

### Example 1.6 – A trace table for an algorithm expressed in pseudocode

GCSE Computer Science (2016), SAM Paper 2, Q4(b)(ii)

- (b) In the summer, Sparky Autos is open more hours each day. This means additional members of staff are needed.

Pseudo-code that works out the number of days that additional members of staff are needed is shown.

```

2  # This program is owned by Sparky Autos
3
4  SET dates[] TO ["01/06/2014", "02/06/2014", "03/06/2014"]
5  SET extra[] TO ['Y', 'N', 'Y']
6  SET visitors[] TO [122, 51, 147]
7  SET staff[] TO [12, 6, 10]
8
9  SET length TO LENGTH (extra)
10 SET count TO 0
11 SET index TO 0
12
13 REPEAT
14     IF extra[index] = 'Y' THEN
15         SET count TO count + 1
16     END IF
17     SET index TO index + 1
18 UNTIL index = length
19

```

- (ii) Complete the trace table showing the execution of the pseudo-code. You may not need to fill in all the rows in the table.

(4)

Length	Count	Index	Extra (index)

### Example 1.7 – Bubble sort

*iGCSE Computer Science, SAM Paper 2, Q4(b)*

- (b) Here is a list of numbers that need to be sorted in **ascending** order.

28	7	26	21	34	18	16	9
----	---	----	----	----	----	----	---

Perform the first pass of a bubble sort.

Use this space for working to help you answer the questions.

- (i) Complete the table to show how the list will have changed at the end of the first pass.

(1)

--	--	--	--	--	--	--	--

- (ii) State the number of comparisons made in the first pass.

(1)

- 
- (iii) State the number of swaps made in the first pass.

(1)

---

## Sample questions for iGCSE Computer Science

(c) A bubble sort is only one type of sorting algorithm.

(i) Give **one** reason why a bubble sort is inefficient when sorting a large dataset.

(1)

.....

.....

(ii) State the position in a list that will always remain unchanged after the first pass of any ascending order bubble sort.

(1)

.....

**Example 1.8 – Merge sort**  
*GCSE Computer Science (2016), SAM Paper 1, Q6(e)*

(e) A list is made up of the numbers 84, 52, 4, 6, 68, 39, 53, 1.

Show the steps involved when sorting this list of numbers using a merge sort algorithm.

(2)

Question Number	Answer	Additional Guidance																																						
6(e)	<div><div><div><div>Award 1 mark for each row of the process.</div></div></div><div><table><tr><td>84</td><td>52</td><td>4</td><td>6</td><td>68</td><td>39</td><td>53</td><td>1</td></tr></table></div><div><table><tr><td>52</td><td>84</td><td></td><td>4</td><td>6</td><td></td><td>39</td><td>68</td><td></td><td>1</td><td>53</td><td>(1)</td></tr></table></div><div><table><tr><td>4</td><td>6</td><td>52</td><td>84</td><td></td><td>1</td><td>39</td><td>53</td><td>68</td><td>(1)</td></tr></table></div><div>This leads to:</div><div><table><tr><td>1</td><td>4</td><td>6</td><td>39</td><td>52</td><td>53</td><td>68</td><td>84</td></tr></table></div></div>	84	52	4	6	68	39	53	1	52	84		4	6		39	68		1	53	(1)	4	6	52	84		1	39	53	68	(1)	1	4	6	39	52	53	68	84	<div><div><div>Award 1 mark for each step.</div><div>Marks should not be awarded for the last row.</div><div>Any notation showing distinct lists at each stage is acceptable.</div></div></div>
84	52	4	6	68	39	53	1																																	
52	84		4	6		39	68		1	53	(1)																													
4	6	52	84		1	39	53	68	(1)																															
1	4	6	39	52	53	68	84																																	

### Example 1.9 – Binary search

GCSE Computer Science (2013), Paper 1, June 2016, Q4(c)(ii)

Manuel is writing a binary search routine to search for an individual pupil in a list of all pupil numbers.

Here is the list of pupil numbers.

837, 1529, 1683, 2245, 3901, 3921, 4524

- (ii) Complete the table showing the pupil numbers visited and the associated sublists when using a binary search to locate the pupil number 1683.

(5)

Pupil number visited	Sublist

Pupil number visited	Sublist
2245	2245 > 1683

### Examples 1.10 – Efficiency of algorithms

GCSE Computer Science (2016), Specimen Paper 2, Q5(a)

- 5 The accounts department at the HappyPetBox Company processes sales, subscriptions, and invoices.

