

Computer Studies

1st Edition

Sample Answers

(including questions for Theory - section 3)

A Complete Course
for Matsec
at Ordinary Level

First Published in 2001
Published by Merlin Library Ltd, Blata l-Bajda
Copyright 2001 St Martins Institute of Information Technology

email: mail@merlinlibrary.com
website: <http://www.merlinlibrary.com>

The authors have used their best efforts in preparing these answers. These efforts include development and research of the material. The answers are not to be taken as model answers but more as sample answers. Any recommendations will be appreciated via email to mail@merlinlibrary.com with the text COMPUTER STUDIES in the subject field.

This book may not be reproduced, stored in a retrieval system or transmitted, in whole or in part, including illustrations, in any form or by any means, electronic, mechanical, photocopying recording or otherwise, without prior written permission from the author and from the publishers.

Contents

<i>Pascal</i>	181
---------------------	-----

Theory

Section 1

1.0	<i>Introduction To Computers</i>	24
2.0	<i>Computer Hardware – A General Overview</i>	29
3.0	<i>Boolean Logic Gates</i>	35
4.0	<i>Peripherals – Input and Output Devices</i>	43
5.0	<i>Peripherals - Secondary Storage Devices</i>	48

Section 2

1.0	<i>Software</i>	53
2.0	<i>Application Software</i>	54
3.0	<i>Programming Languages</i>	59
4.0	<i>Operating Systems And Networks</i>	60
5.0	<i>Files And File Access</i>	62

Section 3

1.0	<i>Systems Analysis And Design</i>	64
2.0	<i>Documentation</i>	68
3.0	<i>Errors And Validation Techniques</i>	71
4.0	<i>Designing Algorithms And Flowcharts</i>	74
5.0	<i>Number Systems</i>	77
6.0	<i>The EDP Department</i>	82

Appendix

<i>Section 3 Questions</i>	86
----------------------------------	----

Pascal

Chapter 1

Exercise 1

```
program chapter1_exercise1;

uses crt;

begin
  clrscr;
  writeln ( 'Name':6, 'Age':7, 'Telephone':13 );
  writeln;
  writeln ( 'John':6, 76:7, '455287':13 );
  writeln ( 'Ann':6, 2:7, '0994-4552':13 );
  writeln ( 'Andrew':6, 100:7, '986332':13 );
  readln;
end.
```

Exercise 2

```
program chapter1_exercise2;

uses crt;

begin
  clrscr;
  writeln ( '1 + 5 + 45 / 87 = ', 1+5+45/87:7:4 );
  readln;
end.
```

Exercise 3

```
program chapter1_exercise3;

uses crt;

begin
  textcolor ( white + blink );
  textbackground ( red );
  clrscr;
  gotoxy ( 30, 12 );
  write ( '...wait a moment...' );
  readln;
end.
```

Chapter 3

Exercise 4

```
program chapter3_exercise4;

uses crt;

var
  chick : byte;
  proverb : string;

begin
  clrscr;
  chick := 1;
  writeln ( chick );
  chick := 2;
  writeln ( chick );
  chick := 3;
  writeln ( chick );
  proverb := 'Do not count your chicks before they are hatched!';
  write ( proverb );
  readln;
end.
```

Chapter 4

Exercise 5

```
program chapter4_exercise5;

uses crt;

var
  age : byte;

begin
  clrscr;
  writeln ( 'Enter your age:' );
  readln ( age );
  clrscr;
  age := age + 1;
  writeln ( 'Your age next year will be ', age );
  readln;
end.
```

Exercise 6

```
program chapter4_exercise6;

uses crt;

var
  name, item : string[25];

begin
  clrscr;
  writeln ( 'Enter your name:' );
  readln ( name );
  writeln ( 'Enter the name of the item you need:' );
  readln ( item );
  clrscr;
  writeln ( 'Dear ', name, ',' );
  writeln;
  writeln ( 'Thank you for your interest in our products.' );
  writeln ( 'I regret to inform you that we have run out of ',
                                                    item );

  writeln;
  writeln ( 'Yours truly,' );
  writeln ( 'John Fenech' );
  readln;
end.
```

Chapter 5

Exercise 7

```
program chapter5_exercise7;

uses crt;

var
  num1, num2, num3 : integer;

begin
  clrscr;
  writeln ( 'Enter the 3 numbers, you would like to find the average
                                                    of:' );

  readln ( num1, num2, num3 );
  writeln;
  writeln ( 'Total':9, num1+num2+num3:9 );
  writeln ( 'Average':9, (num1+num2+num3)/3:9:2 );
  readln;
end.
```

Exercise 8

```
program chapter5_exercise8;

uses crt;

var
  fahrenheit : integer;
  centigrade : real;

begin
  clrscr;
  writeln ( 'Enter temperature in Fahrenheit:' );
  readln ( fahrenheit );
  centigrade := ( fahrenheit - 32 ) * ( 5/9 );
  writeln;
  writeln ( fahrenheit, 'F =', centigrade:8:2, '°C' );
  readln;
end.
```

- ★ To output °C press ALT+248 and then press C.

Chapter 6

Exercise 9a

num1 and num2	equal	big_1	small_1	big_2	small_2
20 and 10	false	false	true	false	true
20 and 20	true	false	true	false	true
50 and 10	false	true	false	false	true
30 and 20	false	false	false	false	true
60 and 30	false	true	false	false	false
40 and 50	false	true	false	true	false

