Computer Science AS & A Level | OCR | H046 & H446

2015 specification
first exams in 2017 (2016 for AS)

OCR

TECHNICAL TOPICS

Presentations and Worksheets

for A Level OCR Computer Science

Includes AS and A Level

AK10/ 6157

POD 6157

zigzageducation.co.uk

Contents

Teacher's Introduction

This resource is designed to support the delivery of the logical and mathematical topics from the A Level OCR specification (for first teaching in September 2015; first exams from June 2017).

The topics covered are as follows:

1. *Data Compression* *
2. *Encryption* *
3. *Relational Databases*
4. *Database Normalisation* *
5. *Structured Query Language* *
6. *Binary and Hexadecimal*
7. *Binary Arithmetic*
8. *Negative Numbers*
9. *Fractions* ++
10. *Arrays*
11. *Stacks and Queues*
12. *Linked Lists and Hash Tables* *
13. *Graphs and Trees* *
14. *Boolean Algebra* ++
15. *Logic Gates* ++
16. *Sequence and Selection*
17. *Iteration*
18. *Recursion* *
19. *Subroutines*
20. *Assembly Language*
21. *Graph and Tree Traversal* *
22. *Searching Algorithms* *
23. *Sorting Algorithms* *
24. *Shortest Path Algorithm* *
25. *PageRank Algorithm* *
26. *Big O Notation* *

* This entire topic is for A Level only
++ This topic is covered at AS but also contains some A Level-only content

For each of the topics above, there is an animated presentation, providing a step-by-step walk-through of the key concept, plus a worksheet giving students the opportunity to demonstrate their understanding.

These presentations and accompanying worksheets can be used in a number of ways:

- ✓ The animated presentations and worksheets can be used in class to introduce topics.
- ✓ The worksheets can be used as homeworks to test understanding.
- ✓ The animated presentations make perfect revision aids.
- ✓ As part of a flipped classroom, where students watch the animated presentations as preparation for the lesson. The students could complete the worksheets in class to test their understanding prior to a more in-depth discussion of the topic.

The animated presentations are provided in PowerPoint (PPTX), HTML5 and PDF formats. The HTML5 versions are included so that students can use the presentations more easily on devices which lack PowerPoint support (such as tablet computers and even smartphones), making them great for revision. Hard copies of the PDF versions have been included at the back of this pack.

Answers are provided for each worksheet which facilitate self and peer assessment.

As this resource also includes all the content needed for the separate AS qualification (for first teaching in September 2015, with the first exams in June 2016), content which is only required for the A Level course is indicated with the icon shown on the right.

A Level Only

The CD-ROM contains the animated presentations in three formats (PPTX, HTML5 and PDF), which are linked together via a HTML frontend **(index.html)**.

If using on a network, it is recommended that you provide a shortcut to the frontend to allow easy access for your students.

Alternatively, you can access the individual files directly (without using the frontend), simply by navigating to the relevant folder on the CD.

Data Compression

1. Below is an image encoded in binary. Apply run-length encoding to eac

1	1	0	1	1	→	
1	1	0	1	1	→	
0	0	0	0	0	→	
1	1	0	1	1	→	
1	1	0	1	1	→	

2. Apply run-length encoding (RLE) to this string of text: (1)

   CCCCCLL[illegible]QQ[illegible]

3. Suggest possible codes that could be used to represent each line of the dictionary-based compression. (5)

1	1	0	1	1	→	
1	1	0	1	1	→	
0	0	0	0	0	→	
1	1	0	1	1	→	
1	1	0	1	1	→	

4. State one advantage and one disadvantage of lossy compression comp

Advantage:	Disadvantage:

Encryption

1. Encrypt the plain text below using the Caesar cipher with a right shift of

   `COMPUTER SCIENCE ROCKS`

2. Encrypt the plain text below using the Caesar cipher with a right shift of

   `HAIL CAESAR`

3. Decrypt the cipher text below using the Caesar cipher with a right shift

   `UGEWTG`

4. Decrypt the cipher text below using the Caesar cipher with a right shift

   `EOHWFKOHB`

5. Describe the symmetric encryption method. (2)

6. Describe the asymmetric encryption method. (3)

7. State which is the most secure and explain why. (3)

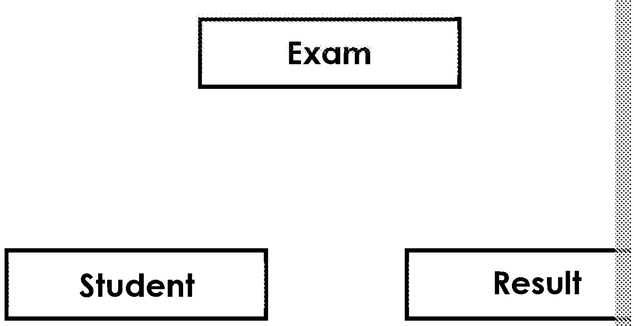
Relational Databases

An exam board uses a database to store the results of each exam for each s
tables: Student, Exam and Result.

1. Details of the three tables are displayed below. Underline the correct fie
for each table. (3)

```
Student (StudentID, Forename, Surname, School, Gender, Da
Exam (ExamID, ExamName, Subject, Level)
Result (ExamID, StudentID, RawMark, Grade)
```

2. Draw the correct relationships between each of the tables. (2)



3. Which part of a database stores all the data about one object or person

4. Underline the foreign keys used in the database. (2)

```
Student (StudentID, Forename, Surname, School, Gender, Da
Exam (ExamID, ExamName, Subject, Level)
Result (ExamID, StudentID, RawMark, Grade)
```

Database Normalisation

1. The following data is non-atomic; convert it into atomic data. (3)

   `Lesson: Computing Tuesday 12:30`

2. State whether or not the following table is in 1NF and give your reasons

MemberID	Name	Telephone no.
001	[illegible]	020 7625 1234 07345 678910

3. Shown below is the structure for a database for an airline. Use the tables database in 2NF. (2)

FlightID*	DestinationID*	DestinationCountry	DepartureTime	A

Flight

Destination

4. Shown below is the structure for a database for an online store. Use the database in 3NF. (2)

OrderID*	ProductID	ProductName	Price	Quantity

Order

Product

Structured Query Language

An exam board uses a database to store the results of each exam for each s
database has three tables: Student, Exam and Result.

```
Student (StudentID, Forename, Surname, School, Gender, DateOf
Exam (ExamID, ExamName, Subject, Level)
Result (ExamID, StudentID, RawMark, Grade)
```