The accounts office has a list of paid invoices, but they are all jumbled up. A bubble sort can be used to sort the list of paid invoice numbers.

The pseudo-code for the sort algorithm is shown.

```
1
2 # Bubble sort paid invoices
3
4 PROCEDURE sort()
5 BEGIN PROCEDURE
6     SET size TO LENGTH (invoices)
7     FOR i FROM 0 TO size - 1 DO
8
9         FOR j FROM 0 TO size - (i + 1) DO
10             IF (invoices[j] > invoices[j + 1]) THEN
11
12                 SET temp TO invoices[j]
13                 SET invoices[j] TO invoices[j + 1]
14                 SET invoices[j + 1] TO temp
15
16             END IF
17         END FOR
18
19     END FOR
20 END PROCEDURE
21
22
```

- (a) Explain an improvement that could be made to the algorithm to increase the efficiency of the sort.

(3)

.....

.....

.....

### Example 1.11 – Decomposition

*iGCSE Computer Science, SAM Paper 2, Q6*

- 6 Open the file named **Q06** in the code editor.

In file **Q06**, the names and years of birth of artists are stored in a 2-dimensional data structure.

Labels for their work need to be created by joining the first letter of their last name, the first letter of their first name and their year of birth.

For example, a label for ('Andy', 'Warhol', 1928) would be 'WA1928'.

Write a program to:

- process each artist to create a label
- store all the labels in the data structure named 'theLabels'
- display the labels for all the artists
- find and display the name and year of birth of the youngest artist.

**Your program should function correctly, even if 'theArtists' data structure has more, fewer, or different artists.**

You **must** use the data structures in file **Q06**.

Save your amended code as **Q06FINISHED** with the correct file extension for the programming language.

**(Total for Question 6 = 20 marks)**

```
theArtists = [
    ["Andy", "Warhol", 1928],
    ["Pablo", "Picasso", 1881],
    ["Salvador", "Dali", 1904],
    ["Lavinia", "Fontana", 1552],
    ["Jackson", "Pollock", 1912],
    ["Henri", "Matisse", 1869],
    ["Frida", "Kahlo", 1907],
    ["Georgia", "O'Keeffe", 1887],
    ["Kara", "Walker", 1969],
    ["Yayoi", "Kusama", 1929]
]

theLabels = []    # Put the new user labels into this structure

# Add your code here
```



### Example 2.1 – Code readability

*GCSE Computer Science (2016), SAM Paper 2, Q4(a)*

- 4 A computer programmer uses a programming language to write program code for Sparky Autos.

(a) State **two** techniques that the programmer could use to make the code easy to read.

(2)

1 .....

2 .....

### Example 2.2 – Code readability

*iGCSE Computer Science, SAM Paper 2, Q1(b)*

- (b) Open the file **Q01b** in the code editor.

Answer these questions about the code.

- (i) Identify **one** technique that could be used to make the code in **Q01b** more readable.

(1)

- (ii) State **one** reason why code should be readable.

(1)

```
1. x = 44
2. t = 0
3. for i in range (0, 60):
4.     t = t + i - x + 12
5.     if (i >= 10):
6.         print (i * 10 / 14)
7.     else:
8.         print (i)
9.     print (t)
```

### Example 2.3 – Types of error

*iGCSE Computer Science, SAM Paper 2, Q1(c)*

- (c) (i) Give a definition of a syntax error.

(1)

- (ii) Open the file **Q01c** in the code editor.

Amend the code to correct three program errors.

Save your amended code as **Q01cFINISHED** with the correct file extension for the programming language.

(3)

```
1. myNumbers = [10, 20, 30, 40 ,50, 60, 70, 80, 90, 100]
2. total
3. for theNumber in myNumbers:
4.     total = total + theNumber
5.     if (theNumber % 2 = 0):
6.         print("Even")
7.     else:
8.         print("Odd")
9. print(total)
```

## Example 2.4 – Test data

GCSE Computer Science (2016), Specimen Paper 2, Q2(b)

- 2 The HappyPetBox Company offers discounts based on the type of box ordered and the length of the subscription selected.

```

1
2 IF ((subscriptionLength = 12) AND (boxType <> "Standard")) THEN
3     SEND "gold star discount" TO DISPLAY
4 ELSE
5     IF (subscriptionLen > 5) AND ((boxType = "Large") OR (boxType = "Medium")) THEN
6         SEND "silver star discount" TO DISPLAY
7     ELSE
8         IF ((subscriptionLen > 2) AND (boxType <> "Standard")) THEN
9             SEND "bronze star discount" TO DISPLAY
10        ELSE
11            SEND "regular pricing" TO DISPLAY
12        END IF
13    END IF
14 END IF
15
16

```

- (b) The algorithm needs to be tested thoroughly.

