

Booklet 4

This covers the following topics found in section 4 of the computer studies syllabus:

<i>Item</i>	<i>Topic</i>	<i>Page numbers</i>
1	Automatic data capture	2
2	Validation techniques	3
3	Check digits	4
4	Verification techniques	5
5	Files	5
6	Binary data	5
7	Word processors	7
8	Desk top publishing (DTP)	8
9	Spreadsheets	8
10	Databases (DBMS)	11
11	Authoring software	12
12	“Off the shelf”/bespoke software	12
13	Macros	13
14	Problems based on section 4 topics	14

4.1 Data

Automatic Data Capture

There are many techniques available to collect data automatically without the need for manual entry. Manual entry is slower, more expensive (need to employ people to key in data) and also prone to error (e.g. typing in 4.1 instead of 1.4). The following is a list of available automatic data capture techniques:

Data logging	this technique involves collecting data automatically using sensors; often used when doing scientific experiments or monitoring a system (examples of sensors is given in booklet 5).
Barcode readers	these take data from printed barcodes and allow automatic stock control in, for example, supermarkets.
Radio Frequency Identification (RFID)	this method involves using small electronic devices containing a microchip and antenna; they work in a similar way to bar codes but can be read from a distance of 5 metres; often used to track live stock, vehicles, library books and goods sold in shops.
Biometrics	this involves obtaining data and identifying characteristics automatically in security systems e.g. use of finger prints, palm prints, facial images and iris prints.
Magnetic stripes	these contain information/data stored on magnetic material often on the back of a credit/debit card; the information is automatically read by swiping the magnetic stripe past a reading head.
Optical character recognition (OCR)	information on paper is automatically read by a scanner and is then analysed/processed by OCR software and stored in an electronic format.
Voice recognition	these systems recognise spoken words e.g. for disabled people who can't use keyboards where they speak commands instead of having to type.
Smart cards	these contain embedded microchips and receive power from the card readers; the microchip is made up of RAM, ROM and 16-bit processor and the stored data is automatically read by the card reader; used in credit cards, security cards, loyalty cards, etc.
Optical mark recognition (OMR)	OMR technology scans a printed form and reads pre-defined positions (where specific fields have been filled in e.g. ●—● or ■); the system records where marks have been made so can automatically determine responses to, for example, a questionnaire.

Data Collection

Data collection can be either automatic (see previous page) or manual. Manual techniques can involve:

- keyboards/keypads to type in data
- touch screens to select data/options

Validation Techniques

Validation is a process of checking if data satisfies certain criteria when input i.e. falls within accepted boundaries. Examples of validation techniques include:

Range check this checks whether data is within given/acceptable values e.g. checks if a person's age is > 0 but is also < 140 .

Length check this checks if the input data contains the required number of characters e.g. if a field needs 6 digits, then inputting 5 digits or 7 digits, for example, should be rejected.

Character check (type check) this checks that the input data doesn't contain invalid characters; e.g. a person's name shouldn't contain numbers.

Format check (picture check) this checks that data is in a specified format (template) e.g. date should be in the form dd/mm/yyyy.

Limit check this is similar to a range check except only ONE of the limits (boundaries) is checked e.g. input data must be > 10 .

Presence check check that data is actually present and not missed out e.g. in an electronic form, somebody's telephone number may be a required field.

Consistency check/
Cross field check this checks if fields correspond (tie up) with each other e.g. if **Mr.** has been typed into a field called "Title" then the "Gender" field must contain **M** or **Male**.

Check digit this is an extra digit added to a number which is calculated from the digits; the computer re-calculates and validates the check digit following input of the number (see next page).

(NOTE: check digits can identify 3 types of error:

- (1) if 2 digits have been inverted e.g. 23459 instead of 23549
- (2) an incorrect digit entered e.g. 23559 instead of 23549
- (3) a digit missed out all together e.g. 2359 instead of 23549)

This section shows how check digits are calculated. The ISBN-10 (used on books) has been chosen as the example; this uses a modulo 11 system which includes the letter **X** to represent the number 10.