Exercise 9b

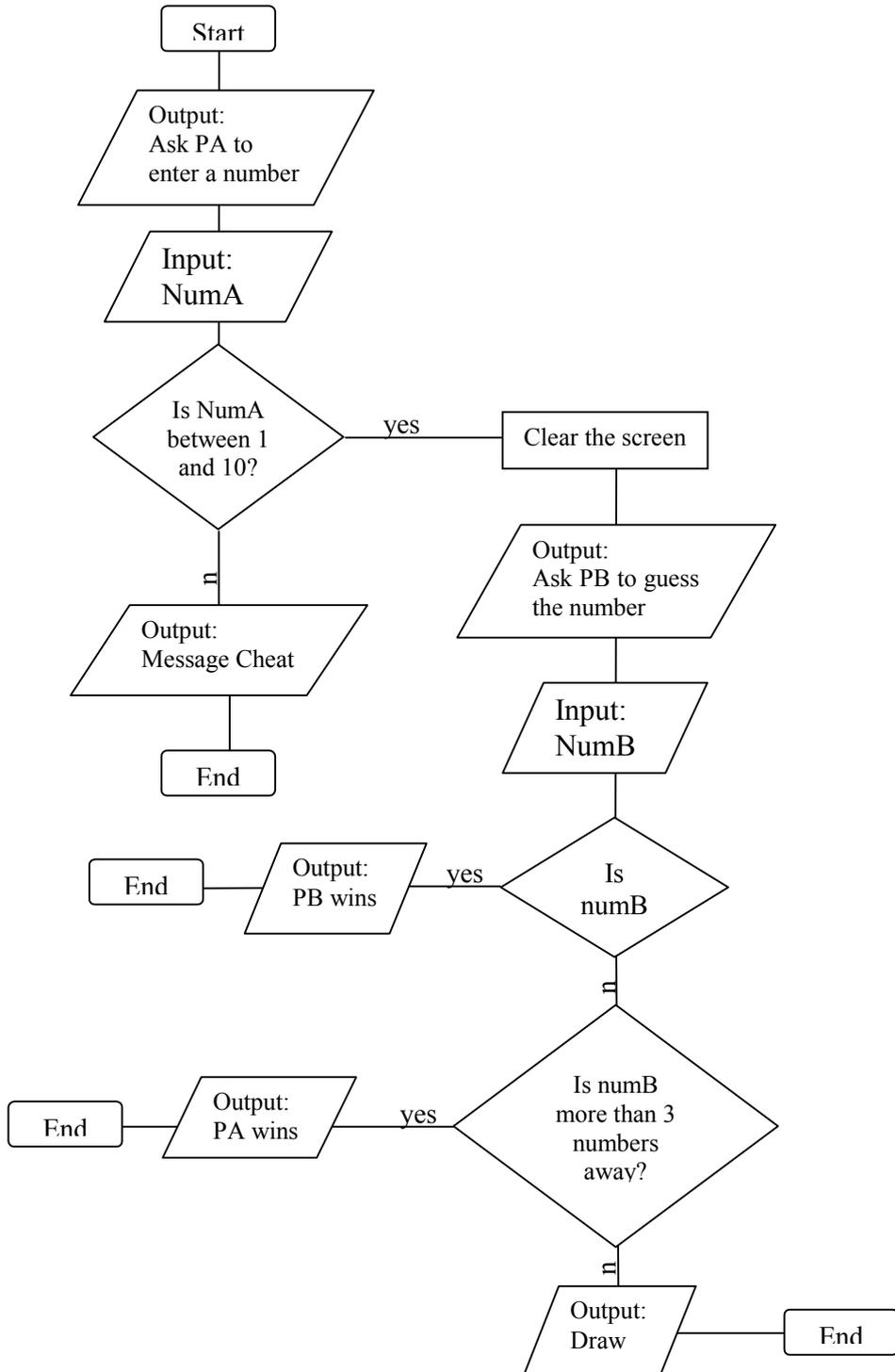
```
big_2 := ( num2 > num1 ) and big_1;
small_2 := ( num2 < num1 ) and small_1;
```

Chapter 7

Exercise 10

Flowchart

[PA refers to Player A and PB refers to Player B]



```

program chapter7_exercise10;

uses crt;

var
    numA, numB : integer; { These will hold the numbers entered by
                           player A and player B respectively }

begin
    clrscr;
    writeln ( 'Player A: Enter a number between 1 and 10' );
    readln ( numA );
    if ( numA < 1 ) or ( numA > 10 ) { Checking that the number
                                     entered by player A is within
                                     the range }
    then writeln ( 'You are CHEATING!!!' )
    else begin
        clrscr;
        writeln ( 'Player B: Try and guess the number entered by
                                                           player A' );

        readln ( numB );
        if numB = numA
        then writeln ( 'Player B: You WIN!' )
        else begin
            if ( numB > ( numA + 3 )) or ( numB < ( numA - 3 ))
            then writeln ( 'Player A: You WIN!' )
            else writeln ( 'Draw' );
        end;
    end;
    readln;
end.

```

Exercise 11a

```

program chapter7_exercise11;

uses crt;

const
    min = 1;
    max = 100;

var
    number : integer;

begin
    clrscr;
    writeln ( 'Enter a number between 1 and 100:' );
    readln ( number );
    if ( number < min ) or ( number > max )
    then writeln ( 'Number is out of range!' )
    else writeln ( 'Number is within range' );
    readln;
end.

```

Exercise 11b

It is ideal to declare the minimum and maximum values as constants so that if we need to change the range, we simply change the values under the **const** declaration.

Exercise 12a

M	N	Output
6	9	3
7	3	1
8	4	4
12	8	4

Exercise 12b

It finds the highest factor of the largest number inputted.

Exercise 12c

Swapping M and N means placing the contents of the variable M into N and placing the contents of variable N into M.

```
program chapter7_exercise12c;
uses crt;
var
  n, m, temp : integer;
begin
  temp := m;
  m := n;
  n := temp;
end.
```

Exercise 12d

```
r := m mod n;
```

Exercise 12e

If either M or N were zero, we would be trying to make the computer divide by zero, an operation, which is not allowed. Thus, the program would stop running, giving a runtime error.

Exercise 13

```
program chapter7_exercise13;

uses crt;

var
  num1, num2, num3 : integer;

begin
  clrscr;
  writeln ( 'Enter three numbers:');
  readln ( num1, num2, num3 );
  writeln;
  if ( num1 > num2 )
    then if ( num1 > num3 )
      then writeln ( num1, ' is the largest' );
  if ( num2 > num1 )
    then if ( num2 > num3 )
      then writeln ( num2, ' is the largest' );
  if ( num3 > num1 )
    then if ( num3 > num2 )
      then writeln ( num3, ' is the largest' );
  readln;
end.
```

Exercise 14

```
program chapter7_exercise14;

uses crt;

var
  year : integer;

begin
  clrscr;
  write ( 'Enter a year:' );
  readln ( year );
  writeln;
  if ( year mod 400 ) = 0
    then writeln ( 'Year ', year, ' is a Leap Year!!!' )
  else if ( year mod 100 ) = 0
    then writeln ( 'Year ', year, ' is NOT a Leap Year.' )
  else if ( year mod 4 ) = 0
    then writeln ( 'Year ', year, ' is a Leap Year!!!' )
  else writeln ( 'Year ', year, ' is NOT a Leap Year.' );
  readln;
end.
```

Chapter 8

Exercise 15a

Once.

Exercise 15b

None.

Exercise 15c

Three times.

Exercise 15d

None.

Exercise 15e

Five times.

Exercise 16a

```
program chapter8_exercise16a;

uses crt;

var
  j : integer;

begin
  clrscr;
  { for...do...loop }
  for j := -4 to 20
    do write ( j : 8 );

  writeln;
  writeln;

  { repeat... until...loop }
  j := -4;
  repeat
    write ( j : 8 );
    j := j + 1
  until ( j > 20 );

  readln;
end.
```

Exercise 16b

```
program chapter8_exercisel6b;

uses crt;

var
  j : integer;

begin
  clrscr;
  { for...do...loop }
  for j := -4 to 20
    do write ( j : 8 );

  writeln;
  writeln;

  { while... do...loop }
  j := -4;
  while ( j < 21 )
    do begin
      write ( j : 8 );
      j := j + 1;
    end;

  readln;
end.
```