Complete the table to show boundary test data to meet the requirements.

(3)

Requirements	Input	
	subscriptionLen	boxType
A condition generating a bronze star discount		
Smallest subscription qualifying for a gold star discount		
Largest subscription qualifying for no discount		

### Example 2.5

*iGCSE Computer Science, SAM Paper 2, Q3(b)(vii)*

(vii) One line in the code is identified as **not** working as expected.

State why this line does **not** work as expected.

(1)

```
1. import time
2.
3. theDate = ""
4.
5. def toCelsius (inTemp):
6.     celsius = 0
7.     celsius = (5.0 / 9.0) * (inTemp - 32.0)
8.
9. def toFahrenheit (inTemp):
10.    fahrenheit = ((9.0 / 5.0) * inTemp) + 32.0
11.    return (fahrenheit)
12.
13. def waitTenSeconds ():
14.    time.sleep (10)
15.
16. def waitTime (inSeconds):
17.    time.sleep (inSeconds)
18.
19. print (toFahrenheit (0))
20. print (toCelsius (212.0)) # This line does not work properly
21. print ("Sleeping for 10")
22. waitTenSeconds()
23. print (toCelsius (32.0))
24. print ("Sleeping for 5")
25. waitTime (5)
26. print (toFahrenheit (100.0))
```

**Example 2.6 – Trace table**  
*iGCSE Computer Science, SAM Paper 2, Q5(a)*

- 5 Data, stored as numbers, is very easily processed using computer algorithms.
- (a) Open the file **Q05a** in the code editor.

Complete the trace table to show the execution of the code.

You may not need to fill in all the rows in the table.

(5)

target	rs	rm	r

```
target = 4
r = 1
rs = 0
rm = 0

while (r <= target):
    rs = r ** 2
    print(rs)
    rm = r % 4
    print(rm)
    r = r + 1
```

## Example 2.7 – Structural components

*iGCSE Computer Science, SAM Paper 2, Q1(a)*

1 Programmers use accepted programming constructs when writing code.

(a) Open the file **Q01a** in the code editor.

Answer these questions about the code.

(i) Identify the name given to a **data structure** in the code.

(1)

(ii) Identify the line number(s) showing **repetition**.

(1)

(iii) Identify the line number(s) showing **selection**.

(1)

(iv) Identify the name of a **variable**.

(1)

```
1. myNumbers = [20, 30, 40, 50]
2. total = 0
3. for i in range (5, 15):
4.     total = total + i
5.     if (i < 10):
6.         print (i * 10)
7.     else:
8.         print (i)
9. print (total)
```

## Example 2.8 – Data types and structures

iGCSE Computer Science, SAM Paper 2, Q6

6 Open the file named **Q06** in the code editor.

In file **Q06**, the names and years of birth of artists are stored in a 2-dimensional data structure.

Labels for their work need to be created by joining the first letter of their last name, the first letter of their first name and their year of birth.

For example, a label for ('Andy', 'Warhol', 1928) would be 'WA1928'.

Write a program to:

- process each artist to create a label
- store all the labels in the data structure named 'theLabels'
- display the labels for all the artists
- find and display the name and year of birth of the youngest artist.

**Your program should function correctly, even if 'theArtists' data structure has more, fewer, or different artists.**

You **must** use the data structures in file **Q06**.

Save your amended code as **Q06FINISHED** with the correct file extension for the programming language.

**(Total for Question 6 = 20 marks)**

```
theArtists = [
    ["Andy", "Warhol", 1928],
    ["Pablo", "Picasso", 1881],
    ["Salvador", "Dali", 1904],
    ["Lavinia", "Fontana", 1552],
    ["Jackson", "Pollock", 1912],
    ["Henri", "Matisse", 1869],
    ["Frida", "Kahlo", 1907],
    ["Georgia", "O'Keeffe", 1887],
    ["Kara", "Walker", 1969],
    ["Yayoi", "Kusama", 1929]
]

theLabels = []    # Put the new user labels into this structure

# Add your code here
```

### Example 2.9 – Data types and structures

GCSE Computer Science (2016), SAM Paper 2, Q5(c)

- (c) Complete the table to give the appropriate data type of a variable to store each item.

(4)

Item	Data type
Gender of individual staff member	
Whether an individual car is still under the manufacturer's warranty	
Mean number of hours needed to recharge the battery in each car	
The number on the individual car	

- (d) Each member of staff:

- has a 4-digit ID number, such as 3865 or 4722
- works a whole number of hours on the days they work.
- works no more than 12 hours in a single day.