Example 1

To calculate the check digit for the ISBN 0 - 2 0 1 - 5 3 0 8 2 - ?

- (i) the position of each digit is considered:

$$\begin{array}{cccccccccc} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & & \leftarrow \text{digit position} \\ 0 & - & 2 & 0 & 1 & - & 5 & 3 & 0 & 8 & 2 & - & ? & & \leftarrow \text{number} \end{array}$$

- (ii) each digit is then multiplied by its digit position and the totals added together

$$\begin{aligned} & 0 \times 10 + 2 \times 9 + 0 \times 8 + 1 \times 7 + 5 \times 6 + 3 \times 5 + 0 \times 4 + 8 \times 3 + 2 \times 2 \\ & = 0 + 18 + 0 + 7 + 30 + 15 + 0 + 24 + 4 \\ & = 98 \end{aligned}$$

- (iii) the total is then divided by 11 (modulo – 11) and the remainder, if any, is subtracted from 11 to give the check digit.

$$\begin{aligned} 98 \div 11 &= 8 \text{ remainder } \mathbf{10} \\ 11 - \mathbf{10} &= \mathbf{1} \end{aligned}$$

this gives a check digit of **1**
final ISBN becomes 0 - 2 0 1 - 5 3 0 8 2 - **1**

Example 2

To check the correctness of a check digit the computer re-calculates it as follows:

The ISBN to check is: 0 - 1 3 1 5 - 2 4 4 7 - X

- (i) the position of each digit is considered:

$$\begin{array}{cccccccccc} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & & \leftarrow \text{digit position} \\ 0 & - & 1 & 3 & 1 & - & 5 & 2 & 4 & 4 & 7 & - & X & & \leftarrow \text{number} \end{array}$$

- (ii) each digit is then multiplied by its digit position and the totals added together

$$\begin{aligned} & 0 \times 10 + 1 \times 9 + 3 \times 8 + 1 \times 7 + 5 \times 6 + 2 \times 5 + 4 \times 4 + 4 \times 3 + 7 \times 2 + X \times 1 \\ & = 0 + 9 + 24 + 7 + 30 + 10 + 16 + 12 + 14 + 10 \quad (\text{recall that } X = 10) \\ & = \mathbf{132} \end{aligned}$$

- (iii) the total is then divided by 11; if there is no remainder then the check digit is correct:

$$\begin{aligned} 132 \div 11 &= 12 \text{ remainder } \mathbf{0} \\ \text{hence the check digit is correct} \end{aligned}$$

Verification

Verification is a way of preventing errors when data is copied from one medium to another (e.g. from paper to disk/CD, from memory to DVD, etc.). There are a number of ways in which verification can be done:

Double entry in this method, data is entered twice (using two different people); the data is only accepted if both versions match up. Often used to verify passwords by asking them to be typed in again by the same person.

Visual check this is checking for errors by comparing entered data with the original document (NOTE: this is **not** the same as proof reading!!).

Parity check this is used to check data following potential transmission errors; an extra bit is added to each binary number before transmission – e.g. EVEN parity makes sure each number has an **even** number of 1 – bits;

Example: if 11000110 is sent (four 1's)
and 11100110 is received (five 1's)

then the system will know an error has occurred.

4.2 Files

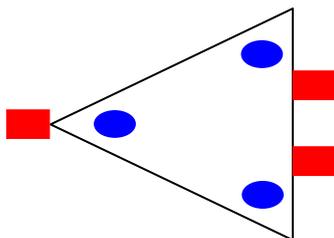
File maintenance is important. Updating of files usually involves amending, inserting and deleting data.

Example A bank would **amend** data if a customer changed their personal details (e.g. telephone number, address or their name through marriage). Data would need to be **inserted** if a new customer joined the bank and **deletion** of data would occur if a customer left the country or died (in both cases their accounts would be closed).