Exercise 17

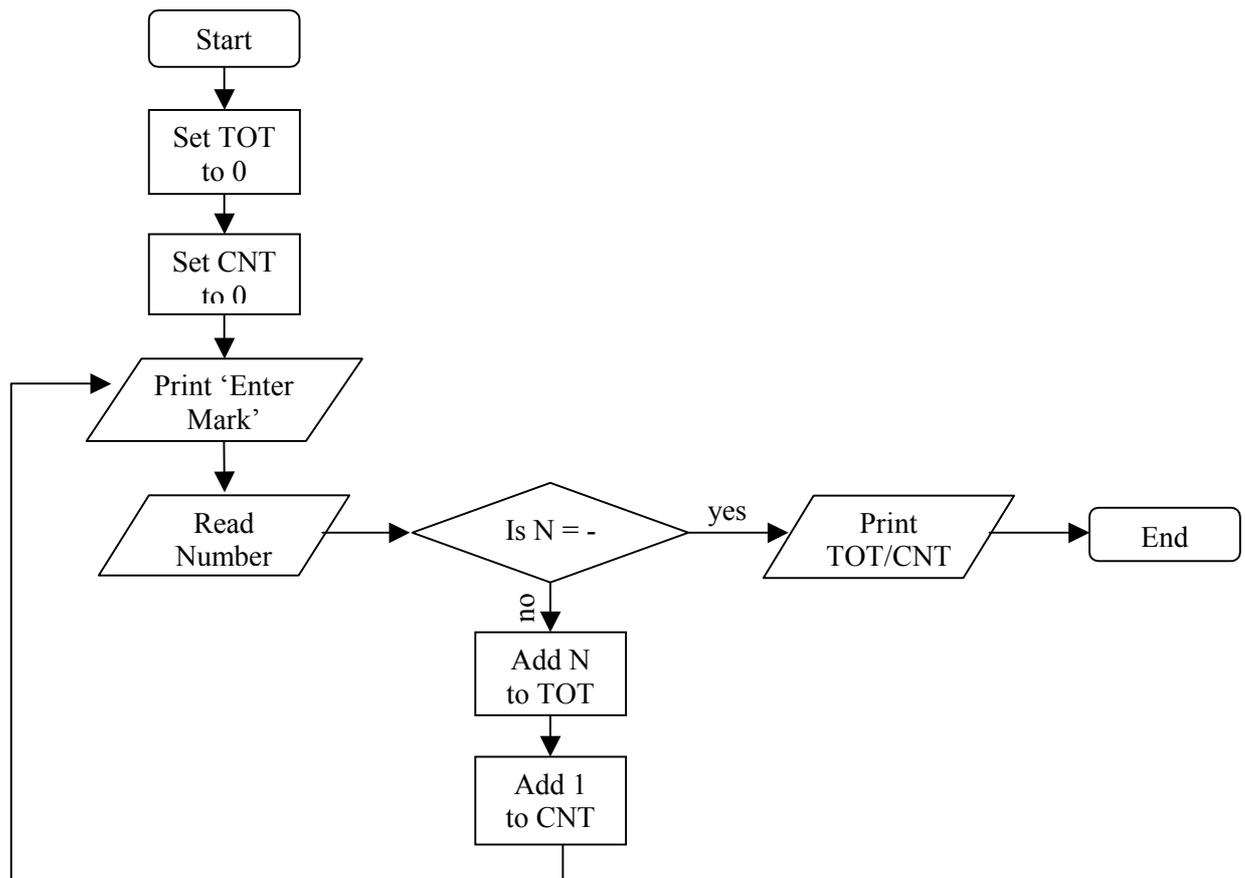
```
program chapter8_exercisel17;

uses crt;

var
  number, i : integer;

begin
  clrscr;
  write ( 'Enter a number: ' );
  readln ( number );
  writeln;
  for i := 1 to 12
    do writeln ( i:2, ' x ', number, ' = ', i*number );
  readln;
end.
```

Exercise 18a



Exercise 18b

The program will stop with a runtime error if the user enters -1 as the first entry. This will cause **TOT** to be divided by zero; since **CNT** at that point would still be zero. Pascal gives a runtime error on dividing by zero.

Exercise 19a

An algorithm is a finite set of rules giving a sequence of operations for solving a specific type of problem.

Exercise 19b

Outputs the average of seven inputted numbers.

Exercise 19c

In step 3, it represents input.

In step 7, it represents output.

Exercise 19d

It represents a decision statement.

Exercise 19e

Total, Count, N

Exercise 19f

Steps 1 and 2

Exercise 19g

Steps 3, 4, 5 and 6

Exercise 19h

```
program chapter8_exercise19h;

uses crt;

var
  total, count, n : integer;

begin
  clrscr;
  total := 0;
  count := 0;
  repeat
    readln ( n );
    total := total + n;
    count := count + 1;
  until count = 7;
  writeln ( total / 7 );
  readln;
end.
```

Chapter 9

Exercise 20

```
program chapter9_exercise20;

uses crt;

var
  a, b : integer;
  c : real;

begin
  clrscr;
  writeln ( 'This program calculates the value of the hypotenuse using
           Pythagoras Theorem' );

  writeln ( 'Enter the length of the sides:' );
  readln ( a, b );
  c := sqrt ( sqr ( a ) + sqr ( b ) );
  writeln;
  writeln ( c : 7 : 2, ' is the length of the hypotenuse.' );
  readln;
end.
```

Exercise 21

```
program chapter9_exercise21;

uses crt;

var
  angle_in_degrees : integer;
  radians : real;

begin
  clrscr;
  write ( 'Enter an angle in degrees:' );
  readln ( angle_in_degrees );
  radians := ( angle_in_degrees * ( 2 * PI ) ) / 360;
  writeln;
  writeln ( sin ( radians ) : 7 : 2, ' is the sine of angle ',
           angle_in_degrees );
  writeln ( cos ( radians ) : 7 : 2, ' is the cosine of angle ',
           angle_in_degrees );

  readln;
end.
```

Exercise 22

```
program chapter9_exercise22;

uses crt;

var
  radians, degrees : real;

begin
  clrscr;
  write ( 'Enter value: ' );
  readln ( radians );
  degrees := ( radians * 360 ) / ( 2 * pi );
  writeln;
  writeln ( arctan ( degrees ) : 7 : 2, ' is the angle ' );
  readln;
end.
```

Chapter 10

Exercise 23

```
program chapter10_exercise23;

uses crt;

var
  title : string[80];
  counter : integer;

begin
  clrscr;
  writeln ( 'Enter the text:' );
  readln ( title );
  clrscr;
  for counter := 1 to length ( title )
  do if ( counter mod 3 ) = 0
    then begin
      textcolor ( white );
      write ( title [counter] );
    end
    else if ( counter mod 3 ) = 1
    then begin
      textcolor ( magenta );
      write ( title [counter] );
    end
    else begin
      textcolor ( cyan );
      write ( title [counter] );
    end;

  readln;
end.
```

Exercise 24

```
program chapter10_exercise24;

uses crt;

var
  full_name : string[80];
  name, surname : string[40];
  counter : integer;
  inname : boolean;

begin
  clrscr;
  writeln ( 'Enter a name' );
  readln ( full_name );
  name := '';
  surname := '';
  inname := true;
  for counter := 1 to length ( full_name )
    do begin
      if full_name[counter] = ' '
        then inname := false
        else if inname
          then name := name + full_name[counter]
          else surname := surname + full_name[counter]
        end;
      writeln;
      writeln ( name );
      writeln ( surname );
      readln;
    end.
end.
```

Chapter 11

Exercise 25

```
program chapter11_exercise25;

uses crt;

type
  string80 = string[80];

procedure wrapc ( x, y, colour : byte; phrase : string80 );
begin
  gotoxy ( x, y );
  textcolor ( colour );
  write ( phrase );
end;

begin
  clrscr;
  wrapc ( 17, 6, 11, 'Programming Project' );
  wrapc ( 17, 8, 11, 'MatSec O Level' );
  wrapc ( 17, 15, 12, 'Sean Vella' );
  wrapc ( 17, 16, 12, 'May 2001' );
  readln;
end.
```