The business is open 7 days a week.

Draw a diagram of a data structure that shows the hours worked for each day of the week. Include data for at least **two** members of staff.

(3)



## Example 2.10 - Validation

GCSE Computer Science(2016), Specimen Paper 2, Q6(a)

- 6 The HappyPetBox Company uses a software application to calculate staff wages and produce payslips. Sample input data for this system is shown.

National insurance (NI) number	Standard format LL123456L
Full time	Y or N Full-time = minimum of 40 hours per week Part-time = maximum of 20 hours per week
Hours worked	Integer Hours worked in current week Maximum of 10 hours overtime in one week for full-time only
Pay rate	Real Hourly pay rate

- (a) The input data needs to be validated.
- (i) Complete the validation rules in the pseudo-code.

Write your answers in the boxes provided.

(6)

```

3  # Validate NI number
4  STRING NInumber
5  INTEGER hoursWorked
6  CHARACTER fullTime
7
8  # Validate National Insurance number
9  IF LENGTH (NInumber)  THEN
10
11      SEND "Error message" TO DISPLAY
12  END IF
13
14  # Validate hours worked this week
15  IF hoursWorked < 0  THEN
16
17      SEND "Error message" TO DISPLAY
18  END IF
19
20  # Validate part-time worker's hours
21  IF fullTime = 'N'  THEN
22
23      SEND "Error message" TO DISPLAY
24  END IF

```

- (ii) Create a suitable 'error message' to explain one of your validation rules to the user.

(1)

### Example 2.11 – Reading from and writing to files

*iGCSE Computer Science, SAM Paper 2, Q3(c)*

Open the code named **Q03c** in the code editor.

Write a program to implement these requirements.

For all lines in the **Cities.txt** file, the code must:

- read the line
- append a line number and a space to the front
- write the new line to a **Numbered.txt** file
- print the line to the display

You must use the structure given in file **Q03c** to complete the program.  
Do not add further functionality.

Save your amended code as **Q03cFINISHED** with the correct file extension for the programming language.

(7)

```
# Do not use any other data structure such as an array or a list.
count = 0                # A counter for the line numbers
theLine = ""             # Holds each line of the file

# Open "Cities.txt" as input

# Open "Numbered.txt" as output

# Use a for/each loop to read each line of
# the input file into a variable named 'theLine'

    # Increment the line count

    # Add the line number to the front of the line followed by a space

    # print the line to the display

    # Write the new line to the output file

# Close the input file

# Close the output file
```

### Example 2.12 – Logical operators

*iGCSE Computer Science, SAM Paper 2, Q2(c)*

```
myList = [(800, 23000), (1499, 10000), (1600, 47000), (200, 10000)]
for pair in myList:
    income = pair[1]
    attendance = pair[0]
    print ("Attendance: ", attendance, " income: ", income)

    if ( ):
        print ("Sufficient profit made this week")
    elif ( ):
        print ("income in line with attendance this week")
    elif ( ):
        print ("Attendance is very low this week. Contact the fan club.")
    else:
```

Condition	Output message
Attendance is at least 1500	Sufficient profit made this week
Income is at least 45000	Sufficient profit made this week
Attendance is at least 750; income is at least 22500	Income in line with attendance this week
Attendance is fewer than 500	Attendance is very low this week Contact fan club
All other inputs	Possible accounting error

(4)

### Example 2.13 – Subprograms

*iGCSE Computer Science, SAM Paper 2, Q3(a) and Q3(b)*

- 3** A holiday company has a website. They would like to publish the daily temperatures in their most popular destinations.

(a) Describe **one** benefit of using subprograms.

(2)

- (b) The holiday company needs to be able to convert temperatures between Celsius and Fahrenheit.

Open the file **Q03b** in the code editor.

Answer these questions about the code.

- (i) Identify the name of a **built-in** subprogram in the code.

(1)

- (ii) Identify the name of a **user-defined** subprogram.

(1)

- (iii) Identify the name of **one** input parameter used in a subprogram.

(1)

- (iv) Identify the name of a subprogram that does **not** use input parameters.

(1)

- (v) Identify the name of a **local** variable.

(1)

- (vi) Identify the name of a **global** variable.

(1)

- (vii) One line in the code is identified as **not** working as expected.

State why this line does **not** work as expected.

(1)

### Example 2.14 – Library modules

GCSE Computer Science, June 2016 Q3(c)(i)

#### Candidate 1

(c) Shaneela is planning to use libraries in the code for the website.

