GradeMax

Cambridge International AS & A Level

Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

COMPUTER SCIENCE **9608/12**

Paper 1 Written Paper **October/November 2016**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge will not enter into discussions about these mark schemes.

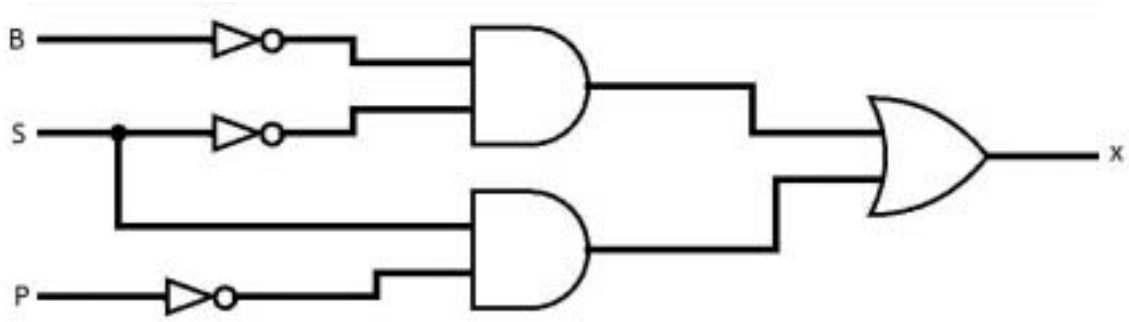
Cambridge is publishing the mark schemes for the October/November 2016 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

® IGCSE is the registered trademark of Cambridge International Examinations.

© UCLES 2016

Page 2	Mark Scheme	Syllabus	Paper
	Cambridge International AS/A Level – October/November 2016	9608	12

1 (a) **ONE** mark for each correct gate.



[6]

(b) **ONE** mark for each pair of rows.

B	S	P	Working space	X
0	**0**	**0**		1
0	**0**	**1**		1
0	**1**	**0**		1
0	**1**	**1**		0
1	**0**	**0**		0
1	**0**	**1**		0
1	**1**	**0**		1
1	**1**	**1**		0

[4]

© UCLES 2016

GradeMax

2 (a) The number of images / frames recorded per second / unit time.
// The frequency with which the images/frames are recorded.

[1]

(b) **ONE** mark per bullet point below. **MAX THREE** marks per type of encoding.

Interlaced encoding

- The data from a single frame are encoded as two separate **fields**.
- One containing the data for the even numbered rows / lines and the other has the data for the odd numbered rows / lines.
- The image is rendered by alternating between the even field and the odd field (of each successive frame).
- The viewer sees data from two frames simultaneously
- The rate of picture display (the field rate) is twice the rate of image frame display (the frame rate).
- Originally used in television broadcasting and adapted for video recordings.
- Produces what appears to the eye to be a high refresh rate.
- Halves the transmission bandwidth requirements.

Progressive encoding

- Stores the data for an entire frame and displays all the frame data at the same time.
- The rate of picture display is the same as the frame rate.
- Used by traditional film / video digitised from a film camera / computer displays progressive encoding.
- High bandwidth requirements.

[4]

(c) (i) **ONE** mark per term.

Description	Term
Pixels in two video frames have the same value in the same location. There is duplication of data between frames.	**Temporal redundancy**
A sequence of pixels in a single video frame have the same value.	**Spatial redundancy**

[2]

(ii) (File) compression

[1]

© UCLES 2016

GradeMax

3 **ONE** mark for each letter in the correct place.