Exercise 26

```
program chapter11_exercise26;

uses crt;

var
  c : byte;

procedure wrapc ( x, y, colour : integer; c : char );
begin
  gotoxy ( x, y );
  textcolor ( colour );
  write ( c );
end;

procedure draw_border ( colour : byte );
var
  i : byte;
begin
  wrapc ( 1, 1, colour, chr ( 218 ) );
  wrapc ( 80, 1, colour, chr ( 191 ) );
  wrapc ( 1, 24, colour, chr ( 192 ) );
  wrapc ( 80, 24, colour, chr ( 217 ) );
  for i := 2 to 79
    do begin
      wrapc ( i, 1, colour, chr ( 196 ) );
      wrapc ( i, 24, colour, chr ( 196 ) );
    end;
  for i := 2 to 23
    do begin
      wrapc ( 1, i, colour, chr ( 179 ) );
      wrapc ( 80, i, colour, chr ( 179 ) );
    end;
end;
```

```

        end;
    end;

begin {main}
    clrscr;
    textcolor ( lightgray );
    writeln ( 'Enter a number between 1 and 15:' );
    readln ( c );
    clrscr;
    draw_border ( c );
    readln;
end. {main}

```

Chapter 12

Exercise 27

```

program chapter12_exercise27;

uses crt;

var
    a, b : integer;

function is_factor_of ( na, nb : integer ) : boolean;
begin
    if ( nb mod na ) = 0
    then is_factor_of := true
    else is_factor_of := false;
end;

begin
    clrscr;
    writeln ( 'Is A a factor of B?' );
    write ( ' Enter A: ' );
    readln ( a );
    write ( ' Enter B: ' );
    readln ( b );
    writeln;
    if is_factor_of ( a, b )
    then write ( a, ' is a factor of ', b )
    else write ( a, ' is NOT a factor of ', b );
    readln;
end.

```

Exercise 28

```
program chapter12_exercise28;

uses crt;

type
  string80 = string[80];

function ask_yn ( prompt : string80 ) : boolean;
var
  c : char;
begin
  gotoxy ( 15, 12 );
  writeln ( prompt );
  readln ( c );
  if ( c = 'y' ) or ( c = 'Y' )
  then ask_yn := true
  else ask_yn := false;
end;

begin
  clrscr;
  repeat
  until ask_yn ( 'Do you want to quit? Y/N' )
end.
```

Chapter 13

Exercise 29

```
program chapter13_exercise29;

uses crt;

const
  maxstudents = 5;

type
  student_details = array [ 1 .. maxstudents ] of string[25];

var
  names, surnames : student_details;
  i, string_counter : integer;
  fullname : string[50];
  name, surname : string[25];
  inname : boolean;

begin
  clrscr;
  for i := 1 to maxstudents
  do begin
    textcolor ( i + 1 );
    write ( 'Enter full name for student ', i, ': ' );
    readln ( fullname );
    name := '';
    surname := '';
  end;
```

```

inname := true;
for string_counter := 1 to length ( fullname )
do begin
    if fullname[string_counter] = ' '
    then inname := false
    else if inname
        then name := name + fullname[string_counter]
        else surname := surname + fullname[string_counter]
    end;
names[i] := name;
surnames[i] := surname;
end;

{ Displaying list of students' names on the screen }
clrscr;
textcolor ( lightgray );
writeln ( 'Students :' );
writeln;
for i := 1 to maxstudents
do begin
    textcolor ( i + 1 );
    write ( names[i], ' ' );
    writeln ( surnames[i] );
end;
readln;
end.

```

Exercise 30

```

program chapter13_exercise30;

uses crt;

var
    numbers : array [ 1 .. 3 ] of integer;
    i : integer;

procedure swap ( locationa, locationb : integer );
var
    toswap : integer;
begin
    if numbers[locationa] > numbers[locationb]
    then begin
        toswap := numbers[locationa];
        numbers[locationa] := numbers[locationb];
        numbers[locationb] := toswap;
    end;
end;

procedure order_numbers;
begin
    swap ( 1, 2 );
    swap ( 2, 3 );
    swap ( 1, 2 );
end;

begin
    clrscr;
    writeln ( 'Enter three numbers:');
    for i := 1 to 3

```

```

do readln ( numbers[i] );

order_numbers;

{ Displaying numbers in order }
writeln;
for i := 1 to 3
do write ( numbers[i] : 5 );
readln;
end.

```

Exercise 31a

Places the largest number of the array contents in the last array location. Thus, since 75 is the largest number, the algorithm places this number in array location A[7].

Exercise 31b

```

program chapter13_exercise31b;

uses crt;

var
a : array [ 1 .. 7 ] of integer;
n, i, toswap, counter : integer;

begin
clrscr;
n := 7;
i := 1;
repeat
while ( a[i] > a[i+1] )
do begin
toswap := a[i];
a[i] := a[i+1];
a[i+1] := toswap;
end;
inc ( i );
until ( i = n );
for counter := 1 to 7
do write ( a[counter] : 5 );
readln;
end.

```

Exercise 31c

The largest two numbers would have taken the last two locations in the array.

Exercise 31d

All the contents of the array would be sorted.

Exercise 32a

M = 5

Exercise 32b

L	R	M
1	10	5
6	10	8
6	7	7

Exercise 32c

```
program chapter13_exercise32c;

uses crt;

var
  a : array [ 1 .. 10 ] of integer;
  l, r, m, s : integer;

begin
  clrscr;
  l := 1;
  r := 10;
  write ( 'Which number do you want to look for? ' );
  readln ( s );
  m := trunc ( ( l + r ) / 2 );
  while ( s <> a[m] ) and ( l < r )
  do begin
    if ( s > a[m] )
    then l := ( m + 1 )
    else r := ( m - 1 );
    m := trunc ( ( l + r ) / 2 );
  end;
  if s = a[m]
  then write ( 'Number has been FOUND!' )
  else write ( 'Number was NOT found' );
  readln;
end.
```

Exercise 33a

```
A := A * 4;
```

Exercise 33b

```
write ( B[5] );
```

Exercise 33c

```
for counter := 1 to 100
do write ( counter : 4 );
```

Exercise 33d

```
if ( C >= 0 )
then write ( 'Positive' )
else write ( 'Negative' );
```

Exercise 33e

```
repeat
  readln ( D );
until not (( d < 1 ) or ( d > 10 ));
```

Exercise 34a

```
Y := Y - 2;
```

Exercise 34b

```
write ( 'Enter length: ' );
readln ( L );
write ( 'Enter breadth: ' );
readln ( B );
P := ( L + B ) * 2;
writeln ( 'The perimeter is: ', P );
```

Exercise 34c

```
if ( Num1 > Num2 )
  then write ( Num2 )
  else write ( Num1 );
```

Exercise 34d

```
T := (( 2 * pi ) * ( sqrt ( L / g )));
```

Exercise 34e

```
for counter := 1 to 12
  do writeln ( counter : 2, ' x 5 = ', ( counter * 5 ));
```

Chapter 14

Exercise 35

```
program chapter14_exercise35;

uses crt;

var
  total, i, generated : integer;

begin
  clrscr;
  randomize;
  total := 0;
  writeln ( 'The numbers generated were : ' );
  for i := 1 to 5
  do begin
    generated := ( random ( 1000 ) + 1 );
```

```

        total := total + generated;
        writeln ( generated : 25 );
    end;
writeln;
writeln;
writeln ( 'Total = ', total );
readln;
end.