4.3 Binary Data

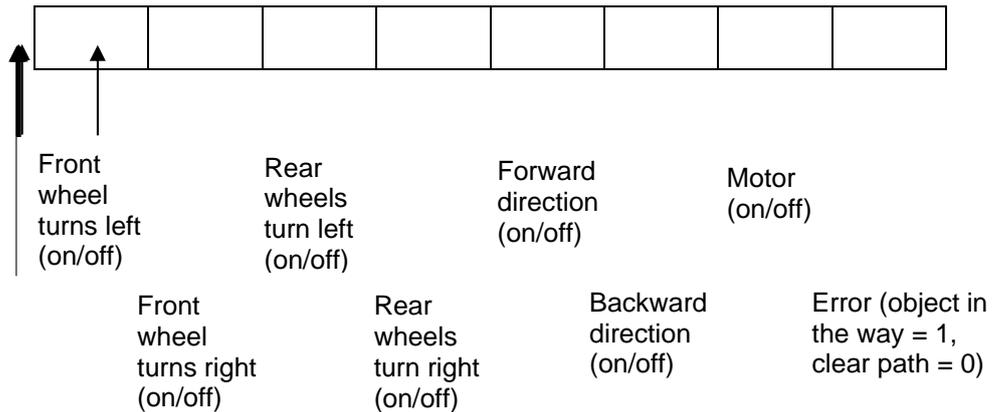
Controlled devices usually contain registers which are made up of binary digits (bits). The following example shows how these registers can be used to control a device.

Example



The device on the left is a mobile trolley with 3 wheels. All three wheels can turn left or right and each wheel has its own electric driving motor. Sensors at the front and rear of the trolley detect an object in its path which would subsequently stop all movement. An 8-bit register is used to control the device.

-  these represent the three wheels
-  these represent the three sensors



(1 = ON and 0 = OFF)

Therefore, **1 0 1 0 1 0 1 0**
 means: front wheel turns left
 back wheels turn left
 the direction is forward
 the motors are on

therefore the trolley is moving forward and turning left.

(i) What does this register mean?

0 0 0 1 0 1 1 1 ?

(ii) How would the following be represented using the above register?
 - front wheel turning right
 - back wheels turning left
 - moving in a forward direction
 - motors on
 - no object in its path

Answers

(i) - front wheel **not** turning left or right
 - rear wheels turning right
 - going in backward direction
 - motors on
 - error – object in path

So the vehicle is going nowhere.