1. Student number 10 has completed exam number 22 with a raw mark of
   Write the SQL commands to make this entry. (2)

2. Student number 6 has moved to Camden School. Write the SQL comma
   the student's entry. (3)

3. Student number 8 was entered by mistake. Write the SQL commands to

4. The exam board want to produce a list of a[illegible] with an A grade. Th
   appear for each result: ExamID, St[illegible], [illegible]ade. Write an SQL query to

5. The exam board want to produce a list of all students that achieved a m
number 17 sorted by raw mark in ascending order. The following inforr
student: StudentID, Forename, Surname, School, RawMark. Write an SQ

6. Write the SQL instruction to create the Result table. (4)

Binary and Hexadecimal

1. Convert the decimal number 45 into binary. (2)

128	64	32	16	8	4	2	1

2. Convert the binary number 01100101 into decimal. (2)

3. Convert the decimal number 165 into hexadecimal. (2)

4. Convert the hexadecimal number B6 into decimal. (2)

5. Convert the hexadecimal number 9B into binary. (2)

6. Convert the binary number 10010010 into hexadecimal. (1)

Binary Arithmetic

Complete the following calculations (give your answers using 5 bits). (24)

1.
```
0111
0101 +
```

2.
```
0101
0111 +
```

3.
```
1011
0111 +
```

4.
```
1011
0011
```

5.
```
1[illegible]
1011 +
```

6.
```
0111
1010 +
```

7.
```
0111
0111 ×
```

8.
```
1101
0111 ×
```

9.
```
1011
1111 ×
```

10.
```
1011
0101 ×
```

11.
```
1011
1001 ×
```

12.
```
0111
1000 ×
```

Negative Numbers

1. Convert this sign and magnitude binary integer to decimal. (2)
   `10110110`

2. Convert this sign and magnitude binary integer to decimal. (2)
   `00110010`

3. Convert this two's complement binary integer to decimal. (2)
   `101101`[illegible]

4. Convert this two's complement binary integer to decimal. (2)
   `00110010`

5. Convert this decimal number to an 8-bit two's complement binary integ
   `-122`

6. Convert this decimal number to an 8-bit two's complement binary integ
   `72`

7. Convert this decimal number to an 8-bit two's complement binary integ
   `-98`

Complete the following calculations. (6)

8.
```
  0111
- 0101
```

9.
```
  1101
- 0111
```

10.
```
  0011
- 0111
```

Fractions

Note: you only need to use two's complement with floating-point numbers, not fix

1. Convert this fixed-point binary number to decimal: (2)

1	0	1	1 ●	1	0	0	0

2. Convert this decimal value to fixed-point binary, with 4 bits before and 4

7.5

3. Convert this fixed point binary number to decimal: (2)

1	0	0	1 ●	1	1	0	0

4. Convert this decimal value to fixed-point binary, with 4 bits before and 4

12.625

---

5. Convert this floating-point binary number to decimal: (2)

Mantissa

0 ●	1	0	1	0

Exponent

0	0	1	0

6. Convert this floating-point binary number to decimal: (2)

Mantissa

1 ●	1	0	1	1

Exponent

0	0	1	1

7. Represent this decimal number using floating-point binary, using 8 bits the exponent: (2)

`-4.5`

8. The decimal value 17.27 has been represented as 17.25. Calculate the a

Absolute:	Relative:

9. Normalise this floating-point binary number: (2)

1 • 1	0	1	1	0

Mantissa

0	1	0

Exponent

10. Normalise this floating-point binary number: (2)

0 • 0	0	1	1	0

Mantissa

0	0	1

Exponent

Arrays

Use the names array shown below to help you to answer the following ques

Index	0	1	2	3	4
Value	Susan	Ian	Barbara	Steven	Ben

1. State the value of names[3]. (1)

2. State the value of names[?]. (1)

3. How can the value 'Ben' be accessed in the names array? (1)

4. How can the value 'Sarah' be accessed in the names array? (1)

5. Create the names array using pseudocode. (2)

6. Write a FOR loop that will cycle through the names array, outputting ea

Use the scores array shown below to help you to answer the following ques

	0	1	2	3
0	45	71	34	5
1	23	82	57	3
2	18	31	53	5
3	32	58	97	4

7. State the value of scores[3][1]. (1)

8. State the value of scores[1][2]. (1)

9. How can the value 71 be accessed in the scores array? (1)

10. How can the value 97 be accessed in the scores array? (1)

11. Create the scores array using pseudocode. (3)

Stacks and Queues

1. Draw a diagram below to show the state of a stack after the values Polly and Ben have been added to it. (4)

2. Draw a [illegible]m below to show the state of the stack from question 1 a

3. What is the current value of the variable used to store the position of th

4. Draw a diagram below to show the state of a circular queue after the va have been added to it. Make sure you also show the location of the Fro

5. Draw a diagram below to show the state of the circular queue from que
left the queue. Show the location of the Front and Rear pointers. (2)

6. Draw a diagram below to show the state of the circular queue from que
been added to the queue. Show the location of the Front and Rear pointers. (2)

Linked Lists and Hash Tables

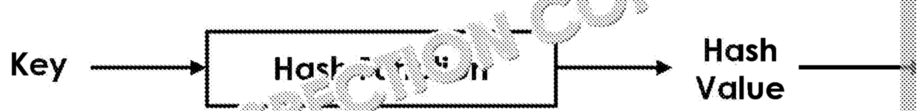
1. Describe the drawback of arrays and how this can be addressed using li

