MARK SCHEME for the October/November 2014 series

0420 COMPUTER STUDIES

0420/13	Paper 1, maximum raw mark 100

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 2014 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.
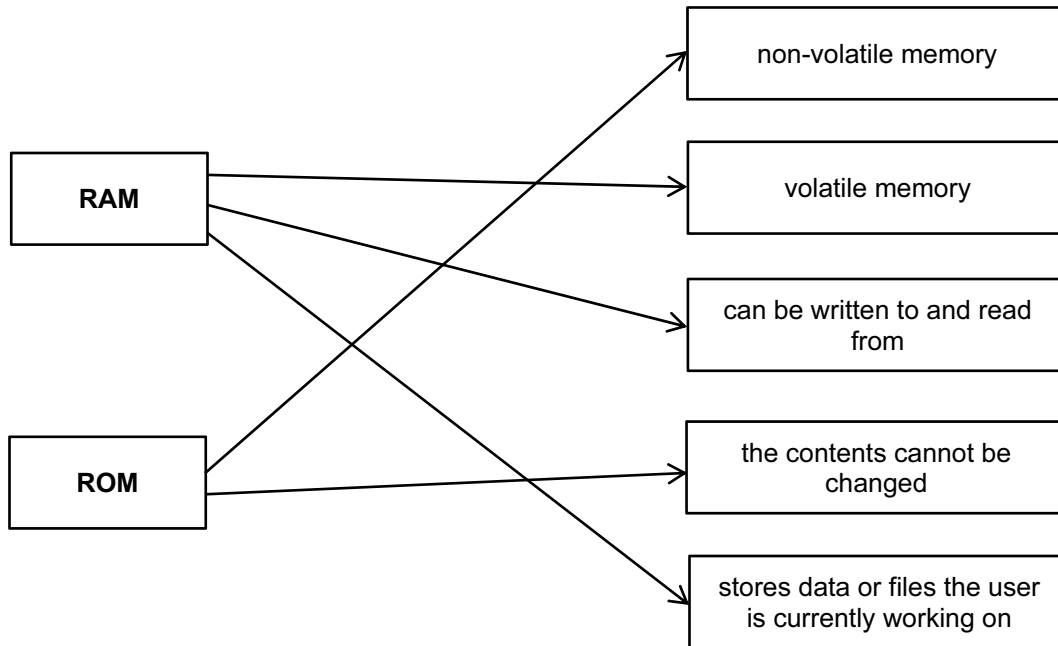
1   Any **three** from (need risk + reason):

Safety issues e.g.:
- electrocution from bare wires or spilling liquids on live equipment
- trip hazard due to trailing wires
- risk of heavy equipment falling from inadequate desks
- risk of fire if insufficient equipment ventilation or overloaded wall sockets

[3]

2

1 mark for each correct line (**max 5**)
Deduct 1 mark for each additional incorrect line.

[5]

3   (a)   **FALSE**   – encryption only stops data being read / making sense (but does not prevent the act of hacking)

   FALSE   – data when backed up could still have the virus attached to it
   – when the backed up data is re-loaded at a later date, the virus could be loaded again into the system together with the stored data

   TRUE   – tapping into a Wi-Fi network is relatively easy (even when the network is protected by passwords)

1 mark for each correct TRUE / FALSE with correct corresponding reason

[3]

(b) (i) drop down boxes help defeat spyware / key logging software **[1]**

(ii) – to ensure that it was in fact Felipe who logged on <u>last time</u> **[1]**
– an additional authentication check

(iii) in case it is not Felipe who attempts to access the account

[1]

4 (a) Any **three** from:

– Expert System asks questions
– Rikki inputs the answers to the questions
– next question asked is based on previous response(s)
– search <u>Knowledge Base</u> ……
– ….. uses the <u>Rules Base</u> to draw conclusions
– use of <u>Inference Engine</u>
– <u>probable</u> faults / solutions output on a screen
– … and the % probability of accuracy is also given

[3]

(b) Any **two** from:

– security purposes as these files are needed to access the Expert System
– licence agreement to only allow an authorised number of computers to access Expert System
– the extra files need to be accessed for the Expert System to work

[2]

(c) Any **one** from:

– chess games
– oil / mineral prospecting
– tax calculations
– weather forecasting
– identify plants / animals/chemical compounds
– careers guidance
– car engine tuning

[1]

5
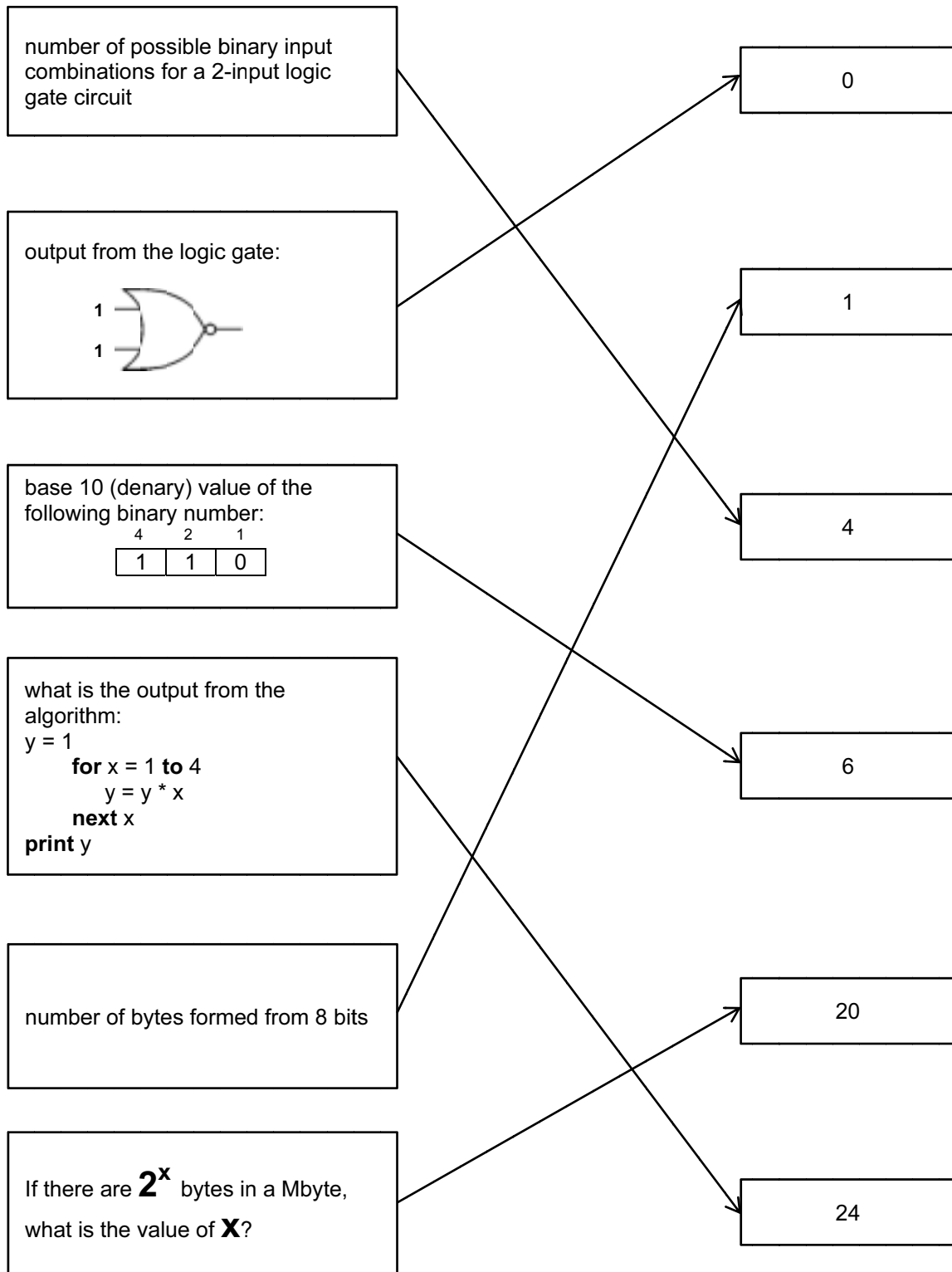
number of possible binary input combinations for a 2-input logic gate circuit