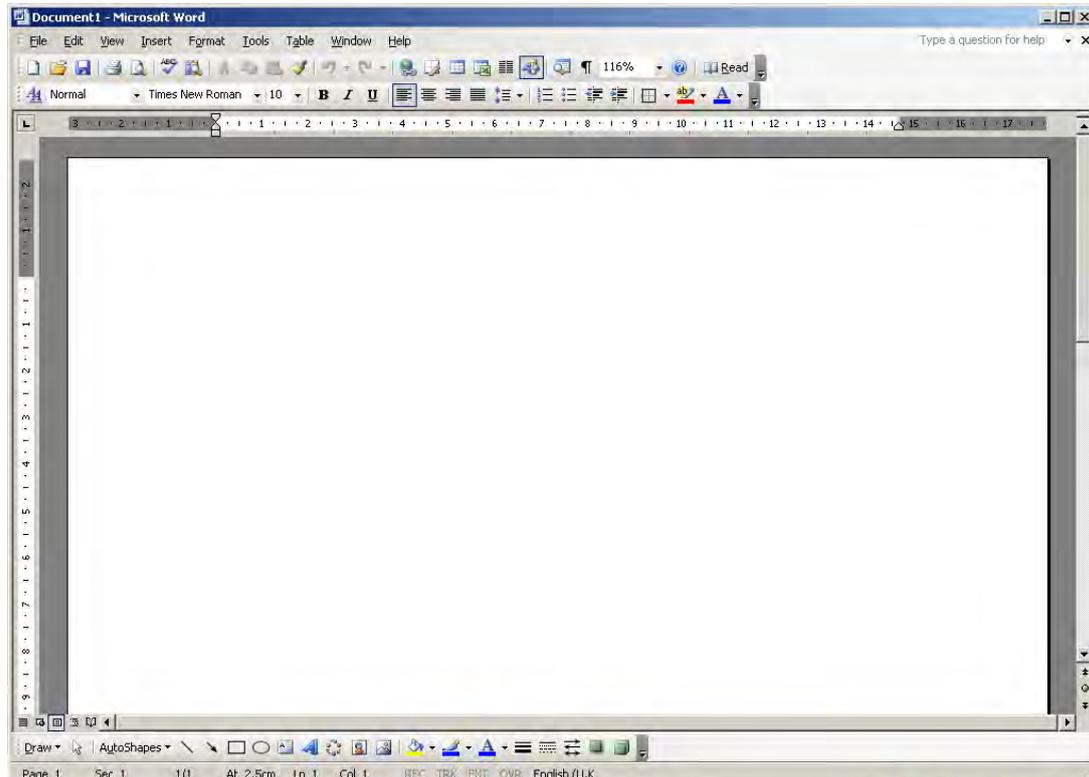
(ii) **0 1 1 0 1 0 1 0**

4.4 Software

Some of the key features of software are considered in this section. Familiarisation with software packages is usually best achieved by carrying out a variety of tasks on the computer.

Word processors

Word processors now have many features over and above the original task of generating a typed document. This section will concentrate purely on word processing features.



Sample of a word processing page showing icons and functions available

Main features available in all word processors:

- ability to set the page size and page orientation
- ability to change font style and font size (e.g. *arial*, *courier*, *times new roman*, etc.)
- ability to change margins (top, bottom and sides)
- page format (e.g. paragraphs, columns, etc.)
- importing text, graphics and pictures into the document body
- inserting and deleting text
- cut and paste/copy and paste (to prevent the need to re-type sections of text or to move/remove text)
- search and replace based on key words
- word wrap (e.g. auto adjust a line to fit into paragraph alignment such as straight margins on left and right, etc.)
- file management (create/delete/move/search for files)
- headers and footers and the automatic numbering of pages
- graphics (embed drawings and graphs into the document)

- macros (*see later notes*)
- merge (merge text files from one file into another e.g. mail merge)
- ability to send documents to a printer
- carry out a word count on the document
- add and manipulate tables in the main body of the document
- windows (edit 2 or more documents at the same time)
- WYSIWYG (*what you see is what you get*) – preview document to see how it will look when printed
- spell checkers, language check and thesaurus
- simple translation from one language to another; for example:

This is an example of a document translated from English to French

C'est un exemple d'un document traduit d'anglais-français

Desk Top Publishing (DTP)

As word processors develop increasingly more sophisticated features, the differences with desk top publishing (DTP) packages becomes more and more blurred.

Essentially in DTP, a user creates a page layout with text, graphics, photos and other elements. The following is a list of key differences to word processors:

- most word processors force users to work on a document in a linear fashion (i.e. paragraph by paragraph and page by page); with DTP it is possible to place elements on the page and then move them around as you wish to create the desired product.
- elements repeated throughout a document (e.g. a company logo) can be placed in *master pages* rather than setting up each page individually as would be the case with a word processor.
- text is usually imported from a word processor or is typed into the document directly using a *frame*. This allows you to create the layout first, link the frames together and add the text into the *frames* later (e.g. in newsletters and magazines where layouts are created before any of the actual text becomes available).

Spreadsheets

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						
8						

The following is a brief summary of the common features found in spreadsheets:

- it is made up of columns and rows; each row is identified by a number and each column is identified with a letter (or letters)
- each cell is identified by its column id and row id e.g. A4, ET300 etc.
- a cell may contain a *label* (text entry such as MEAN VALUE) or a *value* (which can be a number, a date, a formula, result of a calculation, or currency)
- the cursor identifies the current cell selected; it is possible to click on to a cell and a “+” sign appears in the bottom right hand corner; by dragging the formula down it is possible to *replicate* it in other cells

For example:

	A	B	C
1	15	3	=(A1 * B1)
2	11	5	
3	7	7	
4	9	9	
5	10	11	↓

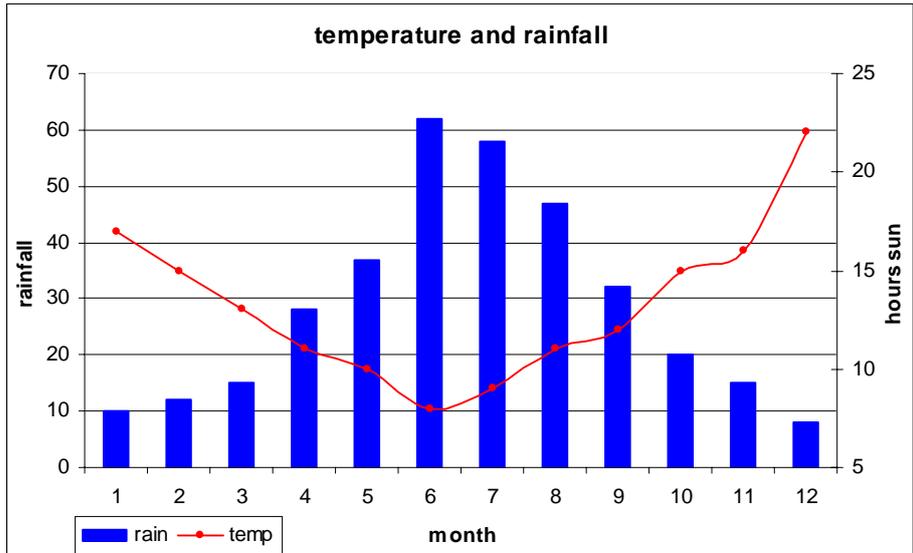
By selecting cell C1 and dragging the formula down to C5 the following formulae will be replicated in C2 to C5: (A2*B2), (A3*B3), (A4*B4) and (A5*B5)

- formulas are used to represent a mathematical function
e.g. = (C5 * D5) * 100
= A5 + B5 + C5 + D5
- functions allow computations to take place
e.g. SUM, AVERAGE, MAX, MIN
e.g. the second formula above could be written as = SUM(A5:D5)
- it is also possible to use mathematical/logical expressions within cells
e.g. IF G5 > 100 THEN “Y” ELSE “N”
- it is also possible to create charts and graphs from spreadsheets (e.g. using a *chart wizard*). For example:

	A	B
1	rainfall	temperature
2	10	17
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13	8	22
14		

Cells A1 to A13 are highlighted and the first chart is created. Then cells B1 to B13 are selected and the second chart created.

Other aspects such as colours, labels, etc. are then added.



- now consider the following spreadsheet:

	A	B	C	D
1	vehicle id	kilometres	litres	kilometres/litre
2	A	150	12.5	12.0
3	B	200	27	7.4
4	C	300	20	15.0
5	D	250	15	16.7
6	E	100	10	10.0
7	F	400	33	12.1
8			average km/l	12.2
9			maximum km/l	16.7

formula in cell D2 is = (B2/C2)

replicating this formula (by highlighting then drag down to D7) gives all the formulas needed in cells D3 to D7

formula in D8 (to find average) can be done in 3 ways:

= AVERAGE(C2:C7) or
 = SUM(C2:C7)/6 or
 =(C2+C3+C4+C5+C6+C7)/6

formula in D9 to find maximum value in column is = MAX(C2:C7)

- problems associated with spreadsheets:
 - capacity (i.e. maximum number of rows and columns) can be limiting when dealing with large data sets
 - security issues in some spreadsheets (i.e. if you can open a spreadsheet you can also change any part of it; this can lead to fraud)
 - lack of concurrency; usually only one person can work on a spreadsheet at any given time (databases are better in this aspect since they can usually be opened by more than one user at a time)
 - because of their structure it is easy to enter an error (whether on purpose or by mistake) by entering data in the wrong cells or dependencies among cells incorrectly applied

Databases (Database Management Systems – DBMS)

Databases are used to collect and organise information. Most database have all of the following properties:

Tables

Data is stored in rows and columns (much like a spreadsheet – the main difference is how the data is organised). Each row in a table is called a *record* which is made up of a number of *fields* (columns in the table). The data type in the fields is usually either text, numeric or date/time. Most databases contain a number of tables which are usually linked together in some way.

