

Please write clearly in block capitals.

Centre number

--	--	--	--	--

Candidate number

--	--	--	--

Surname

---

Forename(s)

---

Candidate signature

---

# A-level COMPUTER SCIENCE

## Paper 2

Thursday 22 June 2017

Morning

Time allowed: 2 hours 30 minutes

### Materials

For this paper you must have:

- a calculator.




### Instructions

- Use black ink or black ball-point pen.
- Fill in the boxes at the top of this page.
- Answer **all** questions.
- You must answer the questions in the spaces provided. Do not write outside the box around each page or on blank pages.
- Do all rough work in this book. Cross through any work you do not want to be marked.

### Information

- The marks for questions are shown in brackets.
- The maximum mark for this paper is 100.

### Advice

- In some questions you are required to indicate your answer by completely shading a lozenge alongside the appropriate answer as shown. 
- If you want to change your answer you must cross out your original answer as shown. 
- If you wish to return to an answer previously crossed out, ring the answer you now wish to select as shown. 

For Examiner's Use

Question	Mark
01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
<b>TOTAL</b>	





---

---

---

---

---

---

---

0 1 . 2

During the **decode and execute stages** of the fetch-execute cycle the instruction that is being processed is stored in the CIR. Explain why the instruction could **not** be processed directly from the MBR.

[2 marks]

---

---

---

---

---

---

---

The computer system shown in **Figure 1** uses the von Neumann architecture. The Harvard architecture is an alternative to this.

0 1 . 3

Explain why the Harvard architecture is sometimes used in preference to the von Neumann architecture.

[2 marks]

---

---

---

---

---

---

---

8

Turn over ►

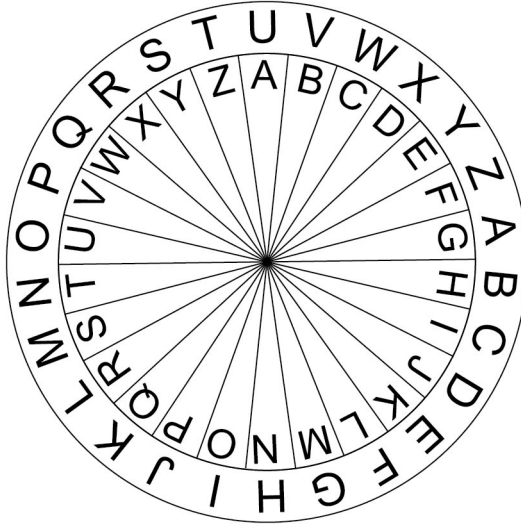


0 2

Figure 2 shows a message being encrypted using a Caesar cipher.

Figure 2

Wheel settings to use



Example

Plaintext: COMPUTER



Ciphertext: IUSVAZKX

0 2 . 1

Decrypt the ciphertext "QGOZRKT" using the Caesar cipher with the settings shown in Figure 2.

[1 mark]

---

The Vernam cipher is a more sophisticated cipher system that, under certain circumstances, offers perfect security.

0 2 . 2

State **two** conditions that must be met for the Vernam cipher to offer perfect security.

[2 marks]

Condition 1

---



---

Condition 2

---



---



Both the Caesar and Vernam ciphers are symmetric ciphers, whereas a public and private key encryption system is an asymmetric cipher system.

0 2 . 3

Explain the difference between a symmetric and an asymmetric cipher system.

[1 mark]

---

---

---

---

---

4

**Turn over for the next question**

**Turn over ►**



0 3

In a particular communications system, eight different voltage levels are used to encode the value of groups of bits. Each voltage level encodes the value of one group of bits.

0 3 . 1

Given that eight different voltage levels are used, how many bits can be in a group that is encoded by a voltage level?

[1 mark]

---



---

0 3 . 2

The baud rate for this system is 500 baud.

What is the system's bit rate?

