Cambridge IGCSE®

Computer Science

Revision Guide

David Watson and Helen Williams

CAMBRIDGE UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: education.cambridge.org

First published 2015

Printed in Poland by Opolgraf

A catalogue record for this publication is available from the British Library

ISBN 978-1-107-69634-1 Paperback

Example answers and all other end-of-chapter questions were written by the authors

® IGCSE is the registered trademark of Cambridge International Examinations.

Contents

Introduction

This book has been written as a form of revision for students sitting for the IGCSE or O Level Computer Science qualification. It has been assumed throughout that the student/reader has already studied the topics to the required level and will use this book as a form of examination preparation, although it should be pointed out that this revision book is also a stand-alone source of information.

To get a good understanding of Computer Science, a student should not just rely on one source of information; it should be a combination of two or more of the following.

- Textbooks
- Notes from teachers/lesson notes
- Student's own research from libraries/Internet sites
- Revision of past papers and mark schemes
- Student's own experiences
- Revision notes based on all of the above
- A revision textbook.

This revision book closely follows the Cambridge Computer Science syllabus and covers all of the topics therein. It also takes the subject slightly further to take into account potential syllabus revisions within the near future.

There are a number of revision questions at the end of each chapter which test the understanding of the student/reader. At the very end of the book, you will find two practice papers which will cover most of the topics in the 12 chapters of this book. The two question papers closely follow the new Computer Science syllabus where papers will be sat for the first time in 2015. Sample answers have also been supplied so that the student/reader can self-assess his/her performance accordingly. The sample answers have been written by the authors. Cambridge International Examinations bears no responsibility for these answers or for the comments offered.

Computer Science is an ever-changing subject and no single textbook could hope to cover all aspects of any given topic. The student/reader needs to keep up to date with developments and keep a note of these changes to see how they impact on the subject matter of this textbook.

Different types of question

A review of Computer Science and Computer Studies past papers over the last 20 years reveals a number of different question types.

The following is a list of the most common types of question you may possibly encounter.

Name

In these questions, one word answers would suffice.

For example, '*Name three devices used to input data into a computer.*'

Acceptable answers would be: *(1) Keyboard (2) Barcode reader (3) Microphone*

Describe

In these questions, you are expected to write a sentence to describe the computing term, feature, process, etc.

For example, '*Describe what is meant by the term interrupt.*'

Acceptable answer would be: *This is a signal from a device, such as a printer, sent to the CPU; the CPU will then temporarily stop what it is doing.*

Advantages and disadvantages

In these questions, you need to write a sentence or more to describe the advantages (benefits) and disadvantages (drawbacks) of some computer process. Usually, the question may involve some comparison with another computer process.

For example, '*Give one advantage and one disadvantage of using emails rather than the normal post to send a message.*'

Acceptable answer could include: *Advantage: emails are delivered almost immediately; Disadvantage: emails require investment in a computer system.*

Trace tables

In these type of questions, you are likely to be given a flowchart and asked to trace through the flowchart using some given data. You would be expected to show the value of the variables in the table at all stages of the process.

It is a good practice to draw a line across the table after any output; this makes it easier to go back and rectify any mistake you may make. There are several examples of this type of question in chapter 6, but basically they will look like this.

x	y	z	TOTAL	OUTPUT
1 2 3 4 5	3 7 1 6	9 14 1 36	1 2	2
1 2 etc.	5 11 etc.	25 132 etc.	1 etc.	 etc.

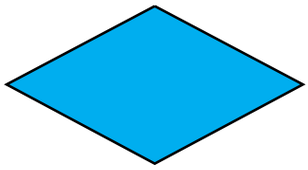
Flowcharts

In these questions, you would be given a computer process, such as '*using barcodes to carry out automatic stock control*'. The flowchart may have all the outline boxes but the actual items shown in these boxes might have been omitted. The items could be given in a numbered list and your task would be to place each stage in the appropriate box. You are well advised just to use the number of the item rather than trying to write the whole description in the given box.

Also remember the function of each flowchart box.