```

Exercise 36

```

program chapter14_exercise36;

uses crt;

var
    which_table, generated, answer, i : byte;

begin
    clrscr;
    randomize;
    write ( 'Choose a multiplication table from 2 to 10: ' );
    readln ( which_table );
    for i := 1 to 4
    do begin
        writeln;
        writeln;
        generated := random ( 13 );
        write ( 'What is ', which_table, ' x ', generated, '? ' );
        readln ( answer );
        if ( answer = ( which_table * generated ))
            then write ( 'RIGHT!!' )
            else write ( 'WRONG, the answer is ', ( which_table *
                                                         generated ));
        end;
        readln;
    end.

```

Theory

Section 1 Chapter 1

PAPER 1 – SAMPLE ANSWERS

1. (a) What do you understand by process control?

Process control is when the output is redirected to the input to control the ongoing process.

- (b) Give an example of its use in the industry.

An ABS braking system is one example when the brake pedal is pressed the control unit is continually monitoring the pressure on the pedal, until you let go.

2. (a) What is meant by saying that the MILEOMETER of a car displays its results in *digital form*, while the SPEEDOMETER provides an *analogue* display?

The reading from a car mileometer is read in one chunk, you don't bother to report any fraction because even while the last digit is appearing half way you decide to quote it or not.

The speedometer on the other hand is continuously changing and it is difficult to take an accurate reading.

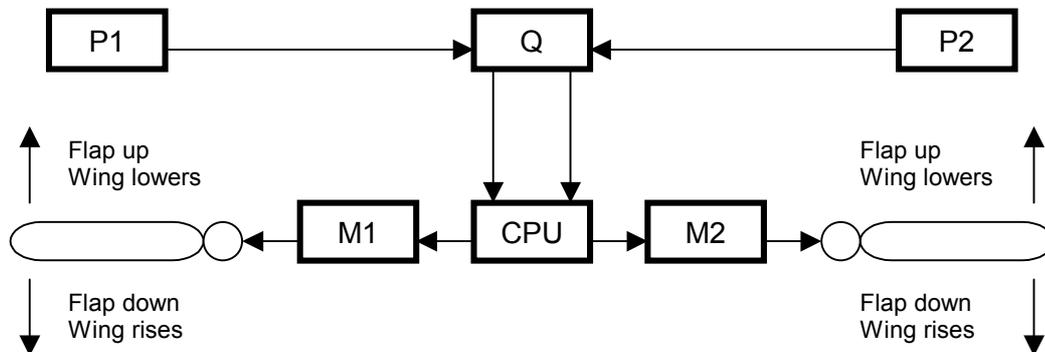
- (b) Explain what is meant by an *analogue-to-digital converter* and give one example of it use.

Computers only understand digital signals, to communicate with other computers which are connected via telephone lines such as the internet, one needs a converter. This converter must translate the analogue voltage of a telephone line to digital and vice versa.

PRACTICE EXERCISES – PAPER 2B

1. An aeroplane uses two pressure sensors at the tip of each wing. These pressure sensors translate air pressure to the equivalent altitude – height, in the form of an analogue voltage to box ‘Q’. They have another vital function of stabilising the aircraft. The general plan of the sensors and devices are shown in the diagram below.

- (a) What is the box labelled ‘Q’ called and why is it necessary to have.



Box ‘Q’ is an analogue to digital converter and it is required because the sensors give analogue voltage and the computer understands only digital signal. An analogue to digital converter translates digital to analogue and vice versa.

- (b) What is the condition for the aeroplane to be stable in the air? Tick the correct box.

$P1 > P2$	<input type="checkbox"/>	$P1 < P2$	<input type="checkbox"/>	$P1 = P2$	<input checked="" type="checkbox"/>
-----------	--------------------------	-----------	--------------------------	-----------	-------------------------------------

- (c) Look carefully at the diagram above. What happens when P1 gives a higher reading than P2 ? Follow the passage below and fill in and tick the boxes as you think appropriate.

The **analogue** data from P1 and P2 is fed to ‘Q’ and converted into **digital** format because the CPU only understands this. The CPU compares these two signals,

If $P1 > P2$ then the CPU will switch on:

Motor M1 only	<input type="checkbox"/>	Motor M2 only	<input type="checkbox"/>	Motor M1 and Motor M2	<input checked="" type="checkbox"/>
---------------	--------------------------	---------------	--------------------------	-----------------------	-------------------------------------

Then,

FLAP 1	Moves up	<input checked="" type="checkbox"/>	Moves down	<input type="checkbox"/>	Unchanged	<input type="checkbox"/>
FLAP 2	Moves up	<input type="checkbox"/>	Moves down	<input checked="" type="checkbox"/>	Unchanged	<input type="checkbox"/>

- (d) In heavy air turbulence the CPU is receiving too many readings where there is a difference between P1 and P2 and many times the CPU cannot correct quickly enough and gives an error. The company, which makes the stabilising unit, has suggested it will replace the 300Mhz CPU with a 1Ghz CPU. Do you think this is a good solution? Why?

Yes because the CPU will react much faster to each air turbulence before the next air turbulence affects the aeroplane. This will therefore help the aeroplane keep a smoother flight.

- (e) What will happen at present if P1 or P2 fails?

If P1 fails P2 will always give a higher reading and the system will not function well and may endanger the flight. The same happens if P2 fails.

- (f) What do you suggest to do in order to avoid this?

You can double the sensors at each end.

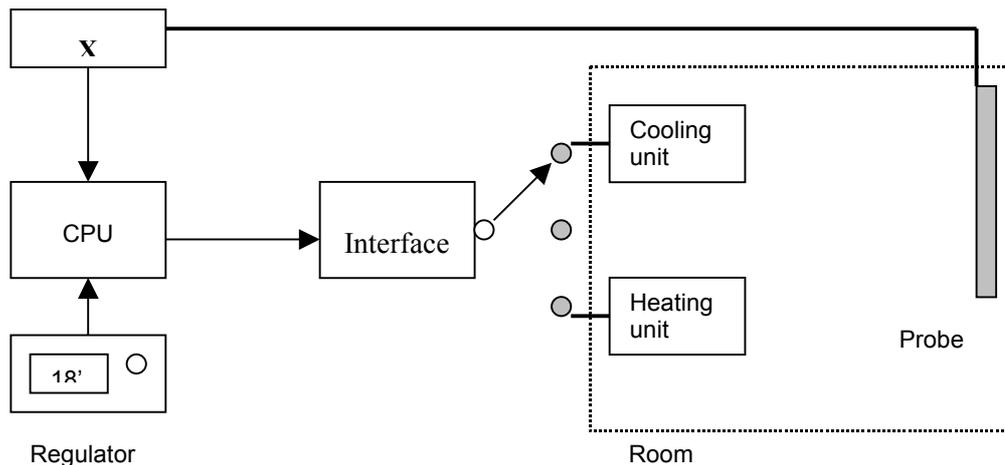
- (g) Do you note another weakness in the system hardware?