[1 mark]

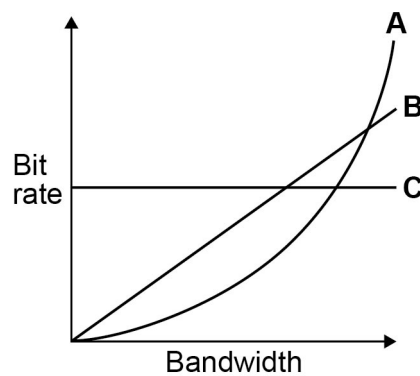
---



---

**Figure 3** shows three suggested relationships between bandwidth and bit rate.

**Figure 3**



0 3 . 3

Shade **one** lozenge to indicate which of the lines, **A**, **B** or **C** in **Figure 3**, shows the correct relationship between bandwidth and bit rate.

[1 mark]

**A**  **B**  **C**



The system sends the data over a long distance using serial communication.

0 3 . 4

Explain why serial communication is more appropriate in this instance than parallel communication.

[2 marks]

---

---

---

---

---

---

---

5

Turn over for the next question

Turn over ►



0 4

A computer process, X, can only start executing once processes A and B have finished executing and either communication channel C or communication channel D or both are available to use.

The states of processes and communication channels can be read using the following Boolean variables:

- **A** is set to TRUE if process A has completed and FALSE if process A is still running.
- **B** is set to TRUE if process B has completed and FALSE if process B is still running.
- **C** is set to TRUE if communication channel C is available and FALSE if it is not available.
- **D** is set to TRUE if communication channel D is available and FALSE if it is not available.

The Boolean variable **X** should be set to TRUE if the values of the variables **A**, **B**, **C** and **D** indicate that process X can start and to FALSE if they indicate that process X cannot start yet.

0 4 . 1

In the space below, draw a logic circuit that will represent the logic of the system described above for the inputs **A**, **B**, **C** and **D** and the output **X**.

[3 marks]



0 4 . 2

Write a Boolean expression to represent the logic used to start process X.

[2 marks]

X = \_\_\_\_\_







D-type flip-flops can be included in logic circuits.

0 4 . 4

Explain the general purpose of a D-type flip-flop.

[1 mark]

---

---

0 4 . 5

One input to a D-type flip-flop is a data signal.

State what the other input to a D-type flip-flop is **and** what it is used for.

[2 marks]

---

---

---

---

---

---

---

12



0 5

**Figure 4** shows the structure of an example machine code instruction, taken from the instruction set of a particular processor.

**Figure 4**

Opcode						Operand(s)								
Basic Machine Operation						Addressing Mode								
0	1	1	1	1	0	0	1	0	1	1	1	0	0	1

0 5 . 1

State the purpose of the operand part of an instruction **and** explain how the addressing mode is related to this.

**[2 marks]**

---



---



---



---



---



---

**Question 05 continues on the next page**

**Turn over ►**



**Figure 5** shows an assembly language program together with the contents of a section of the main memory of the computer that the program will be executed on. Each main memory location and register can store a 16-bit value.

The assembly language instruction set that has been used to write the program is listed in **Table 1** on the next page.

**Figure 5**

Program
LDR R1, 100
LSL R2, R1, #2
ADD R1, R1, R2
LDR R3, 101
CMP R3, R1
BEQ labela
MOV R4, #0
B labelb
labela:
MOV R4, #1
labelb:
STR R4, 102
HALT

Memory Address (in decimal)	Main Memory Contents (in decimal)
100	10
101	50
102	80

0 5 . 2

Complete the trace table below, **in decimal**, to show how the values stored in the registers and main memory change as the program in **Figure 5** is executed. You may not need to use all of the rows.

[4 marks]

Register Contents				Main Memory Location Contents		
R1	R2	R3	R4	100	101	102

0 5 . 3

Explain what the assembly language program in **Figure 5** does.