0

output from the logic gate:

1

1

1

base 10 (denary) value of the following binary number:

4	2	1
1	1	0

4

what is the output from the algorithm:
y = 1
    for x = 1 **to** 4
      y = y * x
    next x
print y

6

number of bytes formed from 8 bits

20

If there are 2^x bytes in a Mbyte, what is the value of **x**?

24

1 mark for each correct line **(max 5)**

[5]

6 1 mark for each error located with corresponding correction (description or corrected pseudocode acceptable)

    error: line 10: sum not initialised
    correction: sum = 0

    error: line 40: incorrect formula for sum
    correction: sum = sum + n

    error: line 50: incorrect IF statement
    correction: IF sum > 50 THEN ……

    error: lines 50 and 60: value of count causes a problem e.g. loop never ending
    correction: **either**   count = 19   on line 50
                      or   count = count + 1   between lines 30 and 40
                      or   any other correct solution

    error: line 80: output of n is incorrect
    correction   **output** sum   or   **print** sum

          [5]

7 (a) wikis   **[1]**

   (b) social networking sites   **[1]**

   (c) podcasts   **[1]**

   (d) tagging   **[1]**

   (e) blogs

          [1]

8 (a)

    `d u u l a f x   n a c c   x e   o y u o v .`

    <------------- 1 mark ------------->   <--------- 1 mark --------->

          [2]

   (b)

    `c o m p u t i n g   i s   f u n .`

    <------------- 1 mark ------------->   <--------- 1 mark --------->

          [2]

(c) Any **three** from:

- customers need a password / PIN
- use of card readers / use of Transaction Authentication Number (TAN)
- only certain characters from password / PIN requested…
- …the requested characters change each time user logs on
- card security code requested
- use of drop down boxes
- use of a customer reference number
- inform customer when they last logged on to the website
- use of <u>image</u> verification code e.g. CAPTCHA
- make reference to something unique to the customer e.g. their mobile phone number
- use of secure protocol e.g. https, padlock symbol

[3]

9 (a) 1 mark for input device + 1 mark for its <u>matching</u> use

input device:	touch screen
use:	select film / cinema seats / price

input device:	keyboard / keypad
use:	input number of tickets / card PIN

input device:	magnetic stripe reader / chip and PIN reader / card reader
use:	reading credit / debit card details

input device:	scanner
use:	to read any promotional vouchers (etc.)

[4]

(b) 1 mark for each different output device + 1 mark for its <u>matching</u> use

output device:	screen / monitor
use:	show films available / seating plan / prices of each seat / payment details

output device:	printer
use:	print receipt / tickets

output device:	loudspeaker / beeper
use:	to indicate error in input / confirmation of keys pressed

[4]

10 (a) 1 mark for each different sensor (**max 2**)

- pressure sensor
- example of sensor to detect if door closed / open e.g. magnetic field sensor, proximity sensor
- moisture / humidity sensor
- temperature / infrared sensor

[2]

(b) 1 mark for each item of data (**max 2**):

  – is the food frozen / uncooked / cooked?
  – <u>cooking</u> time
  – start / end time
  – power
  – weight
  – type of food
  – additional cooking feature e.g. browning

  1 mark for each corresponding input method (**max 2**):

  – turn dial to select option
  – touch screen / buttons / concept keyboard / keypad to select options
  – use of barcode readers (to read barcodes on food packaging which stores an automatic cooking programme)

[4]

(c) Any **four** from:
  – (microprocessor) checks / receives readings / data / signals from sensors….
  – …..continuously
  – if door open, (microprocessor) sends signal to sound alarm / stop process
  – (microprocessor) compares weight of food against stored values…
  – …and automatically sets cooking time / power
  – cooking time controlled by (microprocessor) comparing with stored values
  – (microprocessor) sends signal to beeper / notify when cooking program complete