Yes, if the CPU fails the system will not work.

PRACTICE EXERCISES – PAPER 2A

1. A company manufacturing air conditioners has decided to modernise its unit by embedding a dedicated computer to control the system. An analogue thermometer probe reads room temperature and feeds this analogue signal to 'X'. 'X' then changes this signal to another form which the computer understands.

The computer then compares the room temperature to the required temperature set at the 'regulator unit'. After computing this result the computer 'decides' whether to switch on, the 'cooling unit' or the 'heating unit' or leave both switched off.



- (a) What is the meaning of analogue data? Give another example of analogue data.

Analogue data means continuously changing data such as voltage at a telephone line.

- (b) What is a dedicated computer? Give two more examples of dedicated computers found in modern homes.

A dedicated computer is a computer designed to solve a specific task. It is unable to perform other tasks than that for which it was designed.

Two examples of a dedicated computer are the VCR - videocassette recorder and the automatic washing machine – with an inbuilt washing program.

- (c) The output reading from the thermometer probe is continuously redirected to the input of the CPU. What is the type of process called?

This is known as process control and the input uses the feedback from the output.

- (d) The room temperature reading by the thermometer probe is 26 degrees Celsius. John sets the regulator at a temperature of 18 degrees Celsius. The process involved to adjust the temperature is shown incomplete below. Fill in the required information.

- The thermometer probe reads the current room temperature 26 degrees Celsius, it then sends this analogue signal to the device called an **Analogue to Digital Converter** which converts this quantity to an equivalent **DIGITAL** signal.

- The converted signal is fed into the CPU, which also 'reads' the required temperature setting at the regulator. The CPU can make three decisions.

If the room temperature is **HIGHER** than required temperature the cooling unit is switched on.

If the room temperature is **LOWER** than required temperature the heating unit is switched on.

If the room temperature is **EQUAL** to the required temperature the units are both switched off.

The program to control the air-conditioning unit is hard coded within a piece of memory chip called ROM. The program uses binary code – a series of '1s' and '0s'. Suppose the program uses a four bit set (a group of four bits either '1s' or '0s'). Two bits to mark if the room temperature is higher or lower than the required temperature and another two bits to indicate if the cooling unit or the heating unit will be switched on. Hint, consider '1' as meaning 'yes' and '0' to mean 'no'.

(e) Complete the following. The first one has been done for you.

Room Temp	Required Temp	Room temp Higher	Room Temp Lower	Cooling unit Switch on	Heating unit Switch on
26	18	1	0	1	0
22	26	0	1	0	1
24	20	1	0	1	0
20	20	0	0	0	0

(f) Using the above notation if you were presented with the following code what would you be able to conclude about the room temperature etc. What is your comment about the code?

Code	0	1	1	0
------	---	---	---	---

Using the same code as above this would mean that the room temperature is lower than the required temperature.

It is also indicating that the cooling unit should be switched on, and this is wrong when in fact the heater should be switched on.

Section 1 Chapter 2

PRACTICE EXERCISES 2 – PAPER 1

1. Explain the following terms:
 - (a) bit smallest unit of storage, can store either a '1' or a '0'
 - (b) byte a collection of eight bits
 - (c) word length the amount of bits a CPU can process at a time
 - (d) kilobyte 1024 bytes (2^{10})
 - (e) Megahertz 1 million cycles per second
2. Central memory is made up of two different types of memory each of which has distinct features. Name these two different types of memory and mention their use in a computer and two different characteristics.

First type: RAM – random access memory

Second type: ROM – read only memory

Use first : used to store programs for immediate access to user

Use second: used to store bootstrap loader

First difference: RAM volatile, ROM permanent

Second difference: RAM stores continuously changing data, ROM can only store data once

3. State the purpose of the **arithmetic logic unit**, the **control unit** and the **main memory** of the central processing unit (CPU).

Arithmetic Logic Unit

The arithmetic logic unit does all the mathematical calculations within the accumulator or the shift registers. It then passes the result back to central memory.

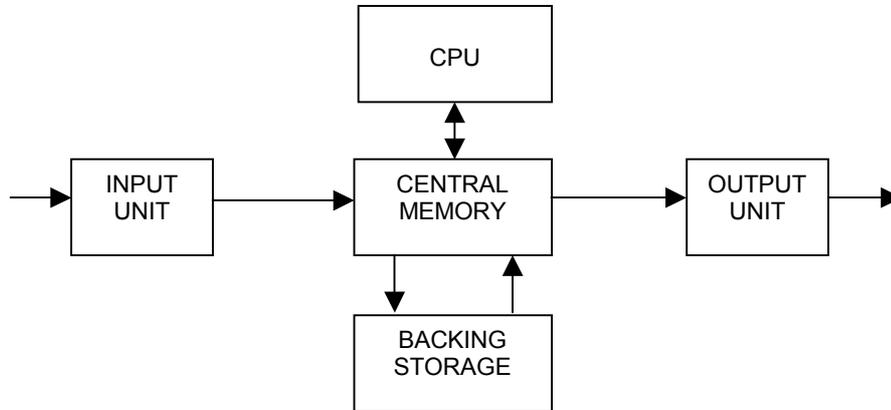
Control Unit

Manages data transfer to and from central memory, registers, and input/output subsystem. Controls the fetch – execute cycle.

Main memory is used to store for immediate access programs which the user is currently using for editing. Once they are saved a copy of the file currently in main memory is saved on backing storage for permanent safe keeping.

PRACTICE EXERCISES 2 – PAPER 2B

1. (a) A computer system will include apart from the CPU, the input/output subsystem, central memory and backing storage. Draw a simple hardware block unit showing the relationship between the units mentioned and show how the data will flow from the input to the output after processing. Show also in you diagram how the control unit interacts control with the other units.



- (b) Name and explain the use of two registers found in the ALU and two other registers found in the Control Unit.

Two registers found in the arithmetic unit and two other in the control unit are:

The ALU

The accumulator is used to store the current result from a calculation.

The shift register, shifts binary values to the right to divide them by 2, and to the left to multiply them by 2.

The control unit

The program counter stores the address of the next instruction to be executed.

The instruction register stores a copy of the current instruction.

- (c) What is the main difference between RAM and ROM?

RAM – random access memory is volatile that is on switching off all data stored in RAM is lost

ROM – read only memory, data stored in ROM is written by manufacturer permanently and cannot be altered.

2. (a) What is the difference between data bus and address bus?

The bus is a series of long parallel copper wire which connects the CPU to central memory and to all the peripherals, along this bus both the address part and data part of an instruction will flow.

The address bus is therefore that section of the bus on which the address will pass. The width of the address bus (how many lines) will determine the address space of the CPU.

The data bus is the remaining part of the bus along which data will flow. The width of the data bus determines the word length of the CPU.

- (b) What would be the effect of having a CPU with a wider data bus? Explain.

A CPU with a wider data bus will pass more bits at a single time. This width determines the word length of the CPU, or how many bytes the CPU can send and receive.

- (c) What difference would it make between an address bus that has 4 lines and an address bus that has 8 lines? Explain.

An address bus with 4 lines will have an address space of $2^4 = 16$ addresses.

An address bus with 8 lines will have an address space of $2^8 = 256$ addresses.