[1 mark]

---



---



---



**Table 1 – standard AQA assembly language instruction set**

LDR Rd, <memory ref>	Load the value stored in the memory location specified by <memory ref> into register d.
STR Rd, <memory ref>	Store the value that is in register d into the memory location specified by <memory ref>.
ADD Rd, Rn, <operand2>	Add the value specified in <operand2> to the value in register n and store the result in register d.
SUB Rd, Rn, <operand2>	Subtract the value specified by <operand2> from the value in register n and store the result in register d.
MOV Rd, <operand2>	Copy the value specified by <operand2> into register d.
CMP Rn, <operand2>	Compare the value stored in register n with the value specified by <operand2>.
B <label>	Always branch to the instruction at position <label> in the program.
B<condition> <label>	Branch to the instruction at position <label> if the last comparison met the criterion specified by <condition>. Possible values for <condition> and their meanings are: EQ: equal to                      NE: not equal to GT: greater than                LT: less than
AND Rd, Rn, <operand2>	Perform a bitwise logical AND operation between the value in register n and the value specified by <operand2> and store the result in register d.
ORR Rd, Rn, <operand2>	Perform a bitwise logical OR operation between the value in register n and the value specified by <operand2> and store the result in register d.
EOR Rd, Rn, <operand2>	Perform a bitwise logical XOR (exclusive or) operation between the value in register n and the value specified by <operand2> and store the result in register d.
MVN Rd, <operand2>	Perform a bitwise logical NOT operation on the value specified by <operand2> and store the result in register d.
LSL Rd, Rn, <operand2>	Logically shift left the value stored in register n by the number of bits specified by <operand2> and store the result in register d.
LSR Rd, Rn, <operand2>	Logically shift right the value stored in register n by the number of bits specified by <operand2> and store the result in register d.
HALT	Stops the execution of the program.

**Labels:** A label is placed in the code by writing an identifier followed by a colon (:). To refer to a label the identifier of the label is placed after the branch instruction.

#### Interpretation of <operand2>

<operand2> can be interpreted in two different ways, depending on whether the first character is a # or an R:

- # – use the decimal value specified after the #, eg #25 means use the decimal value 25.
- R<sub>m</sub> – use the value stored in register m, eg R6 means use the value stored in register 6.

The available general purpose registers that the programmer can use are numbered 0 to 12.

Turn over ►





**0 6**

In a functional programming language a function named `square` and three lists `a`, `b` and `c` are defined as follows.

```
square x = x * x
```

```
a = [1, 3, 5]
```

```
b = [1, 5, 10, 15]
```

```
c = [9, 7, 2]
```

**0 6****1**

What is the list or value that is the result of applying the functions `head(tail(tail b))`?

**[1 mark]**


---

**0 6****2**

Calculate the results of making the function calls listed in **Table 2** with the lists `a`, `b` and `c` above.

**[3 marks]****Table 2**

Function Call	Result
<code>map square a</code>	
<code>filter (&lt;10) b</code>	
<code>fold (+) 0 c</code>	

**0 6****3**

`map` is an example of a higher-order function.

Explain what a higher-order function is.

**[1 mark]**


---



---



---



---

**5****Turn over for the next question****Turn over ►**







0	8
---	---

Two methods of representing and playing music on a computer are sampled sound and MIDI.

Sound is being sampled using a 16-bit sample resolution and a sample rate of 20 000 Hz.

1 Hz is one sample per second.

0	8	.	1
---	---	---	---

Calculate the amount of storage space that will be required to store 30 seconds of recorded sound. Express your answer in kilobytes.

You **must** show your working.