Forms

Forms are commonly called *data entry screens* since they are the user interface which allows data in the tables to be viewed, entered or edited. Forms permit the control of how other users interact with the information in the database e.g. only allow certain fields to be seen or only allow certain operations to be carried out. This subsequently helps the protection of the information and also ensures it is entered correctly.

Reports

Reports are produced as a result of questions such as “how many times has a type of car broken down” or “which cities have the highest house prices”. Reports can be printed out, viewed on the screen or exported (e.g. to a word processor, an email message, etc.).

Queries

Queries most commonly allow information to be retrieved from tables. Since the information is often spread across several tables, queries allow it to be viewed as one single *datasheet*. They also allow filtering so only the records required are seen. Queries can be either results seen directly on the screen or the output to another form or report. Example of a query: (house > 200 000) OR (no_of_rooms < 4).

Sample database/table from a database

Name of substance	Toxic	Smallest size (microns)	Largest size (microns)	Filtered out by 10 micron mesh?
cement dust	no	3	100	some
coal dust	no	1	100	some
common sand	no	10	2000	yes
paint pigments	yes	0.01	5	no
talcum powder	no	3	80	some
pollen	no	10	100	yes
smog	no	0.01	1	no
viruses	yes	0.002	0.04	no
bacteria	yes	0.2	20	some
human hair	no	20	300	yes

- Points to note:
- (1) there are 10 records in this section of the database
 - (2) each record has 4 fields
 - (3) sample queries:

(Smallest size (microns) < 1) OR (Toxic = “yes”)
would output records 4, 7, 8 and 9

(Largest size (microns) > 99) AND (Toxic = “no”)
Would output records 1, 2, 3, 6 and 10

The query should match up with the field titles to ensure a correct search is carried out. Search engines on the Internet work in a similar way; the clever part is how the information is stored on the databases so that the time to do the search (based on key words) and the effectiveness of the search itself results in a very powerful and very useful tool.

Authoring Software

Also known as *authorware*; this is a program that helps in the writing of *hypertext* or *multimedia* applications. *Authoring tools* usually allow the creation of the actual required application by linking together *objects* which can be text, graphics, music file, etc. By defining the relationship between *objects* it is possible to produce very useful and eye catching applications.

Many authoring packages support a scripting language to allow for even more sophisticated applications to be produced. Scripting languages have *pre-built functionality* (such as menu styles, graphic displays, etc.) that is embedded within HTML. These *pre-built functions* allow users to engage with a web-site in real time.

The distinction between authoring tools and programming tools is not at first obvious. However, *authoring tools* require less technical knowledge to master and are usually used for applications that present a mixture of text, graphics, and audio/music files.

Bespoke versus off-the-shelf software

Usually software packages are bought off-the-shelf. However, certain applications (usually for commercial businesses, industrial companies, etc.) require specially written software which is unique to that customer and is known as *bespoke software*.

There are many advantages and disadvantages of both systems.

Off-the-shelf – advantages

- tends to be less expensive since development costs can be spread over the many users
- can be more sophisticated since large sales bring in considerable revenue which can be used to enable a lot of development to be done.
- there will be considerable support from other user groups
- sharing of files with other packages is usually easier since they are likely to be compatible (e.g. *Excel* and *Word*)
- the package is available immediately – no need to wait for development period
- tried and trusted software since usually highly developed due to considerable user feedback

Off-the-shelf – disadvantages

- can be over-complex as it tries to cover as many aspects as possible (e.g. most users of *Word* only utilise about 10% of its features)
- the software is often a compromise since it is intended for many users all with different requirements
- may be very complex to learn because of the complexity caused through having unwanted features
- may not fit exactly into the company requirements causing a compromise in the way it has to be used within the organisation
- if there are major concerns with a software problem it is unlikely that it will be fixed immediately (since it may cause problems with other users and/or cause software instability)