2. Describe the linked list data structure. (2)

3. Describe the drawback of linked lists and how this can be addressed usi

4. Describe the hash table data structure. (2)

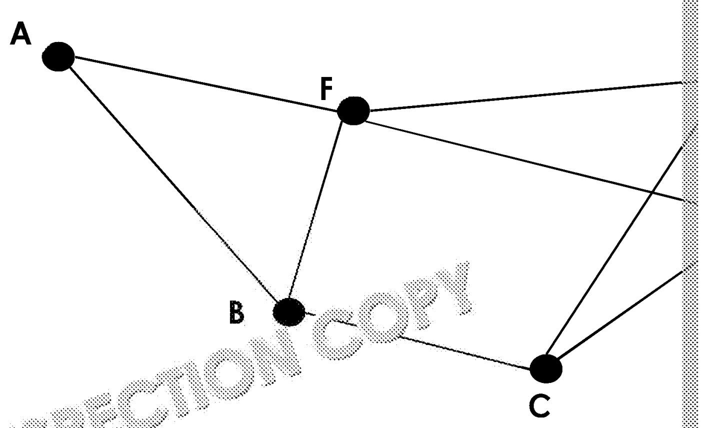
5. The following hash table uses a hash function that generates a hash val
the key. Write the value 'David' in the appropriate place in the table. (1)



6. Show how the value 'Diane' would be added to the hash table above us

Graphs and Trees

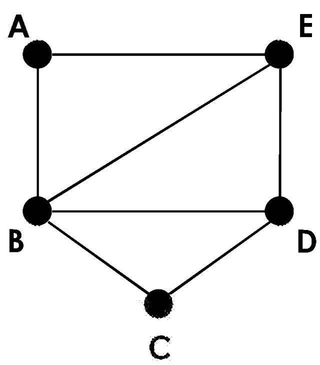
1. Label the graph to show examples of vertices and edges. (2)



2. State two vertices that are neighbours in the graph above. (1)

3. What is the degree of F in the graph above? (1)

4. Complete the adjacency matrix for the graph shown below. (5)



	A	B	C	D
A				
B				
C				
D				
E				

5. Complete the adjacency list for the graph shown above. (5)

Vertex	Adjacent Vertices
A	
B	
C	
D	
E	

Boolean Algebra

1. Simplify the following using a Karnaugh map. (4)
   (B ^ C) v (¬A ^ B) v (¬C ^ B)

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	[illegible]	[illegible]	
1	[illegible]	1	

A\BC	00	01	1
0			
1			

Simplified expression:

2. Simplify the following using a Karnaugh map. (5)
   (A ^ ¬C) v (A ^ ¬B) v (B ^ ¬C) v (B ^ C)

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

A\BC	00	01	1
0			
1			

Simplified expression:

---

3. Which of the following is not a statement of De Morgan's Theory? (1)
   ¬ (A v B) = ¬A ^ ¬B   [illegible]A ^ [illegible]) = ¬(A v B)   ¬(A ^

4. Simplify the following using De Morgan's Theory. (3)
   ¬ ((X v Y) ^ (Y v Z))

Logic Gates

1. Draw the following logic gates: (3)

AND	OR	

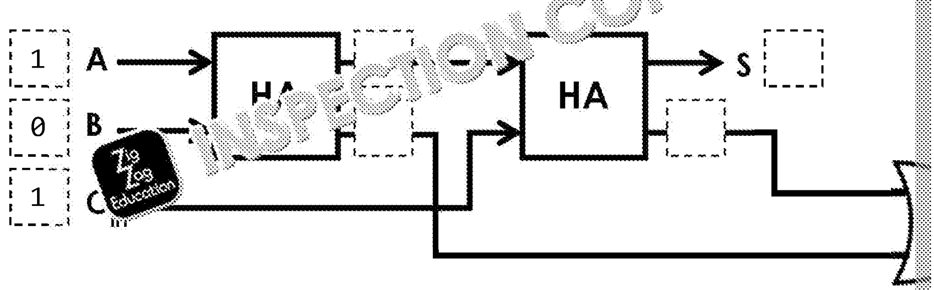
2. Complete this truth table for an OR gate. (4)

A	B	A v B

3. Draw the logic diagram for this expression. (3)

¬ ((A ^ B) v C)

4. Label this full adder diagram showing the outputs at each stage. (5)



Sequence and Selection

Program A	Program B

`01 INPUT A, B`
`02 Total = A + B`
`03 Average = Total / 2`
`04 OUTPUT Average`
	`01 INPUT A, B`
	`02 IF A == B THE`
	`03     OUTPUT`
	`04 ELSE IF A > B`
	`05     OUTPUT`
	`06 ELSE`
	`07     OUTPUT`
	`08 END IF`

1. Which one of the two programs shown above is an example of a seque

2. Give the line number of a statement from the program you identified in question. (1)

3. Identify the different variables that are used in Program A. (3)

4. Which one of the two programs shown above is an example of selectio

5. Give the line numbers of the conditions from the program you identifie question. (2)

6. Write the meaning of each of the comparison operators shown the in ta

Operator	Meaning
>	
<	
>=	
<=	
=	
<>	

Iteration

Program A

```
01 INPUT Num
02 FOR i = 1 to Num
03       OUTPUT i
04 NEXT i
```

Program B

```
01 Password = "
02 WHILE Passwo
03       INPUT
04 END WHILE
05 OUTPUT "Pass
```

1. Which one of the two programs shown above features an example of a

2. Give the line number of the condition from the program you identified i question (1)

3. Which one of the two programs shown above features an example of a

4. Describe the purpose of Program A. (3)

5. Describe the purpose of Program B. (3)

6. Explain the difference between REPEAT UNTIL and WHILE loops. (3)

7. Rewrite Program A using a WHILE loop. (3)

Recursion

Program A

```
01 PROCEDURE count(C)
02       IF A <= 0 THEN
03             OUTPUT 0
04       ELSE
05             OUTPUT C
06             count(C-1)
07       END IF
08 END PROCEDURE
```

Program B

```
01 PROCEDURE count(C)
02       WHILE C >= 0
03             OUTPUT C
04             C = C -1
05       END WHILE
06 END PROCEDURE
```