[4]

11
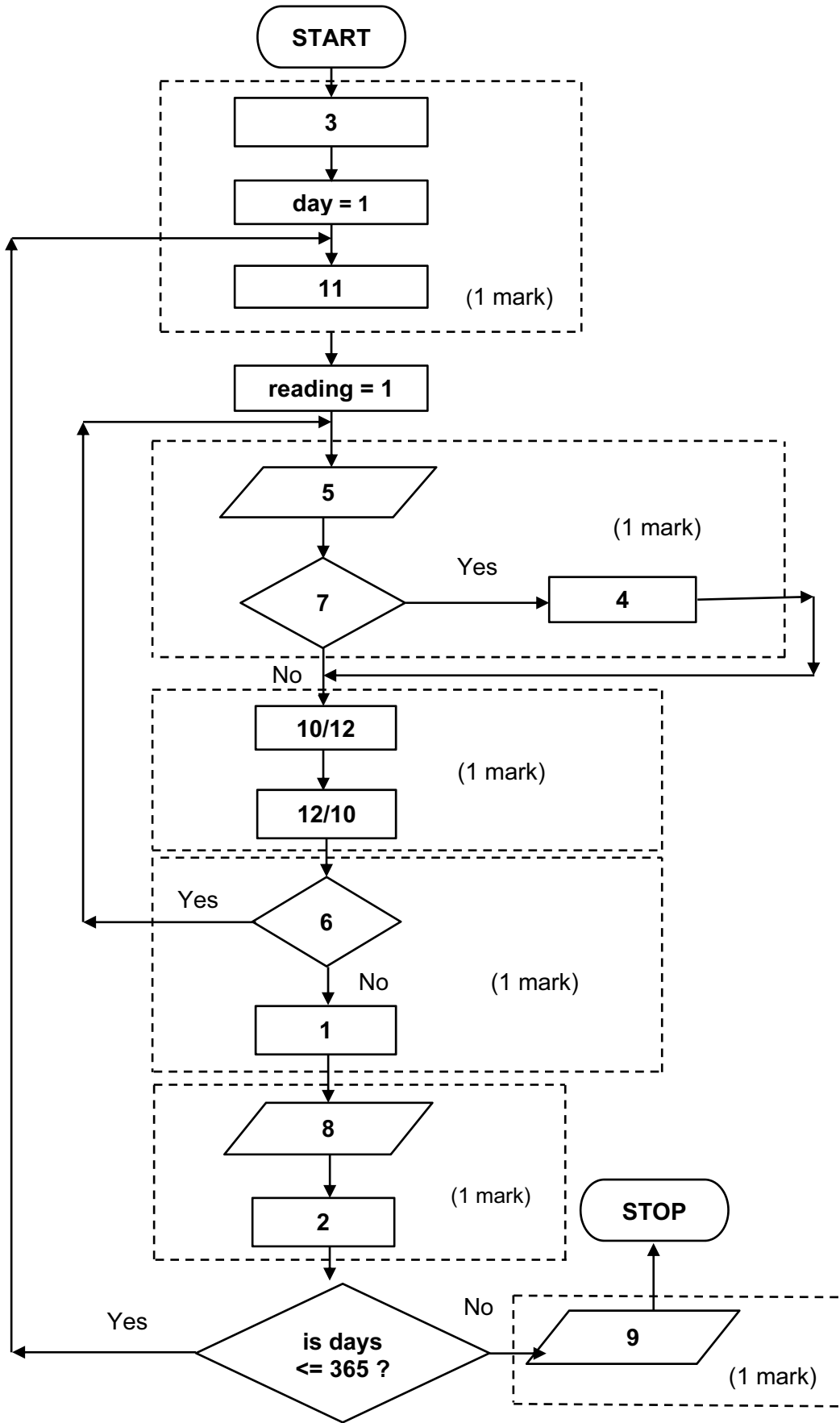
```
                          START

                            3

                          day = 1

                           11            (1 mark)

                        reading = 1

                            5
                                         (1 mark)
                                    Yes
                            7              4

                       No

                          10/12
                                         (1 mark)
                          12/10

               Yes
                            6

                       No        (1 mark)
                            1

                            8
                                         (1 mark)         STOP
                            2

        Yes                          No
                    is days                      9
                    <= 365 ?                              (1 mark)
```