(i) State **two** reasons for using libraries in code.

(2)

1 Don't have to repeat pieces of frequently used long code.

2 The code is easier to understand.

#### Candidate 2

1 Using libraries removes the need to program

simple things that could be included in a ~~kin~~ library, saving time.

2 Libraries are usually without any errors and are compiled ~~already~~ <sup>programs that are</sup> ~~and executed~~ executed by the interpreter, so it will be more ~~efficient~~ efficient to use libraries than ~~writing them~~ <sup>writing them</sup> ~~yourself~~ <sup>yourself</sup>.

#### Candidate 3

(2)

1 Lots of information/<sup>data</sup> can be obtained from libraries.

2 The use of libraries in code speeds up the ~~pre~~-running speed of the program.

### Example 2.15 – Generalising a subprogram

GCSE Computer Science, June 2016 Q4(c)(i)

- (c) Manuel is writing a program to help pupils with their maths revision.

Here is some pseudocode for an algorithm that calculates the area of a circle, with radius of 5 units. It uses a subprogram.

```
calcCircleArea ()  
  
PROCEDURE calcCircleArea ()  
BEGIN PROCEDURE  
    SET area TO Pi * 5 * 5  
    SEND area TO DISPLAY  
END PROCEDURE
```

Manuel wants to change the pseudocode so that the area of any circle can be calculated using calcCircleArea.

- (i) Write the changed pseudocode in the box below.

(3)

#### Candidate 1

SET radius TO VALUE

~~PRO~~  
calcCircleArea()

PROCEDURE calcCircleArea (radius)

BEGIN PROCEDURE

SET area TO Pi \* radius \* radius

SEND area TO DISPLAY

END PROCEDURE

**Candidate 2**

```
calCircleArea ()  
  
PROCEDURE calCircleArea ()  
Be BEGIN PROCEDURE  
    radius = int.(input("what is the radius"))  
    SET area TO  $\pi * \text{radius} * \text{radius}$   
    SEND area TO DISPLAY  
END PROCEDURE
```

**Candidate 3**

```
calCircleArea ()  
    usernumber = input("enter a number")  
PROCEDURE calCircleArea ()  
    BEGIN PROCEDURE  
        SET area to  $\pi * \text{user number} * \text{user number}$   
        SEND area TO DISPLAY  
    END PROCEDURE.
```

---

### Example 3.1 – binary numbers

*GCSE Computer Science (2013), June 2015, Q1(d), Q1(e)*

- (d) The robot is 12 metres from an obstacle. 12 is positive denary.

Fill in the table to show the number 12 (denary) in 8-bit binary, using a sign and magnitude representation.

(1)

--	--	--	--	--	--	--	--

- (e) The robot travels 14 metres. –14 is negative denary.

Fill in the table to show the number –14 (denary) in 8-bit binary, using a two's complement representation.

(1)

--	--	--	--	--	--	--	--

### Example 3.2 – binary arithmetic

*GCSE Computer Science (2013), June 2015, Q1(f)*

- (f) (i) Add these two 8-bit binary integers and write the result in the last row of this table.

(1)

0	1	0	1	0	1	1	0
0	0	1	0	1	0	1	1

Another 8-bit addition generates an overflow error.

- (ii) State what is meant by the term overflow error.

(1)

.....

.....

- (iii) State what happens if the overflow error is ignored and the result is used in other calculations.

(1)

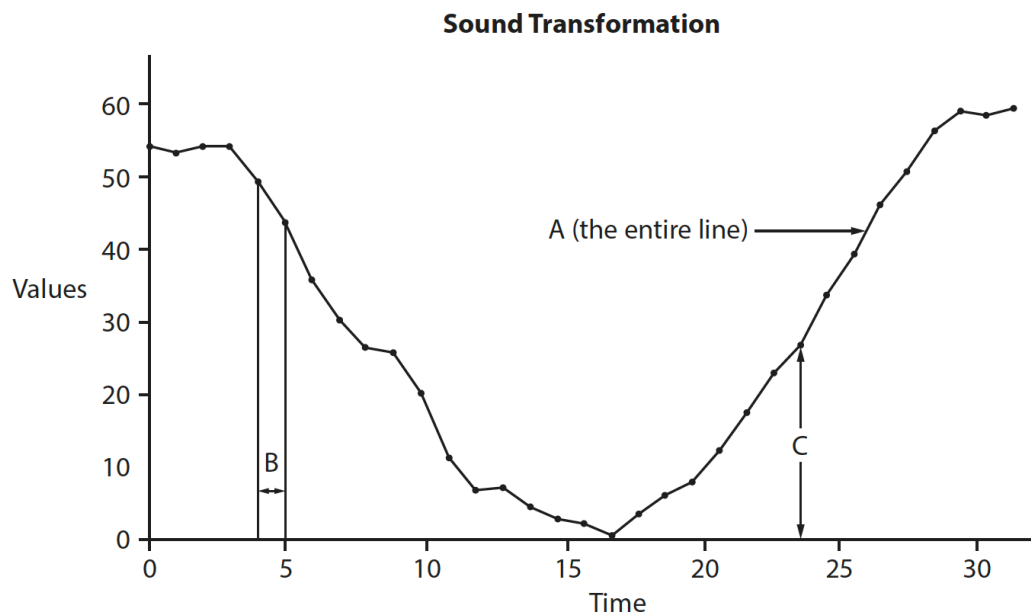
.....