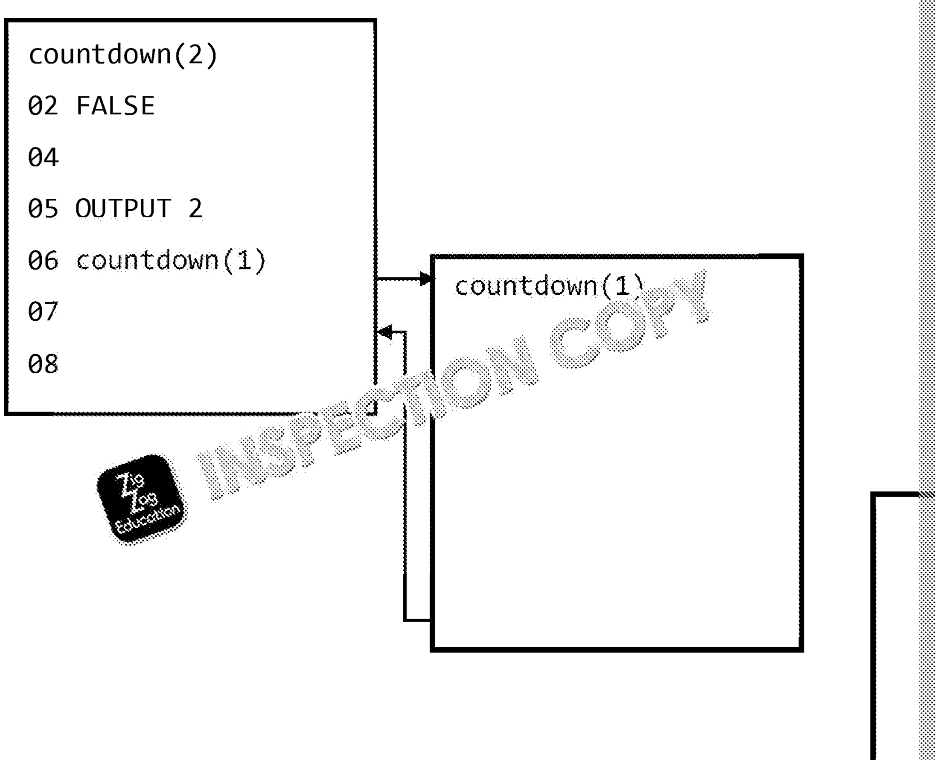
P

1. In which of the programs shown below does recursion occur? (1)

2. Give the line number where recursion occurs in the program you identi previous question. (1)

3. Which two programs will produce the same result? (1)

4. Complete the diagram below to show what happens when Program A i countdown(2). (7)



5. Complete the trace table below to show what happens when Program A
countdown(3). (4)

C	OUTPUT

6. What is a disadvantage of using recursion instead of iteration? (3)

Subroutines

Program A	Program B	P
`__________ areaCalc(W, H)`	`__________ average(A, B, C)`	
`       Area = W * H`	`       Total = A + B + C`	
`       OUTPUT Area`	`       Average = Total / 3`	
`END __________`	`       RETURN Average`	
`areaCalc(10, 8)`	`END __________`	
	`average(4, 3, 4)`	

1. Identify a program that contains a function from the three shown above

2. Identify the parameters that are defined in Program B. (3)

3. Identify the arguments that are passed to the areaCalc subroutine in Pr

4. Identify the arguments that are passed to the compare subroutine in Pr

5. What would be the output of the areaCalc subroutine based on the call

6. What would be the output of the compare subroutine based on the call

7. Describe the difference between a procedure and a function. (2)

Assembly Language

1. Describe what is happening at each line if a user inputs the values 5 and

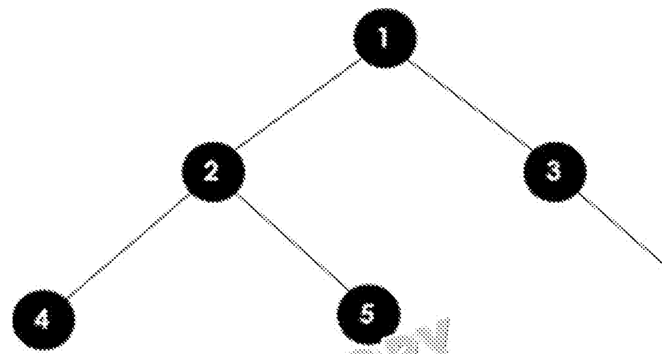
Label	Opcode	Operand	Descrip
	INP		
	STA	Num1	
	INP		
	SUB	Num1	
	OUT		
	HLT		
Num1	DAT		

2. Write t[illegible]bers of the lines that would be executed when the progra
input. (5)

Line	Label	Opcode	Operand
01		INP	
02	loop	OUT	
03		SUB	one
04		BRZ	stop
05		BRA	loop
06	stop	HLT	
07	one	DAT	1

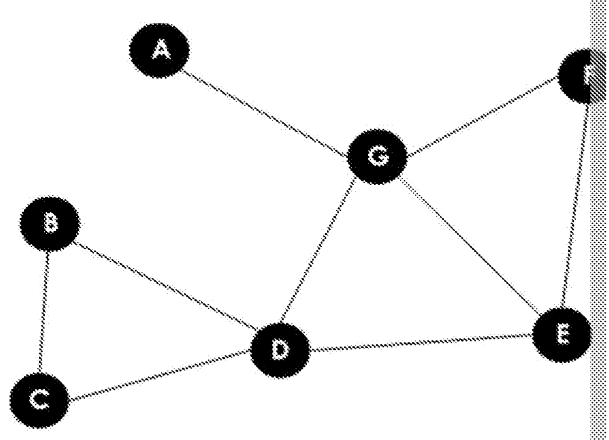
3. Explain what the program above does. (3)

Graph and Tree Traversal



1. Write the sequence in which the nodes in the tree below will be visited w

2. Write the sequence in which the nodes in the tree above will be visited wh



3. Write the sequence in which the nodes in the tree above will be visited w

4. Complete a depth-first traversal on the graph above. (7)

Vertex Visited	Stack

Searching Algorithms

1. Explain how the linear search algorithm works. (2)

2. How many comparisons would be needed to find the value 23 in the lis

   5, 9, 11, 15, 23, 45, 54, 58, 61

3. Create a binary search tree for the list shown below. (3)

   45, 32, [illegible], 9, 38, 23, 6

4. Create a binary search tree for the list shown below. (3)

   56, 12, 4, 7, 98, 32, 65, 86, 26, 77

5. How many comparisons would be needed to find the value 32 in the tre

6. How many comparisons would be needed to find the value 65 in the tre

Sorting Algorithms

1. Show the list below at each stage of sorting using the bubble sort algor

   `45, 32, 2, 78, 5, 9`

2. Show the list below at each stage of sorting using the merge sort algori

   `45, 32, 2, 78, 5, 9, 38, 23`

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td colspan="4"></td><td colspan="4"></td></tr>
<tr><td colspan="8"></td></tr>
</table>