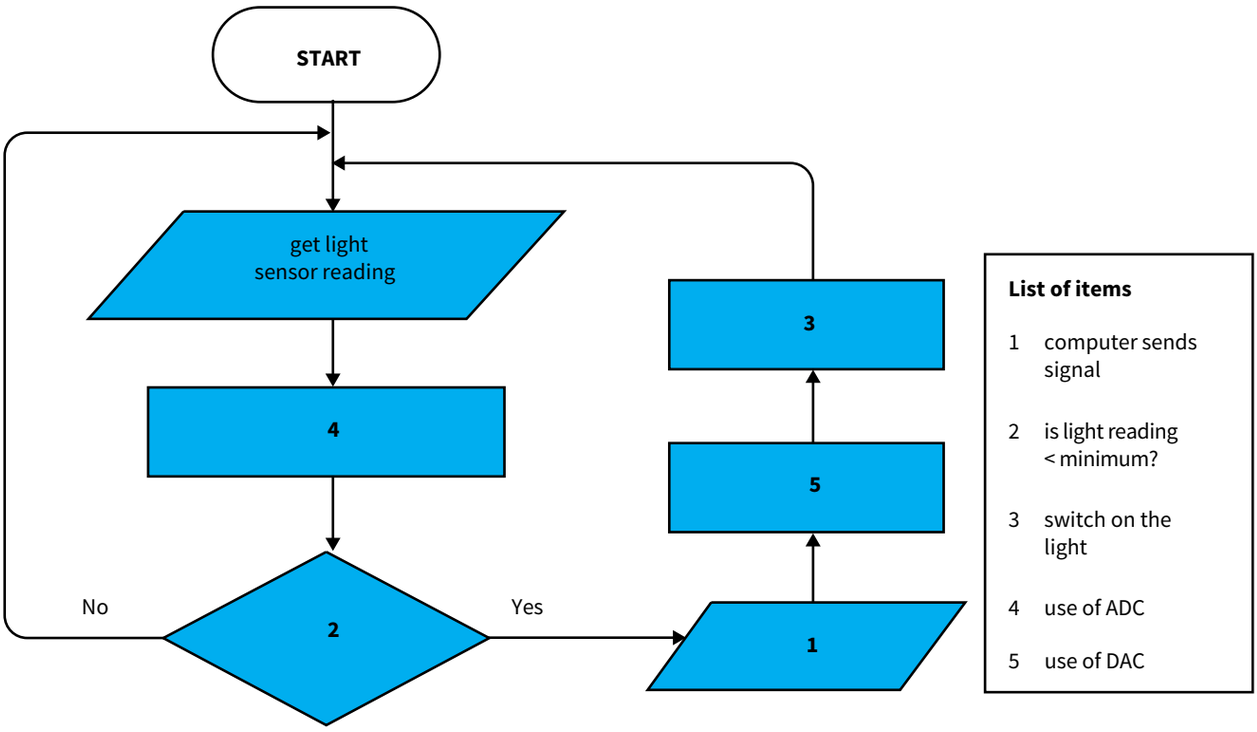
Instruction box – for example, *subtract 1 from the **total***

Input/output box – for example, *read data from the barcode*



Query box – for example, *have all barcodes been read?*

The following example illustrates this process.



Write an algorithm

In these questions, you are likely to be given a problem which needs to be solved using either a flowchart or pseudocode (whichever is easier).

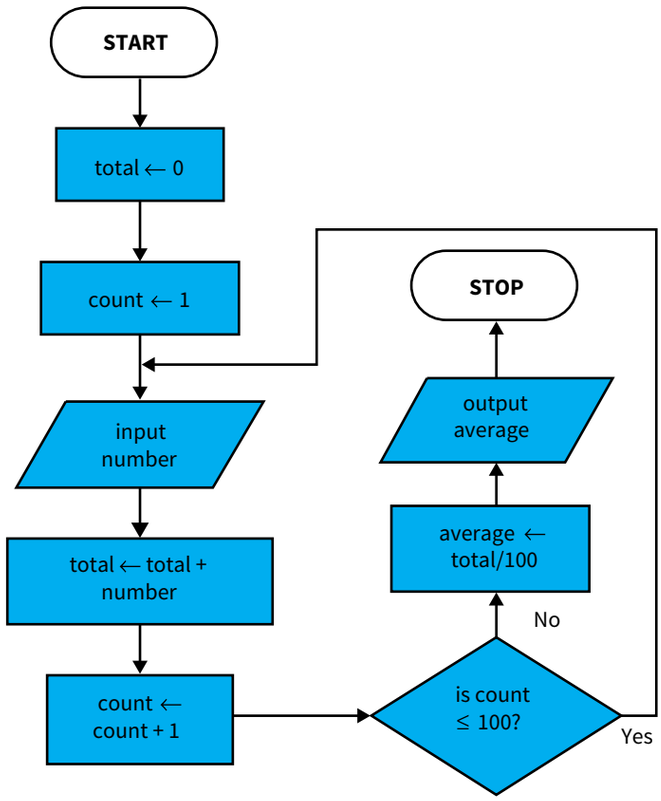
For example, '*Write an algorithm, using pseudocode or a flowchart, which inputs 100 numbers and outputs the average.*'