### Example 3.3 - data representation

GCSE Computer Science (2016), SAM Paper 1, Q7(b)

- (b) Sound can be stored on a digital device, but only after being converted from its naturally occurring state.



- (i) Identify each of the items (A, B, C) shown on the image in relation to the conversion required to store sound on a digital device.

(3)

A .....

B .....

C .....

### Example 3.4 – compression

*GCSE Computer Science (2016), SAM Paper 1, Q2*

2 Different types of compression are used for different purposes.

- (a) A travel company has designed some brochures that contain images and text in desktop publishing format. The travel company sends the documents electronically to a printing company for them to be printed.

Explain why the travel company uses lossless compression to send the documents.

(2)

.....

.....

.....

.....

- (b) Compression normally reduces file size.

State **two** other characteristics of lossy compression.

(2)

1 .....

.....

2 .....

.....

- (c) Run length encoding (RLE) is a type of image compression. Some data for an image is shown.

b	b	b	r	g	g	g	g	r	r
---	---	---	---	---	---	---	---	---	---

Show the result of compressing this data for the image using RLE.

(2)

.....

**Example 3.5 – Calculating file sizes**

*GCSE Computer Science (2016), Specimen Paper 1, Q3(a)*

- 3** A landscape photographer wishes to publish a series of preview images online. He is concerned about file sizes.

(a) The 24-bit RGB images are to be displayed on screen at a resolution of 400 x 250 pixels.

Construct an expression to calculate the size of one of the image files (KB). You do not need to carry out the calculation.

**(3)**

---

---

---

---

---

---

### Example 3.6 – encryption

GCSE Computer Science (2013), Paper 1, June 2016, Q1(b)

(b) Data transmitted over a network is sometimes encrypted.

(i) State **one** reason why data encryption is used on a network.

(1)

#### Candidate 1

So nobody can hack the data stored in the network

#### Candidate 2

If it were to be intercepted one cipher could not be read without the key.

(ii) Complete the table using a Caesar cipher to encrypt and decrypt the text.

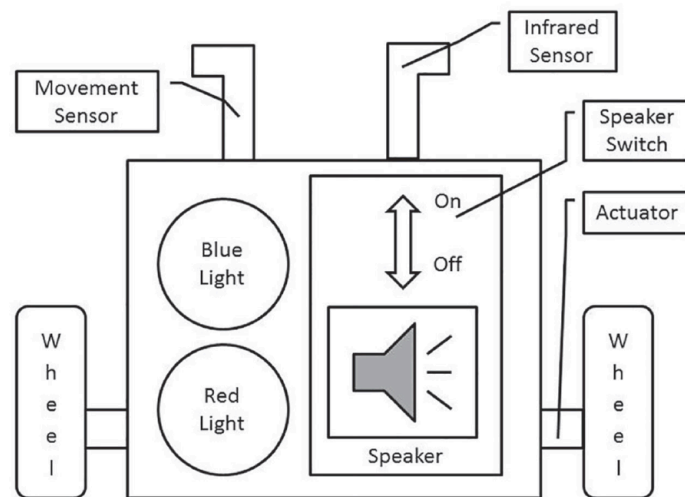
(2)

Plain text	Shift	Cipher text
digit	+3	
	-2	zglypw

### Example 4.1 – input-process-output

GCSE Computer Science (2013), Paper 1, June 2015, Q1(a)

- 1 This is a block diagram of a robot.



- (a) Put a cross to identify whether each of these is an input, process, output, or none.