Then **ONE** mark for any pair of letters in the correct order, but not in the correct place

1	The application program executes a statement to read a file.
2	G
3	The operating system begins to spin the hard disk, if it is not currently spinning.
4	F
5	D
6	H
7	C
8	B
9	A
10	E

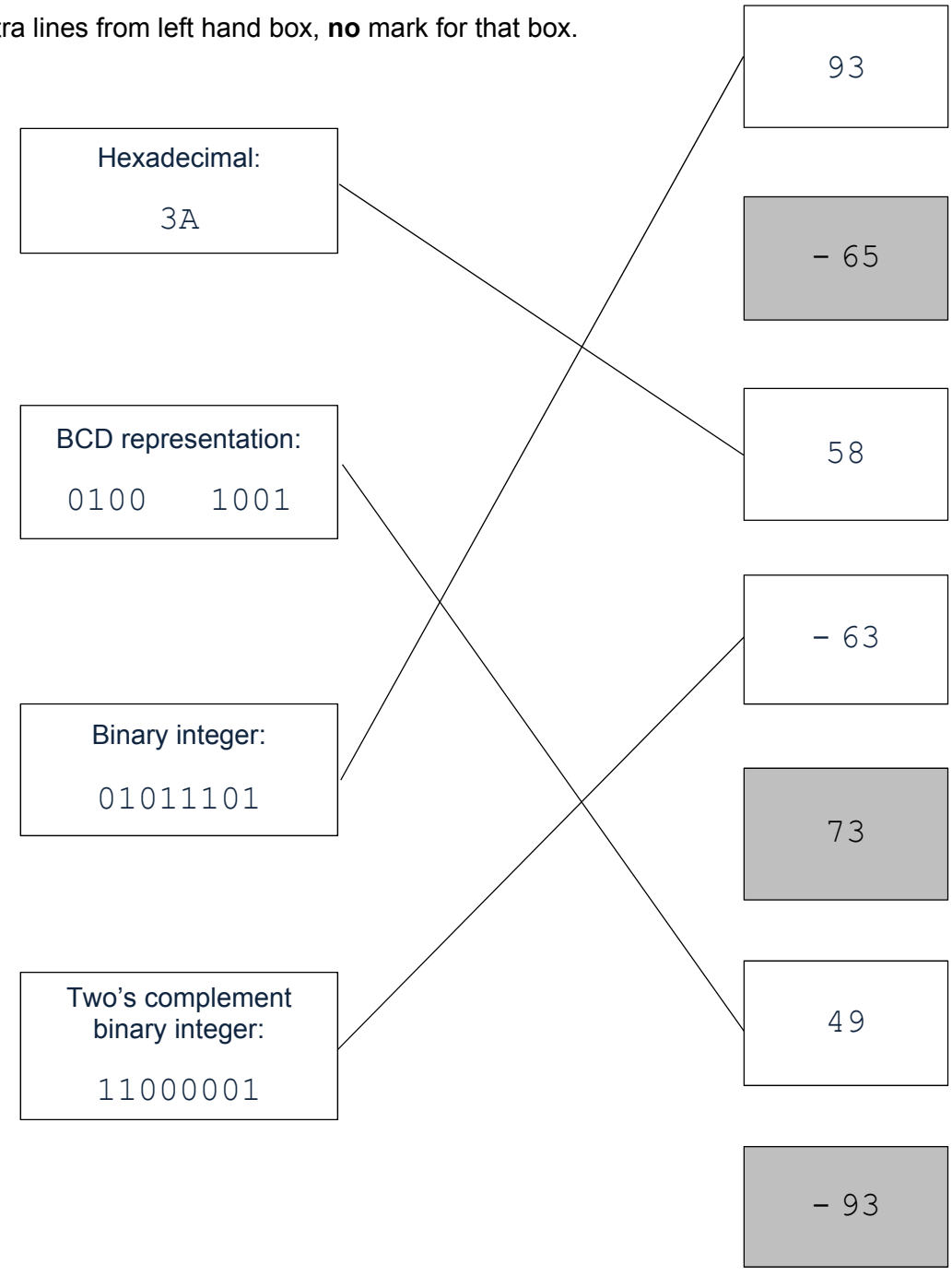
[8]

© UCLES 2016

GradeMax

4 **ONE** mark for each correct line.