- (d) There are two different CPUs available at the moment. One CPU is a 500MHz 16 bit processor and the other a 500Mhz 32 bit processor. Which is the fastest of the two and give reasons for your choice.

The 500Mhz 32 bit CPU will transfer data twice as much as the other 16 bit CPU with similar speed. 32 bit means that this CPU has the capability to transfer to and from 4 bytes at a time.

- (d) What is backing storage and why is it important?

Backing storage is important because we can store programs and files permanently for future use. Other storage means like RAM are volatile and data is lost if not saved on backing storage media.

- (e) Give two examples of backing storage.

Two examples of backing storage are the hard disk drive with hard disk as media and the floppy disk drive with the floppy disk as media.

- (g) Briefly describe the Fetch-Execute cycle.

1. The CPU sets the Program Counter to the address of the first instruction of the required program.
2. CPU looks at Program Counter and fetches the instruction at the indicated address.
3. The CPU stores a copy of the instruction in the Instruction Register.
4. The Program Counter is incremented by 1 to point to the next instruction.
5. The CPU executes the instruction.
6. Repeats step 2 through step 5 until it encounters a 'stop' instruction.

Executes means checking that the right circuitry is switched on and other switched off to allow the transfer of data. It may fetch data from memory, transfer it for processing and then return the result to central memory.

PRACTICE EXERCISES 2 – PAPER 2A

1. (a) Why is a 16-bit 60MHz CPU generally faster than an 8-bit 12 Mhz CPU?

There are two reasons why a 16 bit 60Mhz is faster. Firstly the CPU is working at a higher clock speed. Therefore in one second the CPU performs more instructions.

The second reason is that the 16 bit CPU can pass 2 bytes of data at one time, whereas the other can only pass 1 byte of data at the same time.

- (b) Describe how the CPU can read from and write to main memory using the address bus and the data bus.

The address and data bus connect the CPU to main memory. To read from main memory the CPU places the address on the address bus and a request instruction to move the data.

The required circuitry is switched on and the data transferred.

Once the data is processed and the CPU is ready to write back to central memory, the CPU places the data on the data bus and the address on the address part. It then switches on the required circuitry to send the data to the address indicated to central memory.

- (c) Briefly describe the fetch-execute cycle.

1. The CPU sets the Program Counter to the address of the first instruction of the required program.
2. CPU looks at Program Counter and fetches the instruction at the indicated address.
3. The CPU stores a copy of the instruction in the Instruction Register.
4. The Program Counter is incremented by 1 to point to the next instruction.
5. The CPU executes the instruction.
6. Repeats step 2 through step 5 until it encounters a 'stop' instruction.

Executes means checking that the right circuitry is switched on and other switched off to allow the transfer of data. It may fetch data from memory, transfer it for processing and then return the result to central memory.

- (d) What is meant by a CPU's instruction set?

The CPU instruction set is a hard coded program of how the CPU is to perform the various internal tasks, such as additions, subtractions, movement of data etc. Each different maker will have a different instruction set.

Instruction sets many times determine whether a CPU will be fast in doing certain actions or not.

(e) A 600 Kb program is moved from backing storage to RAM on a 500Mhz computer,

(i) how many memory locations will it occupy?

Memory locations are organised in bytes therefore this program will occupy 600,000 memory locations.

(ii) what is the address space required (state address from – to).

The address space required is from 1 to 600,000

(iii) if a program consists of 100 instructions, and on average to execute and the

CPU can execute one instruction every 10 cycles. How long does it take to execute this program?

$$100 \times 10 = 1,000 \text{ cycles}$$
$$1,000 / 500,000,000 = 1 / 500000 \text{ of a second.}$$

2. Find below a table showing a section an assembly language program. Each instruction indicated by a mnemonic is explained by the column at the right.

	IN 100	
	IN 101	
	MOV 100	Move value at location 100 to Accumulator
LOOP	CMP 101	<i>Compare the accumulator with value at 101</i>
	JLESS STOP	If Accumulator is less than 101, jump to STOP
	SUB 101	Subtract value at 101 from Accumulator
	JMP LOOP	Jump to code at LOOP
STOP	STA 102	Store Accumulator to location 102

(a) What is ASSEMBLY LANGUAGE and why was it introduced?

Assembly language is a low level language introduced to make computer programming a little bit easier by using mnemonics.

(b) What is a MNEMONIC? Give one example of a mnemonic used and its meaning.

A mnemonic is a short word that implies a meaning once it is read. For example a mnemonic for adding is ADD or SUB for subtract. These have a corresponding binary code but is transparent to the user.

(c) What is the meaning of OPCODE and OPERAND? Explain with examples.

Opcode is the operational code or the action to be taken, for example the mnemonic ADD.

Operand is the data on which the opcode is to operate such as a value or a label. For example MOV 100 MOV is the opcode and 100 (the contents of memory location 100) the operand.

(d) What will be the output in location 102 for the assembly code above when,

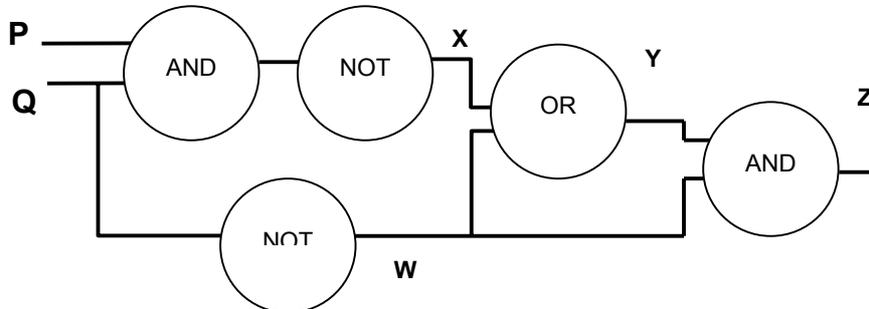
- (i) value in location 100 is 32 and value in location 101 is 7
- (ii) value in location 100 is 7 and value in location 101 is 1
- (iii) value in location 100 is 9 and value in location 101 is 21
- (iv) value in location 100 is 2 and value in location 101 is 0

Answer (i) 4 (ii) 0 (iii) 9 (iv) loops indefinitely

Section 1 Chapter 3

PRACTICE EXERCISES – 3 PAPER 1

1. Figure below shows a logic circuit with inputs P and Q.



(a) Complete the truth table for W, X, Y, Z.

P	Q	W	X	Y	Z
0	0	1	1	1	1
0	1	0	1	1	0
1	0	1	1	1	1
1	1	0	0	0	0

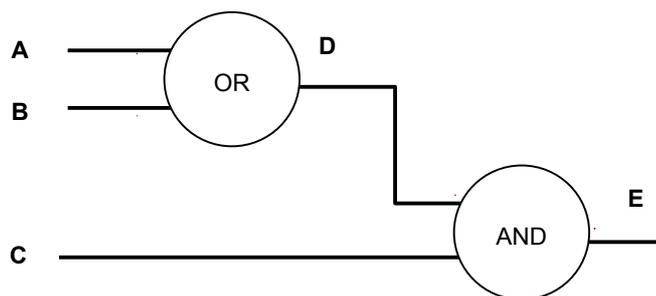
(b) What is the purpose of a truth table?