(3)

	Input	Process	Output	None
Actuator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Infrared Sensor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wheel	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Calculate Distance	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Programming Language	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Movement Sensor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### Example 4.2 – sequential and parallel processing

GCSE Computer Science (2013), Paper 1, June 2016, Q4(b)(ii)

- (ii) Explain the difference between the sequential and parallel computational models.

(4)

A sequential computational model runs one instruction after another doing one operation per <sup>single</sup> clock cycle. A parallel computational model can execute multiple instructions in one full clock cycle. The parallel is used in dual/quadruple core processors while sequential is run on a processor with a single core.

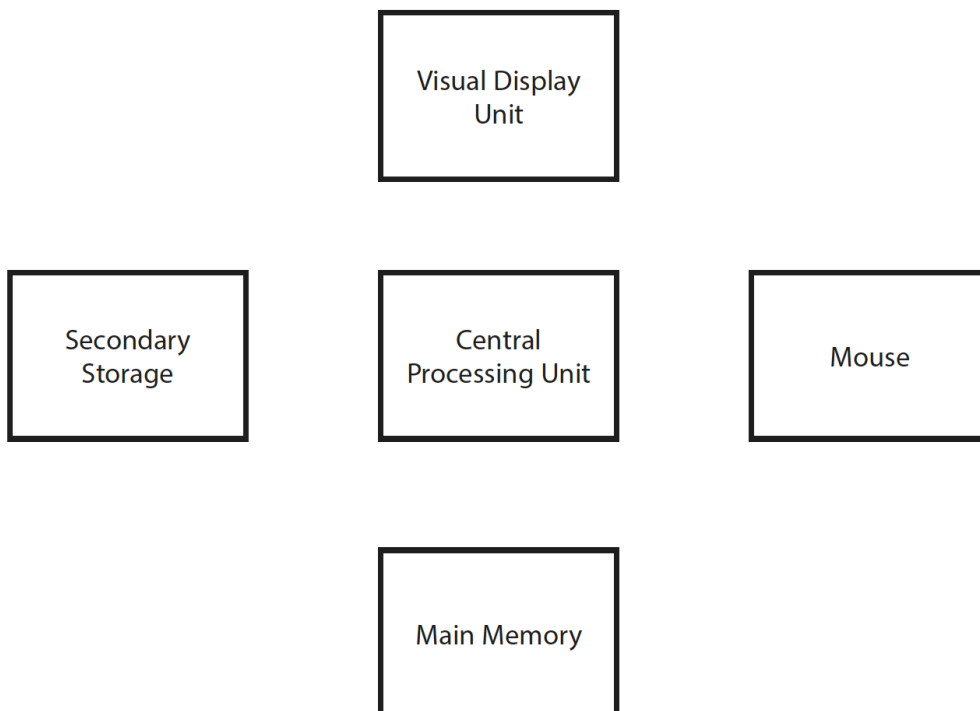


### Example 4.3 – input-output

GCSE Computer Science (2013), Paper 1, June 2015, Q2(c)

- (c) Complete the diagram by adding **six** arrows to indicate the input and output relationships between the devices.

(3)



### Example 4.4 – secondary storage

*GCSE Computer Science (2013), Paper 1, June 2015, Q3(b)(ii)*

- (b) Files for the website are stored on secondary storage media.

Complete this table with the name of the category of secondary storage described in each row.

(3)

Category of secondary storage	Description
	Uses metal platters coated in iron oxide. The platters rotate at high speeds.
	Small pits are burned in patterns onto a flat surface. A laser can be used to interpret light reflected from the flat or pitted surface.
	No moving parts; data is stored as an electrical charge.

### Example 4.5 – truth tables

GCSE Computer Science (2013), Paper 1, June 2015, Q1(h)

(h) Refer to the robot diagram.

Assume **R** is the red light, **B** is the blue light, and **S** is the speaker switch.

(i) Complete the table to show the Boolean expression **Q=(NOT R) AND B**.

(3)

R	B	NOT R	Q
0	0		
0	1		
1	0		
1	1		

(ii) Construct a Boolean expression to determine **if either light is on** at the same time as the **speaker switch is on**.

(1)

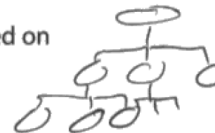
### Example 4.6 – operating systems

GCSE Computer Science (2013), Paper 1, June 2016, Q3(a)

3 Shaneela is designing and coding a website. The files she creates will be stored on a computer.

(a) File management is a function that an operating system performs.

Describe how an operating system organises files.



(3)

It organises the files using a hierarchical structure. It simply starts searching for a file by starting at the top most folder, called the ~~root~~ root node. Then it works its way down the ~~second~~ second row ~~which~~, which is called a subdirectional, and then the last row to find a file.