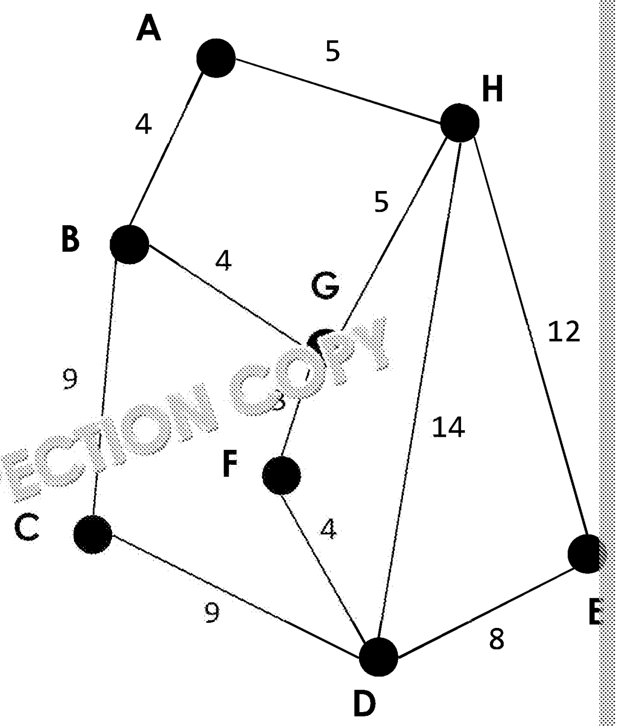
3. Show the list below at each stage of sorting using the insertion sort alg

   `45, 32, 2, 78, 5, 9`

4. Show the list below at each stage of sorting using the quick sort algorit

   `45, 32, 2, 78, 5, 9`

Shortest Path Algorithm



1. Use Dijksta's shortest path algorithm to find the shortest path between
on the graph. (8)

Node	Shortest Distance from Vertex A	Previous Node

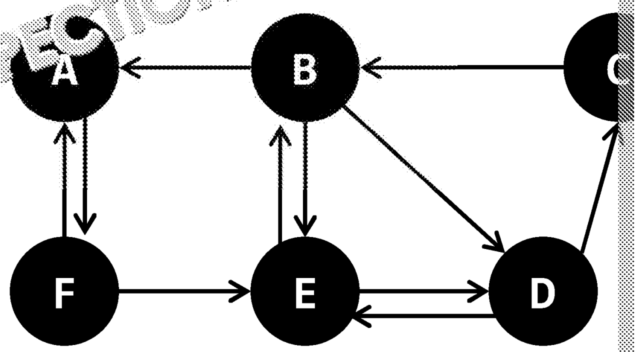
2. What is the shortest path between vertex A and vertex G? (1)

3. What is the shortest path between vertex A and vertex E? (1)

PageRank Algorithm

1. Describe the purpose of the PageRank algorithm. (3)

2. The graph below represents a very simple online website with six pages highest rank when the PageRank algorithm is applied? (1)



3. Describe how the PageRank algorithm works. (2)

4. Explain how weighting works in the PageRank algorithm. (3)

Big O Notation

1. Complete the table below comparing each type of time complexity. (5)

Complexity	Description
Constant	
Linear	
Polynomial	
Exponential	
Logarithm	

2. Which standard algorithm has linear complexity? (1)

3. Which standard algorithm has polynomial complexity? (1)

4. Which two standard algorithms have logarithmic complexity? (2)

5. Number the below from 1 to 5, 1 being the best performance and 5 bei

O(1)	
O(n)	
O(k^n)	
O(log(n))	
O(n^k)	

Answers

Data Compression

1. *1 mark for each row:*

1	1	0	1	1	→	2(1), 1(0), 2(1)
1	1	0	1	1	→	2(1), 1(0), 2(1)
0	0	0	0	0	→	5(0)
1	1	0	1	1	→	2(1), 1(0), 2(1)
1	1	0	1	1	→	2(1), 1(0), 2(1)

2. 5(C), 5(L), 3(Q), 4(P)
3. *1 mark for each row; accept any two suitable values:*

1	1	0	1	1	→	[illegible]
1	1	0	1	1	→	00
0	0	0	0	0	→	01
1	1	0	1	1	→	00
1	1	0	1	1	→	00

4. Advantage: lossy compression usually results in significantly smaller files.
   Disadvantage: permanently destroys some of the data.

Encryption

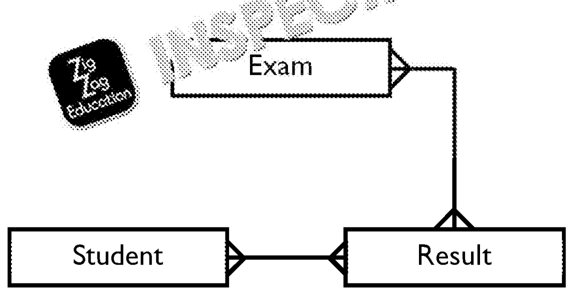
1. EQORWVGT UEKGPEG TQEMU
2. LEMP GEIWEV
3. SECURE
4. BLETCHLEY
5. Uses one key (1) to both encrypt and decrypt the message (1).
6. Uses separate keys to encrypt and decrypt the message (1); a public key w used to encrypt the message (1), and a private key which is kept private a message. (1).
7. Asymmetric encryption (1) because only the public key is shared (1) and t not decrypt it (1).