This can be done by pseudocode or by flowchart.

By pseudocode

```
total ← 0
for count ← 1 to 100
    input number
    total ← total + number
next count
average ← total/100
print average
```

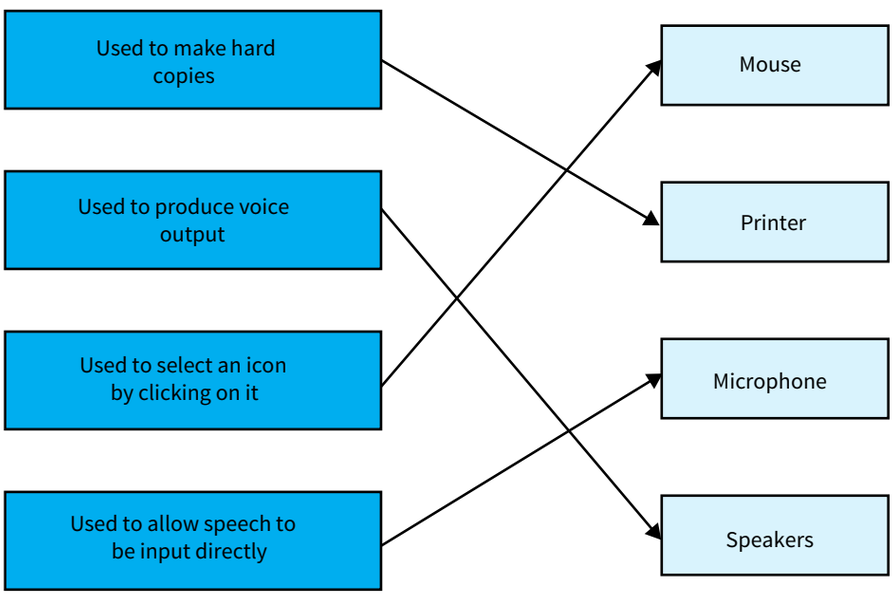
By flowchart



Linking boxes

In these questions, you would be given a description on one side and a computer term or application on the other side of the diagram.

You may be asked to draw arrows to show which description and computer term/application match. For example,



Truth tables

In these questions, you could be expected to complete a truth table to show the output from a logic gate (such as AND, OR, NOT, NAND, NOR or XOR) or from a logic network (which is a combination of these logic gates).

You may be given the binary inputs and it would only be necessary to complete the output part. For example,

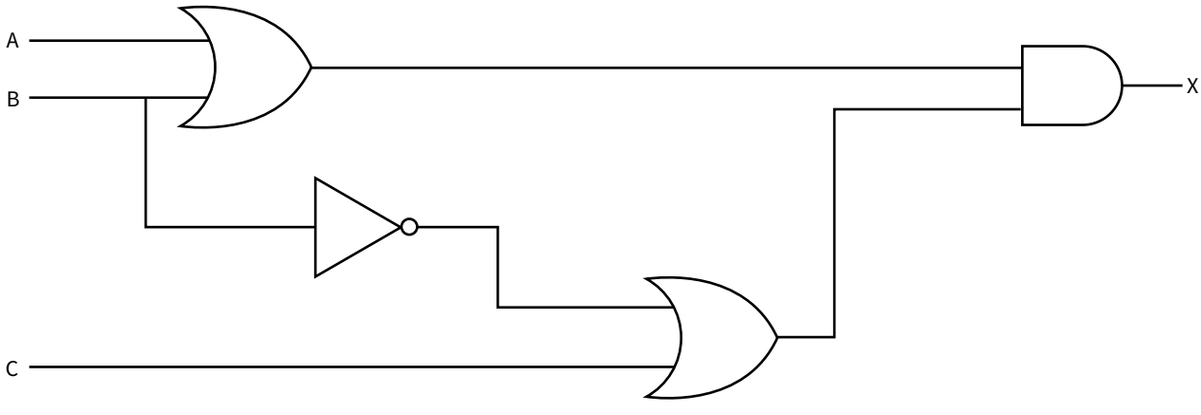
A	B	C	OUTPUT X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

When calculating the output for a logic network, use a pencil to show the output after each logic gate – you are less likely to make a mistake if you do this. For more examples, see chapter 9.