Extra lines from left hand box, **no** mark for that box.



[4]

© UCLES 2016

GradeMax

5 (a) (i)

0	1	1	1	0	0	1	1

[1]

(ii) **ONE** mark for Accumulator contents, **ONE** mark for the explanation.

1	0	1	1	0	0	0	1

- Index Register holds the value 4; 101 + 4 = 105 so load data from address 105

[2]

(iii) **ONE** mark for Accumulator contents, **TWO** marks for the explanation.

0	1	0	0	1	0	1	1

- Memory address 103 contains the value 107
- So address 107 is the address from which to load the data [3]

(b) **ONE** mark for each correct row.

ACC	Memory address			
	810	**811**	**812**	**813**
	28	41	0	0
28				
29				
			29	
41				
70				
				70

[6]

© UCLES 2016

GradeMax

6 (a) **ONE** mark for each difference from the bullet points below.

- RAM loses content when power turned off / volatile memory / temporary memory
  ROM does not lose content when power turned off / non-volatile memory / permanent memory

- Data in RAM can be altered / deleted / read from and written to
  ROM is read only / cannot be changed / altered / deleted

- RAM stores files / data / operating system currently in use
  ROM is used to store BIOS / bootstrap / pre-set instructions [2]

(b) **THREE** from:

- DRAM has to be refreshed / charged
  // SRAM does not request a refresh

- DRAM uses a single transistor and capacitor
  // SRAM uses more than one transistor to form a memory cell
  // SRAM has more complex circuitry

- DRAM stores each bit as a charge
  // SRAM each bit is stored using a flip-flop/latch

- DRAM uses higher power (because it requires more circuitry for refreshing)
  //SRAM uses less power (no need to refresh)

- DRAM less expensive (to purchase / requires fewer transistors)
  // SRAM is more expensive (to buy as it requires more transistors)

- DRAM has slower access time / speed (because it needs to be refreshed)
  // SRAM has faster access times

- DRAM can have higher storage / bit / data density
  // SRAM has lower storage / bit / data density

- DRAM used in main memory
  // SRAM used in cache memory

[3]

© UCLES 2016

GradeMax

7 **ONE** mark per bullet point, **MAX TWO** marks per task.

- Process / resource management
- Scheduling of processes / multi-tasking / multi-programming etc.
- Resolution of conflicts when two or more processes require the same resource

- Main memory management
- Memory protection to ensure that two programs do not try to use the same space
- Use of virtual memory
- Deciding which processes need to be in main memory at any one time
- Location of processes within the memory
- By example, e.g. when process terminates, memory is made available

- Peripheral / hardware / device management
- Installation of appropriate driver software
- Controls access to data being sent to / from hardware / peripherals
- Controls access to hardware / peripherals
- Manages communication between devices / hardware and software

- File / secondary storage management
- Maintains directory structures
- Provides file naming conventions
- Controls access

- Security management
- Makes provision for recovery when data is lost
- Provides usernames and passwords / encryption / user accounts
- Prevents unauthorised access
- Ensures privacy of data

- Provision of a software platform / environment
- On which other programs can be run

- Interrupt handling
- Identifies priorities of interrupts
- Save current memory / process values / saves data on power outage
- Loads appropriate Interrupt Service Routine (ISR)
- Any relevant example

[4]

© UCLES 2016

GradeMax

8 (a) **ONE** mark for each bullet point from **MAX TWO** groups.

- The code is already written
- (So the programmer is not starting over again) which saves time

- The code will have been used by many people
- So it should be already thoroughly tested // relatively error-free

- The programmer can use, e.g. mathematical / graphics functions, etc. (may not know how to code)
- Can be sure that the function will perform as it should // simplifies the program.

- The code should conform to industry standards
- And therefore contribute towards a more robust program

[4]

(b) (i) **ONE** mark for each benefit, and **ONE** mark for a further expansion.