Relational Databases

1. *1 mark for each primary key*

```
Student (StudentID, Forename, Surname, School, Gender, Date
Exam (ExamID, ExamName, Subject, Level)
Result (ExamID, StudentID, RawMark, Grade)
```

2. *1 mark for each correct relationship*



3. Record
4. *1 mark for each correctly identified foreign key*

```
Student (StudentID, Forename, Surname, School, Gender, Date
```

```
Exam (ExamID, ExamName, Subject, Level)
Result (ExamID, StudentID, RawMark, Grade)
```

Database Normalisation

1. *1 mark for each correctly identified field:*
   Subject: Computing, Day: Tuesday, Time: 12:30
2. No (1) because there are two telephone numbers in the telephone field (1) per field (1)
3. Flight

FlightID*	DestinationID*	DepartureTime	ArrivalTime

Destinations

DestinationID*	DestinationCountry

4. Order

OrderID*	ProductID	Quantity	TotalPrice

Products

ProductID	ProductName	Price

Structured Query Language

1. *1 mark for each correct statement*

```
INSERT INTO Result
VALUES (22, 10, 67, "A")
```

2. *1 mark for each correct statement*

```
UPDATE Student
SET School = "Camden School"
WHERE StudentID = 6
```

3. *1 mark for each correct statement*

```
DELETE FROM Student
WHERE StudentID = 8
```

4. *1 mark for each correct statement*

```
SELECT ExamID, StudentID, Grade
FROM Result
WHERE Grade = "A"
```

5. *1 mark for each correct statement*

```
SELECT StudentID, Forename, Surname, School, RawMark
FROM Student, Result
WHERE Student.StudentID = Result.StudentID
AND RawMark >= 50 AND [illegible] = 17
ORDER BY RawMark
```

6. *1 mark for each correctly defined field*

```
CREATE TABLE (
ResultID INT PRIMARY KEY NOT NULL
StudentID INT
RawMark INT
Grade VARCHAR(2) or Grade VARCHAR(1)
)
```

Binary and Hexadecimal

1. 00101101 *(1 mark for the correct answer and 1 mark for working)*
2. 101 *(1 mark for the correct answer and 1 mark for working)*
3. A5 *(1 mark for the correct answer and 1 mark for working)*
4. 182 *(1 mark for the correct answer and 1 mark for working)*
5. 10011011 *(1 mark for the correct answer and 1 mark for working)*
6. 92 *(1 mark for the correct answer and 1 mark for working)*

Binary Arithmetic

(1 mark for the correct answer and 1 mark for working)

1. 01100
2. 01100
3. 010010
4. 011[illegible]
5. 010[illegible]
6. 010001
7. 110001
8. [illegible]
9. 10100101
10. 110111
11. 1100011
12. 111000

Negative Numbers

(1 mark for the correct answer and 1 mark for working)

1. -54
2. 50
3. -74
5. 50
5. 10000110
6. 01001000
7. 10011110
8. 0010
9. 0110
10. 1100

Fractions

1. 11.5 *(1 mark for the correct answer and 1 mark for working)*
2. 0111.1000 *(1 mark for the correct answer and 1 mark for working)*
3. 9.75 *(1 mark for the correct answer and 1 mark for working)*
4. 1100.1010 *(1 mark for the correct answer and 1 mark for working)*
5. -2.5 *(1 mark for the correct answer and 1 mark for working)*
6. 13.5 *(1 mark for the correct answer and 1 mark for working)*
7. Mantissa: 1.0[illegible]1000 Exponent: 0011
8. Absolute: 0.02 Relative: 0.001158
9. Mantissa: 0.110110 Exponent: 0101
10. Mantissa: 0.110 Exponent: 0001

Arrays

1. Steven
2. Ian
3. `names[4]`

4. `names[6]`
5. *1 mark initialising the names array correctly, 1 mark for using " " around th*

```
names ← ["Susan", "Ian", "Barbara", "Steven", "Ben", "Polly
```

6. *1 mark for loop with correct range, 1 mark for using the counter variable (i) value from the names array, 1 mark for correct indentation*

```
FOR i ← 0 to 6
    OUTPUT names[i]
NEXT i
```

7. 58
8. 57
9. scores[0][1]
10. Scores[3][2]
11. *1 mark for initialising the cor[illegible]y, 1 mark for initialising it as a 2D arra*

```
scores ← [[4[illegible], 24, 55],[23, 82, 57, 37],[18, 31, 53, 57
```


Stacks and Queues

1. *1 mark for correct top value, one for correct bottom value and 2 marks for c*

Index	Value
4	Ben
3	Zoe
2	Jamie
1	Polly

2. *1 mark for the removal of the correct value*

Index	Value
4	
3	Zoe
2	Jamie
1	Polly

3. 3
4. *1 mark for each correct row, one for each correctly positioned pointer*

Index	Value	
4	Polly	← Front
3	Jamie	
2	Zoe	
1	Ben	← Rear

5. *1 mark for the removal of the correct value and 1 mark for the correctly pos*

Index	Value	
4		
3	Jamie	← Front
2	Zoe	
1	Ben	← Rear

6. *1 mark for the addition of the new value in the correct position and 1 mark f pointer*

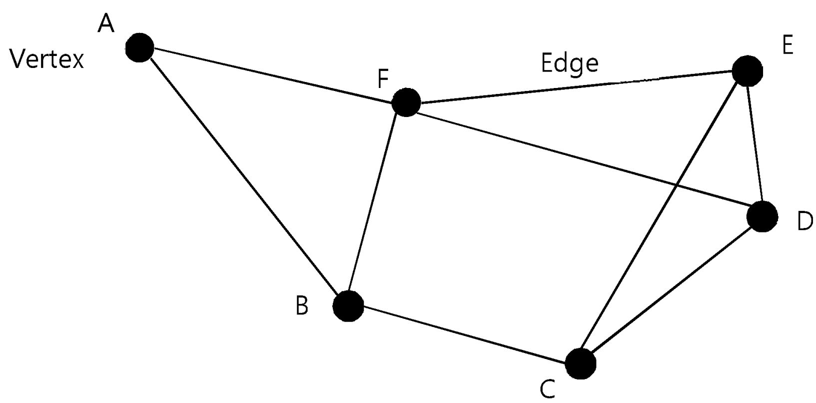
Index	Value	
4	Mel	← Rear
3	Jamie	← Front
2	Zoe	
1	Ben	

Linked Lists and Hash Tables

1. When an array is created its size is declared and it is allocated a section of fixed (1). Linked lists consist of nodes and each one can be stored in any f size flexible (1).
2. Each node in a linked list contains a pointer to the memory location of the node in the list is indicated using the ∅ symbol (1). New nodes can easily b the pointers (1).
3. A disadvantage of linked lists is elements can't be directly accessed. The e the element (1). Hash tables offer a solution to this problem, being both f access to elements (1).
4. Hash tables consist of two parts: an array with an associated hash function piece of data known as a key and generates a hash value; this is used as th
5. The value 'David' should be written in position [illegible].
6. The value 'Diane' should be written in position 4.