Drawing of logic networks

As with above, these are a recent addition to the type of questions which may be asked. You may be asked to draw a logic network based on either a given problem or a logic statement. A blank space would be provided to allow you to draw your logic network (again there are many examples of this type of question in chapter 9).

Example 1: '*Draw the logic network for* ***X = 1 if (A = 1 OR B = 1) AND (B = NOT 1 OR C = 1)***'



Example 2: '*Draw the logic network for the following problem.*'

An alarm sounds (X =1) if certain conditions occur. These are summarised as following.

Input	Binary value	Description
H	1	height >= 0 metres
	0	height < 0 metres
S	1	speed <= 100 km/hour
	0	speed > 100 km/hour
L	1	landing gear OK
	0	landing gear failed

X = if

landing gear OK and height < 0 metres OR

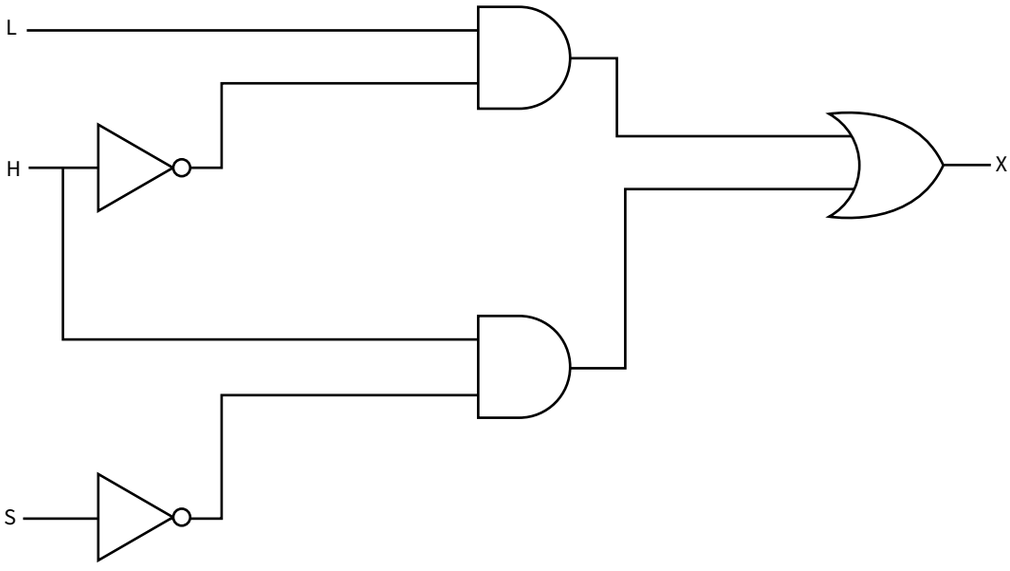
height >= 0 metres and speed > 100 km/hour

Possible solution:

Logic statement: X = 1 if [L = 1 AND H = NOT 1] OR

[S = NOT 1 AND H = 1]

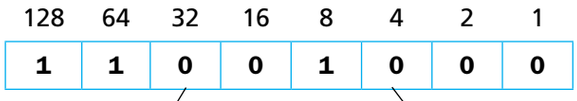
Giving the following logic network:



Miscellaneous types

There are many ***one-off*** question types set each year in Computer Science exams. Guidelines would always be given on how to set out your answers. The most common type seems to involve binary or hexadecimal registers in control applications such as elevators (lifts), robots, vending machines.

For example, the value 200 (base 10) would convert into binary as follows:



And into hex as:

i.e. C 8

Many more examples of the above question types can be found throughout the end of chapter exercises and in the two sample papers at the end of the book. When answering these questions, it is advisable to constantly refer back to this section to ensure you are correctly interpreting what is required.