**[2 marks]**

---

---

---

---

---

---

---

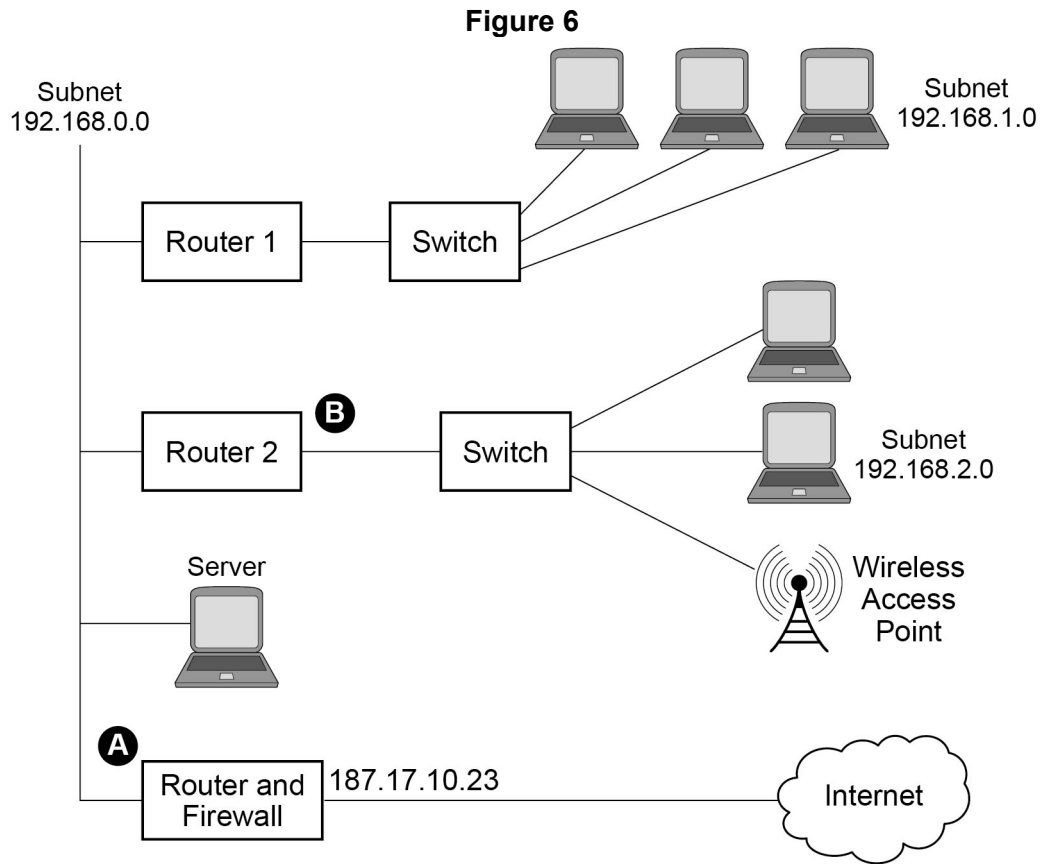
**Answer** \_\_\_\_\_





0 9

**Figure 6** shows the physical topology of a local area network (LAN) and its connection to the Internet. The LAN uses the IPv4 protocol.



0 9 . 1

State suitable IP addresses for:

[2 marks]

The 'Router and Firewall' port labelled **A** \_\_\_\_\_

The 'Router 2' port labelled **B** \_\_\_\_\_

0 9 . 2

Explain the difference between a physical topology and a logical topology.

[2 marks]

---



---



---



---



---



---





1 0

A garage services and repairs cars. It uses a relational database to keep track of the jobs that customers have booked for it to carry out. The database includes jobs that have been completed and jobs that are waiting to be done.

The details of the jobs that the garage does, together with the parts that it stocks and uses are stored in the database using the four relations shown in **Figure 7**.

**Figure 7**

Job ( <u>JobID</u> , CarRegNo, JobDate, InGarage, JobDuration) Car ( <u>CarRegNo</u> , Make, Model, OwnerName, OwnerEmail, OwnerTelNo) Part ( <u>PartID</u> , Description, Price, QuantityInStock) PartUsedForJob ( <u>JobID</u> , <u>PartID</u> , QuantityUsed)
---

- Each car has a unique CarRegNo.
- A type of car can be uniquely identified by the combination of its Make and Model. Different Makes may use the same Model name and a particular manufacturer (Make) will produce several different car Models.
- A booking made for a car on a particular date counts as one job, regardless of how many different tasks are completed upon it.
- A job might require the use of any number of parts, including zero.
- Some of the details are stored in the database as soon as a booking is made and others are only added when a job has been completed.