Graphs and Trees

1. *Any edge and vertex correctly labelled; for example:*



2. *Any suitable example*, for example A and F
3. 4
4. *1 mark for each correct line:*

	A	B	C	D	E
A	0	1	0	0	1
B	1	0	1	1	1
C	0	1	0	1	0
D	0	1	1	0	1
E	1	1	0	1	0

5. *1 mark for each correct line:*

Vertex	Adjacent vertices
[illegible]	B, E
[illegible]	A, E, D, C
C	B, D
D	C, B, E
E	A, B, D

Boolean Algebra

1. *1 mark for the correct outputs in the truth table, 1 mark for the correct Karn identification of the correct grouping and 1 mark for the correct simplified e*

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

A\BC	00	01	11	10
0	0	0	[illegible]	1
1	[illegible]	0	1	1

Simplified expression: B

2. *1 mark for the correct outputs in the truth table, 1 mark for the correct Karn identification of the correct grouping and 1 mark for the correct simplified e*

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A\BC	00	01	11	10
0	0	0	1	1
1	1	1	1	1

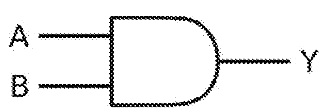
Simplified expression: A v B

3. ¬(A^B) = ¬ (A v B)

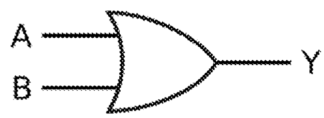
4. ¬ (X v Y) v ¬ (Y v Z) *(1 m[illegible] for initial use of De Morgan)*
   (¬X ^ [illegible]) v [illegible] *(1 mark for repeated use of De Morgan)*
   ¬Y [illegible] ¬Z) *(1 mark for final answer)*

Logic Gates

1. *AND gate:*



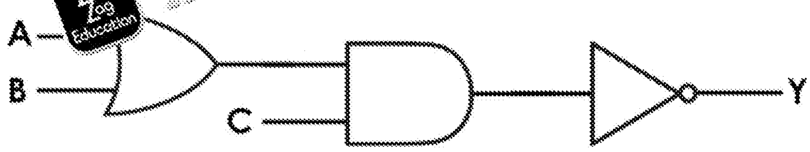
OR gate:



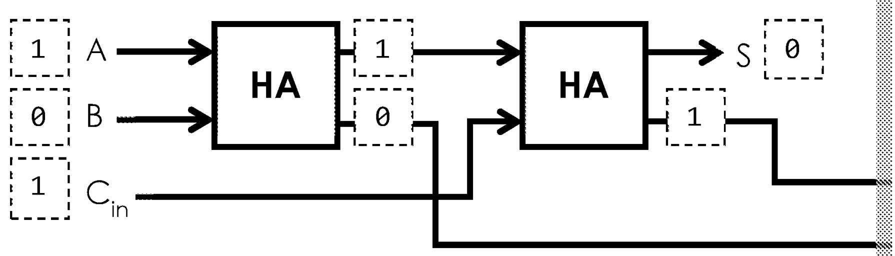
2. *1 mark for each correct line:*

A	B	A v B
0	0	0
0	1	1
1	0	1
1	1	1

3. *1 mark for each correctly placed symbol:*



4. *1 mark for each correctly labelled output:*



Sequence and Selection

1. Program A
2. *Any line from Program A*
3. A, B and Average
4. Program B
5. 02 and 04 (from program B)
6. *1 mark per correct row:*

Operator	Meaning
[illegible]	Greater than
[illegible]	Less than
>=	Greater than or equal to
<=	Less than or equal to
=	Equal to
<>	Not equal to

Iteration

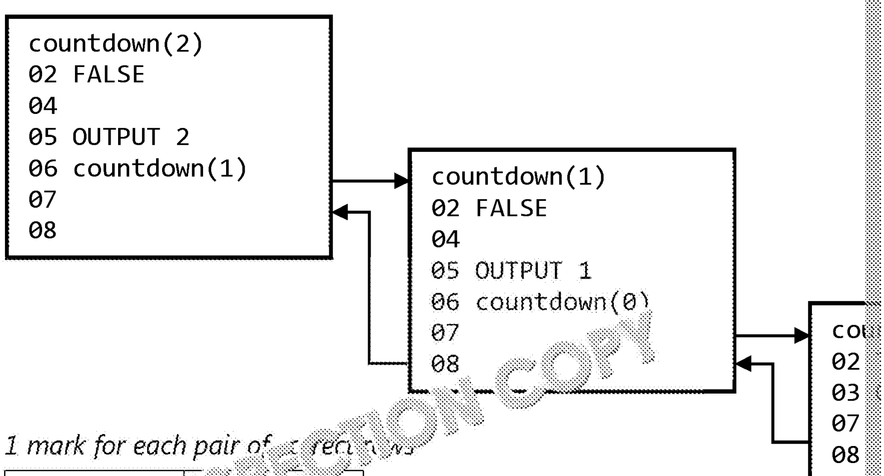
1. Program B
2. Line 02
3. Program A
4. It asks the user to input a number (1) and uses a FOR loop (1) to output th[illegible] the user inputted (1).
5. The value of the password variable is set to 'none' (1). A WHILE loop is use[illegible] input their password (1) until they enter the value 'turing' at which point it[illegible]
6. In a WHILE loop the condition is tested at the start, whereas in a REPEAT U[illegible] (1). A WHILE loop repeats until the condition is TRUE, whereas a REPEAT U[illegible] condition is FALSE (1). In a WHILE loop the statem[illegible] may never be execu[illegible] loop the statements will always be execute[illegible] once (1).
7. *Award 1 mark for each of the foll[illegible] also refer to the example below):*
   - The user is aske[illegible] number.
   - The co[illegible] been initialised correctly (value = 1) before the lo[illegible] co[illegible]dition is correct.