### Example 4.7 – code translators

GCSE Computer Science (2013), Paper 1, June 2015, Q5(a)

5 A student has decided to learn how to program.

- \*(a) Discuss the suitability of **compiled** and **interpreted** programming languages for the student.

(6)

Compiled languages such as Java are useful as they work on any machine. However, they are harder to identify errors in as all the code is translated into machine code before it is run. This is a contrast to interpreted high level languages like Python where the code is translated and run a line at a time - this allows for easier debugging for a beginner who is likely to make a lot of mistakes. However, a downside to interpreted languages are they are slower and require a special environment from which to run unlike a compiled language. Overall,

**Example 5.1 – network protocols**

*GCSE Computer Science (2013), Paper 1, June 2016, Q1(a)(i)*

1 (a) Computer networks are valuable to many businesses and individuals.

(i) State the purpose of network protocols.

(1)

**Candidate 1**

It protects your data and files.

**Candidate 2**

They act as rules so that the devices on the network can communicate.

### Example 6.1 – Environmental impact

GCSE Computer Science (2013), Paper 1, June 2015, Q4(a)

- (a) AJ's chief executive is concerned about the environmental impact of using computers.

Give **three** possible **environmental** impacts of using computing devices.

Suggest **one** possible action AJ's could take to reduce the environmental impact.

(4)

Impact 1

Impact 2

Impact 3

Action

Impact 1
The CO <sub>2</sub> emissions that are produced when supplying the computer with electricity
Impact 2
When the computers are broken they usually are thrown away into land fill
Impact 3
The manufacturing of the devices will cause damage to the environment.
Action
Use the posted <del>source</del> <sup>source</sup>

## Example 6.2 – AI

GCSE Computer Science (2013), Paper 1, June 2016, Q5(a)

5 Artificial intelligence is an emerging trend in computer science.

\*(a) Discuss the use of artificial intelligence by describing some of its characteristics, the ways in which it may be used and the ethical issues associated with its use.

(6)

### Candidate 1

Artificial <sup>intelligence</sup> ~~intelligence~~ is the study of computers <sup>(6)</sup> ~~replicating~~ <sup>minimising</sup> human thoughts or actions ~~(aware)~~ (Self aware). Artificial intelligence is normally found in games the AI controllers in games <sup>talk bots & can even</sup> ~~be found on~~ <sup>or even Siri or Cortana</sup> ~~on operating systems~~ operating systems helping the user via voice talking & having a conversation. Examples include Siri or Cortana. Artificial intelligence has progressed <sup>within the last decade</sup> ~~alot~~ <sup>an</sup> with ~~computer~~ <sup>AI</sup> that was able to beat <sup>world</sup> ~~chess~~ <sup>chess</sup> champions at chess. This technology could ~~be~~ potentially be dangerous ~~in the~~ hands of the military if computers are allowed to kill humans or fight wars instead of humans. Or if the Artificial intelligence ~~became~~ became very self aware & turned against humanity. There are <sup>also</sup> ethical issues ~~connected~~ with AI <sup>development & application</sup> such as in the <sup>military</sup> ~~army~~ allowing <sup>AI</sup> ~~programs~~ to kill humans <sup>(Now drones have human operators for ethical reasons)</sup> ~~instead of humans~~ and creating ~~new~~ 'life'. Despite these points, there are <sup>for many</sup> ~~many~~ advantages to Artificial Intelligence such as helping progress technology further, complex problem solving & if implemented into robots, the robots could go into dangerous zones such as ~~an~~ a fallout zone or post ~~earth~~ earthquake to help survivors / discover new places. Also

if use robots are used in wars instead of humans, no humans have to die. There is also the <sup>ethical issue</sup> ~~issue~~ raised <sup>that if the</sup> if AI is 'aware' enough, would it be considered <sup>a slave</sup> ~~slavery~~ to humans and should they have equal rights.

## Candidate 2

Artificial intelligence is the manufacturing of machines able to replicate human behaviour. Artificial intelligence can be as simple as a robot walking of its own accord, or a machine which can have an actual conversation with a human. It is mainly used in computer games, with artificial intelligence (AI) being able to complete objectives in a game or target the player character. AI can also be developed to be able to work better than factory machines, complete jobs such as pizza delivery and medical diagnosis, and to even operate phone calls. Issues arise due to AI being a machine, so there is always a chance of malfunction or error, and this could cause jobs to be done wrong. People may also say it's wrong to create artificial humans and that replacing humans with machines can cause unemployment. People also worry if machines could become smarter than us.