Bespoke Software – advantages

- specifically designed for the application and therefore more efficient and will only contain the features wanted; since it will be devoid of unwanted features the software can take its specific requirements to new levels which will make it very powerful
- can be customised to interface with all other software within the company thus preventing software clashes
- easier to use since very specific and the writers of the software can also be involved in the training of staff which will be geared towards their requirements
- can be modified/updated as the company's requirements change
- much better customer support since they will be in direct contact with the software designers

Bespoke Software – disadvantages

- very dependent on the suppliers of the software; if they go out of business there will be little or no support if problems occur
- less likely to be as well developed/tested as off-the-shelf software
- much more expensive since all the development costs need to be met by the company (but efficiency savings may compensate for this and it may prove to be less expensive in the long run)
- the development time can be very long which may cause problems
- difficult to tell how good the final software package will be (could be a real problem if the company have waited 2 to 3 years to receive their new software only to find it doesn't meet their needs!!)

Macros

A macro is a name or key that represents a series of commands or key strokes. Many applications allow single word or single nominated key on a keyboard to perform a whole series of actions. Macros can be either written as required and then stored or stored in a library for future use.

For example, suppose you want some data to be input into a spreadsheet if the result of a calculation in cell K40 is negative:

```
move to cell K40  
is result of calculation < 0  
if < 0 then load file "error report"  
if >= 0 then output value
```

Macros can be very sophisticated and save the operator a lot of time. For example, the name and address in a word processor could be set up by pressing key **F1** and the system brings information from a related database.

Other software

Other software, such as web browsers and search engines, communications software (including emails) and graphics interfaces are all covered in other booklets and the information won't be duplicated here.

4.5 Problems

A list of examination-based problems now follows. These can be used as revision questions or centres can use the questions to make up their own assessments to give candidates practice under examination conditions.

(1) Describe **four** methods used for automatic data capture. Give a different application for **each** of the methods chosen.

(2) A record contains the following fields:

- Title (Mr/Mrs/Miss) and Sex (Male or Female)
- Date of birth (written in the form dd/mm/yyyy)
- Pay Number (which is made up of 8 digits)
- Telephone Number
- Pay

Give a **different** validation check for **each** of the above fields and explain why it is used.

(3) A system uses the ISBN-10 (modulo 11) method to generate and validate bar codes:

- each digit in the number is multiplied by its digit position
- the result of each multiplication is added to a total
- the total is then divided by 11
- the remainder is subtracted from 11 to give the check digit (if the remainder is 10, the letter "X" is used as the check digit)

(a) Which of the following numbers have the correct check digit?

(i) 0 – 5 5 5 – 2 1 6 2 2 – 5

(ii) 0 – 1 2 1 – 9 0 0 2 1 – X

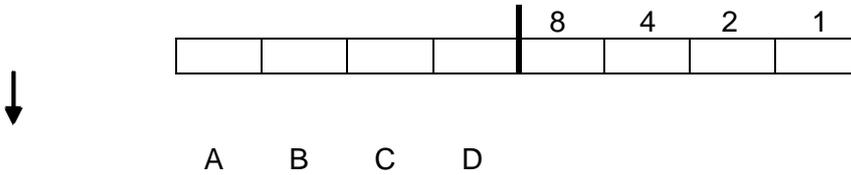
(b) Calculate the check digit for the following number:

0 – 1 5 0 – 2 4 6 2 2 –

(c) Discuss the types of errors that check digits can identify.

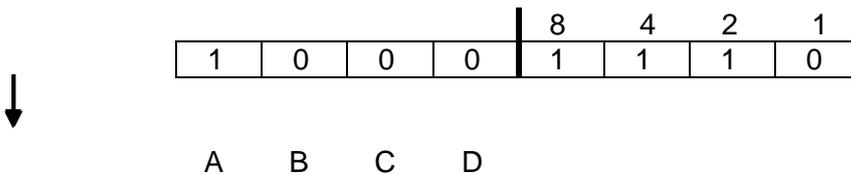
- (4) (a) Describe **three** ways of carrying out verification checks.
 (b) Give **two** examples where verification checks would be made.