   Exam[illegible]

```
01 INPUT Num
02 i = 1
03 WHILE i <= Num
04     OUTPUT i
05     i = i + 1
06 END WHILE
```

Recursion

1. A
2. 06
3. A and B
4. *1 mark for each correctly completed box and 1 mark for each correctly plac[illegible]*



5. *1 mark for each pair of [illegible]*

[illegible]	[illegible]utput
[illegible]	
	3
2	
	2
1	
	1
0	
	0

6. There are many calls to the same subroutine (1), each with its own set of v[illegible] up more memory (1).

Subroutines

1. Program B
2. A, B and C
3. 10 and 8
4. 6 and 4
5. 80
6. 6
7. A procedure is a subroutine that does not normally return a value (1), whe returns values to the calling routine (1).

Assembly Language

1. *1 mark for each correct line:*

Label	Opcode	Operand	Description
	INP		Asks the user to input a value (5)
	STA	Num1	Stores the value in the accumulator (5)
	INP		Asks the user to input a value (7)
	SUB	Num1	Subtracts the value stored in Num1 (5) accumulator (7)
	OUT		Outputs the contents of the accumulat
	HLT		Stops the program
Num1	DAT		Reserves a memory location and labels

2. *1 mark for each correct iteration (line 07 not needed):*

   01, 02, 03, 04, 05,

   02, 03, 04, 05,

   02, 03, 04, 05,

   02, 03, 04, 05,

   02, 03, 04, 06

3. It asks the user to input a number (1) and counts down from that numbe

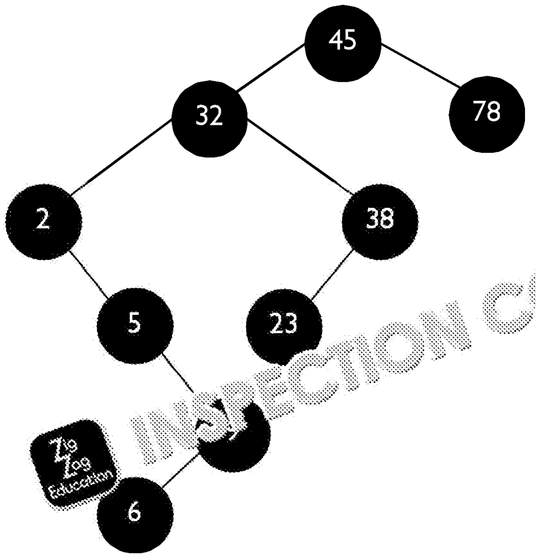
Graph and Tree Traversal

1. 1, 2, 4, 5, 3, 6
2. 4, 5, 2, 6, 3, 1
3. 4, 5, 2, 1, 6, 3
4. *Open mark for each correct line where a vertex is visited:*

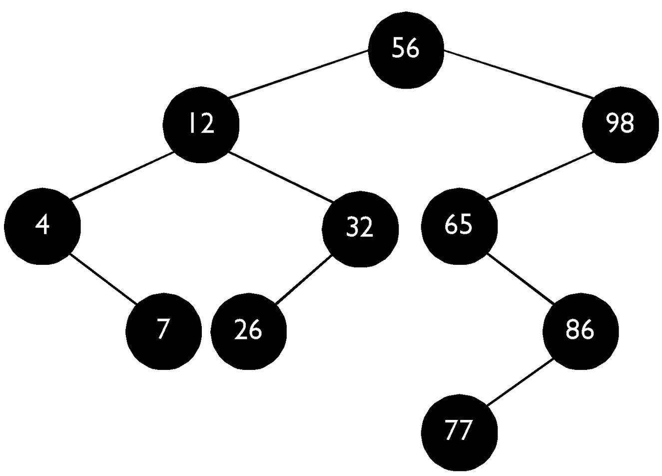
Vertex Visited	Stack
A	A
G	GA
D	DGA
	EDGA
	FEDGA
	EDGA
	DGA
B	BDGA
C	CBDGA

Searching Algorithms

1. Searches each element starting at the start of the list (1) until it finds a ma
2. 1
3. *1 mark for each correct pair of levels:*



4. *1 mark for correct first two levels, 1 mark for the third level, 1 mark for the f*



5. 2
6. 3

Sorting Algorithms

1. *1 mark per correct row:*

45	[illegible]	2	78	
[illegible]	45	2	78	
[illegible]	2	45	78	
[illegible]	2	45	5	
32	2	45	5	
2	32	45	5	
2	32	5	45	
2	32	5	9	
2	5	32	9	
2	5	9	32	

2. *1 mark per correct row:*

45	32	2	78	5	9
32, 45		2, 78		5, 9	
2, 32, 45, 78				5, 9	
2, 5, 9, 23, 32, 38, 45, 78					

3. *1 mark per correct row:*

45	32	2	78
32	45	2	78
32	2	45	78
2	32	45	78
2	32	[illegible]	5
2	32	[illegible]	45
2	[illegible]	32	45
2	[illegible]	32	45
[illegible]	5	32	9
[illegible]	5	9	32

4. *1 mark per correct row:*

45	32	2	78
2	5	9	45
2	5	9	32

Shortest Path Algorithm

1. *1 mark per correct row:*

Node	Shortest Distance from vertex A	Previous Node
A	0	
B	4	A
C	13	B
D	15	F
E	17	H
F	11	G
G	8	B
H	5	A

2. A – B – G
3. A – H – G

PageRank Algorithm

1. Used by search [illegible] determine the rank of webpages (1), and there[illegible] the [illegible]
2. E
3. It uses the number of inbound links a page has to calculate its rank (1); ea[illegible] vote on the importance of the page (1)
4. Some votes are given a greater a weighting than others (1). This is based [illegible] voting page has (1) the more inbound links, the greater the weighting of [illegible]

Big O Notation

1. *1 mark for each correct cell*

Complexity	Description
Constant	The time complexity remains the same regardless of the
Linear	The time complexity is proportional to the number of ite
Polynomial	The rate at which time complexity rises increases as the n
Exponential	The time complexity increases exponentially as the numb
Logarithmic	The increase in time complexity decreases as the number

2. Linear search
3. Bubble sort
4. Binary search and binary tree search
5. *1 mark per correct row:*

O(1)	1
[illegible])	3
$O(k^n)$	5
O(log (n))	2
$O(n^k)$	4