The attribute JobID is the Entity Identifier (Primary Key) of the Job relation.

1 0 . 1

If the JobID attribute were not included in the Job relation, which other attribute or attributes that are currently in the relation could probably be used as an Entity Identifier (Primary Key) instead?

[1 mark]

---

It has been suggested that the owner details (OwnerName, OwnerEmail, OwnerTelNo) should not be stored in the Car relation and that a new relation should be created to store owner details separately from car details.

1 0 . 2

Explain why storing the owner details separately would improve the design of the database.

[2 marks]

---



---



---



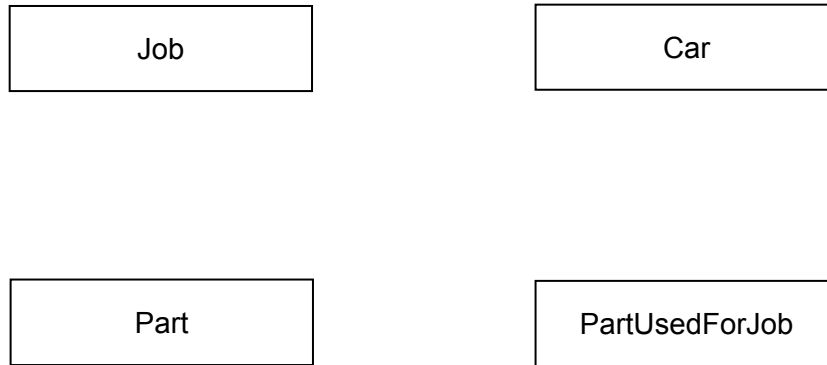
---



1 0 . 3

On the incomplete Entity-Relationship diagram below show the degree of any **three** relationships that exist between the entities.

[2 marks]



When an appointment is made for a job, this is represented in the Job relation. At the time of booking, the InGarage attribute is set to False and the JobDuration attribute is set to 0:00. When the car arrives at the garage the value of the InGarage attribute is changed to True. When the job is finished the value of the JobDuration attribute is updated to indicate how long the job took and details of the parts used are recorded in the database.

The Job with JobID 206 has been completed. The job took 1 hour 30 minutes (1:30) and used two of the parts with PartID 12.

1 0 . 4

Write the SQL commands that are required to record the amount of time that the job took in the database.

[3 marks]

---



---



---



---



---

1 0 . 5

Write the SQL commands that are required to record in the database the fact that two of the parts with PartID 12 were used.

[2 marks]

---



---



---



---

Turn over ►







There are restrictions on which parts can be fitted to which cars. For example:

- The driver's door mirror with PartID 104 can only be fitted to one particular make and model of car.
- The ignition switch with PartID 27 can be fitted to any model of car for one particular make as the maker uses the same ignition switch in all models.
- The tyre with PartID 97 can be fitted to a wide range of cars of different makes and models as it is a standard size.

If the information about which parts could be fitted to which makes and models of cars were represented in the database, it could be used to help a mechanic identify the correct parts to use for a job.

1 0 . 7

Explain how the database design could be modified to represent which makes and models of car a part can be fitted to.