(5) A burglar alarm system uses an 8 – bit register. The first 4 bits indicate if a sensor has picked up an intruder (shown with a **1 – value**) and the second four bits indicate the zone where the sensor has picked up an intruder:



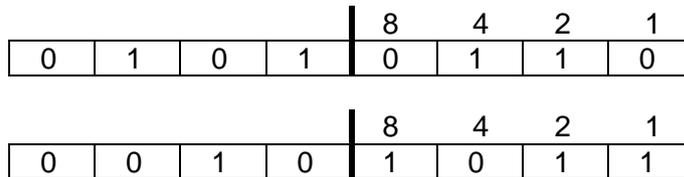
(where A = window sensors
 B = floor sensors
 C = door sensors
 D = infra red heat sensors)

For example:



Indicates: *window sensor detected an intruder in zone (8 + 4 + 2) 14*

(a) What do the following indicate:



(b) What would be the binary pattern for a window broken and an infra red sensor detecting an intruder both in zone 15?

(6) A company is using a word processor to produce a page advertising their products. Unfortunately, the information wouldn't fit onto one single page.

Which **five** features of a word processor could be used to ensure that the advert fits onto one page only.

(7) A college has invested in a computer system to help carry out the following tasks:

- advertise their college (both on the Internet and by paper leaflets)
- keep a track of student and staff information/contact details
- keep a track of payments and fees
- produce on line courses for the students

Describe **four** software packages the college could use to carry out the above tasks and explain why each software package has been chosen.

(8) Describe **three** fundamental differences between word processors and desk top publishing packages.

(9) The following spreadsheet has been set up to compare the depreciation of 8 new cars over a 4 year period.

	A	B	C	D	E	F	G
1	name of car	new price (\$)	Value after year 1 (\$)	Value after year 2 (\$)	Value after year 3 (\$)	Value after year 4 (\$)	Depreciation over 4 years (%)
2	Fiat Bravo 150	28 000	21 000	15 600	11 200	9 800	65
3	Honda Civic "R"	36 000	28 800	23 400	20 000	18 000	50
4	Range Rover SE	80 000	64 000	60 000	50 000	48 000	40
5	Mitsubishi Lancer	20 000	12 000	9 000	6 400	5 000	75
6	Citroen C8 LX	40 000	20 000	13 200	10 000	8 000	80
7	BMW 320 Conv.	72 000	57 600	50 000	43 200	36 000	50
8	Audi A3 1.6 SE	30 000	22 500	18 000	15 000	12 000	60
9	Mercedes SLK 350	68 000	57 800	51 000	48 000	45 600	33
10					Mean	Deprecn	56.6
11					Highest	Deprecn.	80.0

- (a) What formula must be in cell G4 to find the depreciation on a Range Rover after 4 years?
- (b) What formula must be in cell G10 to calculate the mean (average) depreciation of all 8 cars?
- (c) What formula must be in cell G11 to find the highest (maximum) depreciation of all 8 cars?
- (d) Describe how it would be possible to predict the value of each car after 5 years and after 10 years?

- (10) A database has been set up to compare certain features of freezers. A section of the database is shown below.

Item I.d.	Price (\$)	Lowest temp ($^{\circ}\text{C}$)	Capacity (m^3)	Warranty (years)	Colour
A	300	- 20	0.16	1	white
B	250	- 18	0.15	1	silver
C	400	- 22	0.25	2	white
D	300	- 20	0.18	3	white
E	520	- 40	0.35	3	silver
F	360	- 25	0.30	1	white
G	410	- 30	0.30	2	white
H	290	- 18	0.20	2	silver

- (a) How many records are shown in this section of the database?
- (b) What items would be output if the following search condition was input:
(Price(\$) > 300 **OR Warranty(years)** > 1) **AND (Colour = "silver")** ?
- (c) Give the search condition to find all freezers which operate below -25°C and have a capacity of more than 0.25 m^3 .