- The executable file is smaller / the executable does not contain all the library routines …
- … DLL files are only loaded into memory when required.

- Changes / improvements / error correction to the DLL file code are done independently of the main program...
- … So there is no need to recompile the main program
- … All programs using it will benefit

- A single DLL file can be made available to several application programs...
- … Saving space in memory / easing the pressure on memory

[4]

(ii) **ONE** mark for each bullet point from **MAX ONE** group.

- The executable code is not self-contained …
- … the DLL file(s) needed to be included at run time.

- Appropriate (linking) software must be available at run-time …
- … to link / include / import the DLL files.

- The DLL file must be present …
- … otherwise (unable to find X.dll) errors

- Unexpected changes to the DLL file / corrupted DLL file …
- … could mean the program stops working as expected

- Malicious changes to the DLL file …
- … could install a virus on the user's computer / related files could be corrupted

[2]

© UCLES 2016

GradeMax

9 (a) **ONE** mark for each reason and **ONE** mark for a further explanation. **MAX THREE** reasons.

- Reduced data redundancy / data duplication
- Data is stored in (separate) linked tables

- The database (generally) stores data only once / data need only be updated once
- Improved data consistency / integrity / associated data will be automatically updated / easier to maintain the data / elimination of unproductive maintenance

- Complex queries can be more easily written
- To search / find specific data // specific example related to the Health Club

- Fields can be more easily added to or removed from tables
- Without affecting existing applications (that do not use these fields)

- Program-data dependence is overcome
- Changes to the data (design) do not require changes to programs // changes to programs do not require changes to data // the data can be accessed by any appropriate program

- Security is improved
- Each application only has access to the fields it needs // different users can be given different access rights

- Different users can be given different views of the data / data privacy is maintained
- So they do not see confidential information

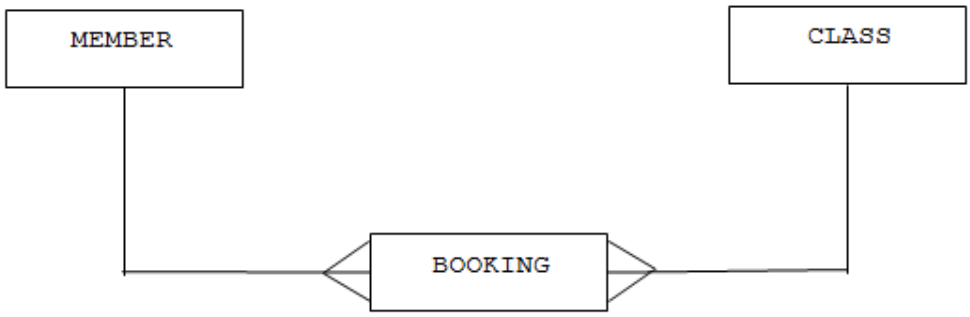
- Allows concurrent access
- Record locking prevents two users updating the same record at the same time // record locking assures data consistency

[6]

© UCLES 2016

GradeMax

(b) **ONE** mark for each correct relationship as shown.



[2]

(c) An example of a script is shown, but different syntax may be used.

```
CREATE TABLE CLASS (
    ClassID VARCHAR(5),
    Description VARCHAR(30),
    StartDate DATE,
    ClassTime TIME,
    NoOfSessions INT,
    AdultsOnly BIT,
    PRIMARY KEY(ClassID)
);
```

Mark as follows:

1 mark for `CREATE TABLE CLASS` and `();`
1 mark for `PRIMARY KEY(ClassID)`
1 mark for both `ClassID VARCHAR(5),` and `Description VARCHAR(30),`
1 mark for both `StartDate DATE,` and `ClassTime TIME,`
1 mark for `NoOfSessions INT,`
1 mark for `AdultsOnly BIT,`

[6]

© UCLES 2016