[6]

12 (a) (i) B3 / B2   (1 mark)

   (ii) (B5 / C4) * 2 or = ((B5 * B2) / B3) * 2   (1 mark)

[2]

(b) = IF(C6>D1, "above", "equal or below")
         ←——→ ←———→
         1 mark 1 mark

OR

= IF(D1<C6, "above", "equal or below")
   ←——→ ←———→
   1 mark 1 mark

Alternative word(s) for "above" are acceptable.

[2]

(c) C4, C6, D6

[1]

13 (a)

T1	T2	T3	A	B	C	OUTPUT
0	0	0				
			3	2	1	
1						
	1		4	8	7	
2			6	0	3	
		1	5	6	9	
	2		4	11	3	
			0	0	0	
						2, 2, 1

  1 mark   1 mark   1 mark   < - - - - - - - - - - 1 mark - - - - - - - - - - - ->   1 mark

If no marks are awarded for the columns then 1 mark can be given for correct initialisation of T1, T2 & T3 as shown in the first row above.

[5]

(b) – any data set (except 0, 0, 0) where 2/3 of the numbers are the same e.g. 2, 8, 8
  – flowchart does not allow for numbers which have the same value

[2]

14 (a) Row number:

	1	2	3	4	5	6	7
Reg 1:	0	1	1	0	0	0	0
Reg 2:	1	0	0	1	0	0	1
Reg 3:	1	0	0	1	0	1	0
Reg 4:	1	0	0	1	1	0	0
Reg 5:	0	1	1	0	0	0	0

Reg 1 + Reg 5 = 1 mark
Reg 2 = 1 mark
Reg 3 = 1 mark
Reg 4 = 1 mark

[4]

(b)

	8
Reg 1:	0
Reg 2:	0
Reg 3:	1
Reg 4:	1
Reg 5:	0

Reg 2 + Reg 5 = 1 mark
Reg 3 = 1 mark
Reg 4 = 1 mark

[3]

15 (a)

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

1 mark

1 mark

1 mark

1 mark

[4]

[4]

(b) 1 mark for correct NOR gate and 1 mark for correct NAND gate in correct positions on left hand side of diagram.

[2]

(c) [A = **NOT** 1 **OR** B = **NOT** 1] 1 mark
AND 1 mark
[B = 1 **AND** C = **NOT** 1] 1 mark

Other notations which are acceptable:

(NOT A OR NOT B) AND (B AND NOT C)

$(\overline{A} + \overline{B}) \cdot (B \cdot \overline{C})$

(A' + B') . (B . C')

[3]

16 Marking points:

- initialise largest and two totals
- control loop for 1000 items
- check if price 1 > price 2 and increment total 1 (inside loop)
- check if price 2 > price 1 and increment total 2 (inside loop)
- calculate price difference (inside loop)
- method of dealing with negative difference
- check if calculated difference > largest difference and action taken if it is
- three outputs OUTSIDE a loop (calculation must have been attempted)

Sample program in pseudocode:

```
largest = 0
smarket1 = 0: smarket2 = 0                              1 mark
for item = 1 to 1000                                    1 mark
      input price1, price2
      if price1 > price2 then smarket1 = smarket1 + 1   1 mark
      if price2 > price1 then smarket2 = smarket2 + 1    1 mark
      difference = price1 – price2                       1 mark
          if difference < 0 then difference = - difference   1 mark
          if difference > largest then largest = difference  1 mark
next item
output smarket1, smarket2, largest                      1 mark
```

(max 6)

[6]