**[3 marks]**

---

---

---

---

---

---

---

---

---

---

18

**Turn over for the next question**

**Turn over ►**

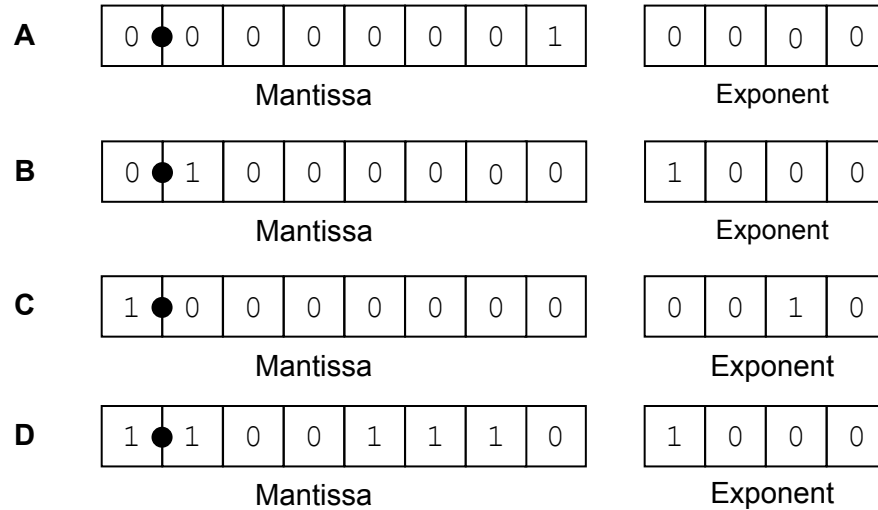


1 1

A particular computer uses a **normalised** floating point representation with an 8-bit mantissa and a 4-bit exponent, both stored using **two's complement**.

Four bit patterns that are stored in this computer's memory are listed in **Figure 8** and are labelled **A**, **B**, **C** and **D**. Some of the bit patterns are valid normalised floating point numbers.

Figure 8



1 1 . 1

Shade **one** lozenge to indicate which bit pattern (**A–D**) in **Figure 8** represents a negative normalised value.

[1 mark]

A	<input type="checkbox"/>	B	<input type="checkbox"/>	C	<input type="checkbox"/>	D	<input type="checkbox"/>
---	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------

1 1 . 2

Shade **one** lozenge to indicate which bit pattern (**A–D**) in **Figure 8** represents the smallest positive normalised value.

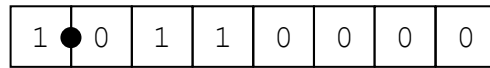
[1 mark]

A	<input type="checkbox"/>	B	<input type="checkbox"/>	C	<input type="checkbox"/>	D	<input type="checkbox"/>
---	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------

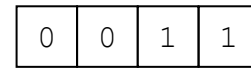


1 1 . 3

The following is a floating point representation of a number:



Mantissa



Exponent

Calculate the decimal equivalent of the number. You **must** show your working.**[2 marks]**


---



---



---



---

Answer \_\_\_\_\_

1 1 . 4

Write the normalised floating point representation of the decimal value 58.5 in the boxes below. You **must** show your working.**[3 marks]**


---



---



---



---

Answer



Mantissa



Exponent

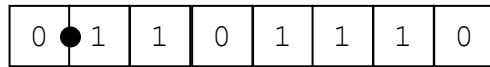
Question 11 continues on the next page

Turn over ►

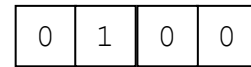


There can be a loss of precision when a decimal number is stored when using a floating point system.

The closest possible representation of the decimal number 13.8 is shown below.



Mantissa



Exponent

By converting this bit pattern back into denary it can be seen that the actual number stored is 13.75, not 13.8.

1	1	.	5
---	---	---	---

Calculate the absolute error that has occurred.

[1 mark]

---



---

**Answer** \_\_\_\_\_

1	1	.	6
---	---	---	---

Calculate the relative error that has occurred. Express your answer as a percentage to two decimal places.

[1 mark]

---



---

**Answer** \_\_\_\_\_

**END OF QUESTIONS**

9
---

**Copyright Information**

For confidentiality purposes, from the November 2015 examination series, acknowledgements of third party copyright material will be published in a separate booklet rather than including them on the examination paper or support materials. This booklet is published after each examination series and is available for free download from [www.aqa.org.uk](http://www.aqa.org.uk) after the live examination series.

Permission to reproduce all copyright material has been applied for. In some cases, efforts to contact copyright-holders may have been unsuccessful and AQA will be happy to rectify any omissions of acknowledgements. If you have any queries please contact the Copyright Team, AQA, Stag Hill House, Guildford, GU2 7XJ.

Copyright © 2017 AQA and its licensors. All rights reserved.