The purpose of a truth table is to map all possible outputs for every possible input combination.

(c) Extract the Boolean expression for the above circuit.

$$Z = ((\text{NOT } (P \text{ AND } Q)) \text{ OR } (\text{NOT } Q)) \text{ AND } (\text{NOT } Q)$$

2. (a) Complete the following truth table for the circuit shown below.



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

(b) Extract the Boolean expression for the circuit shown.

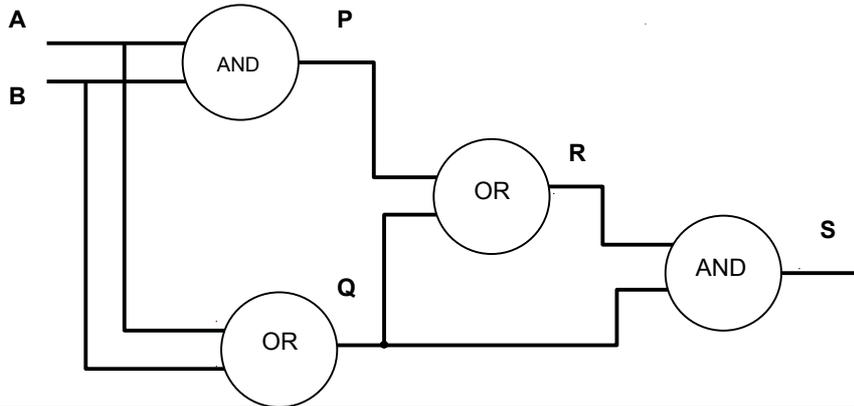
$$E = (A \text{ OR } B) \text{ AND } C$$

PRACTICE EXERCISES – 3 PAPER 2B

1. Figure below shows a logic circuit with some unnamed gates. Its incomplete truth table appears below.

(a) Using the information given in the truth table below, identify and fill in the circuit with the names of the three logic gates.

(b) Complete the truth tables to show the output S.



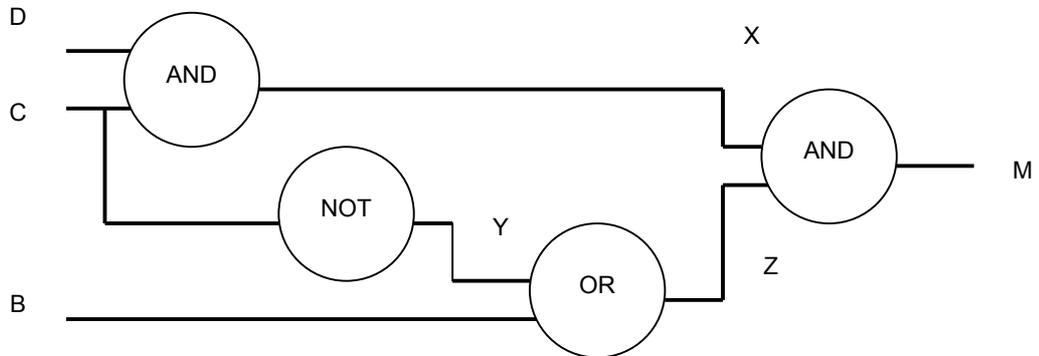
A	B	P	Q	R	S
0	0	0	0	0	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	1	1	1

(b) Extract the Boolean expression for the above circuit.

$$S = ((A \text{ AND } B) \text{ OR } (A \text{ OR } B)) \text{ AND } (A \text{ OR } B)$$

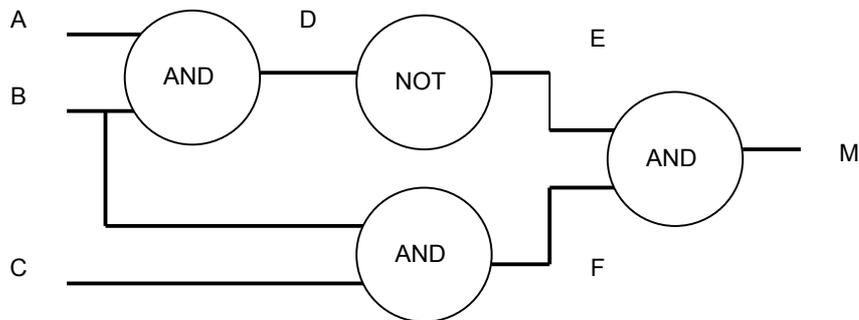
2. Draw the equivalent circuit diagram for the following Boolean expressions and hence draw up and fill properly their corresponding truth table.

(a) $M = (D \text{ AND } C) \text{ AND } ((\text{NOT } C) \text{ OR } B)$



B	C	D	X	Y	Z	M
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	1	1	0
1	0	1	0	1	1	0
1	1	0	0	0	1	0
1	1	1	1	0	1	1

(b) $M = \text{NOT } ((A \text{ AND } B) \text{ AND } (B \text{ OR } C))$



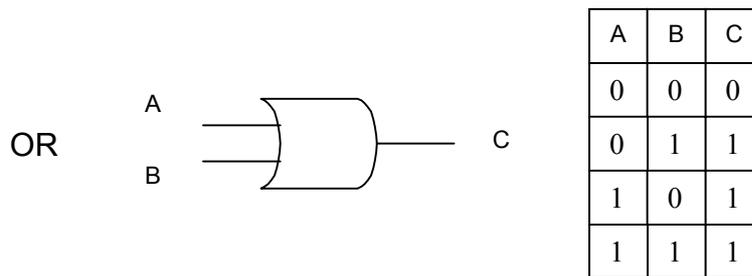
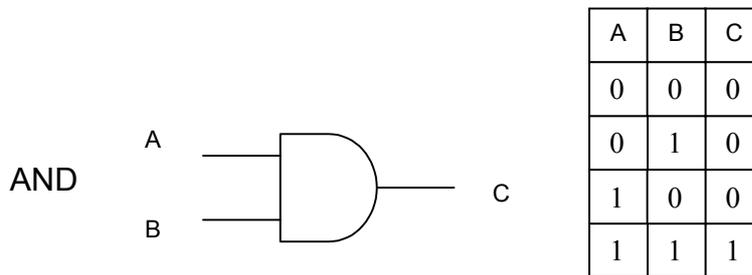
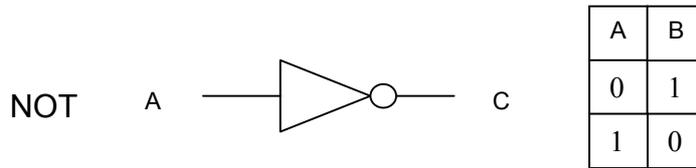
A	B	C	D	E	F	M
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	1	1	1
0	1	1	0	1	1	1
1	0	0	0	1	0	0
1	0	1	0	1	1	1
1	1	0	1	0	1	0
1	1	1	1	0	1	0

PRACTICE EXERCISES – 3 PAPER 2A

1. A car-manufacturing firm wants to introduce new concepts in its high end car range. Among these will include the use of gates to monitor or automate certain tasks.

The first task is to design an alarm, which will light a green lamp when all doors (2 – door sports car) and the front bonnet are locked properly. If these conditions are not met then the car should not start.

(a) Draw the correct symbol for the NOT, AND and OR gates and their equivalent truth table.



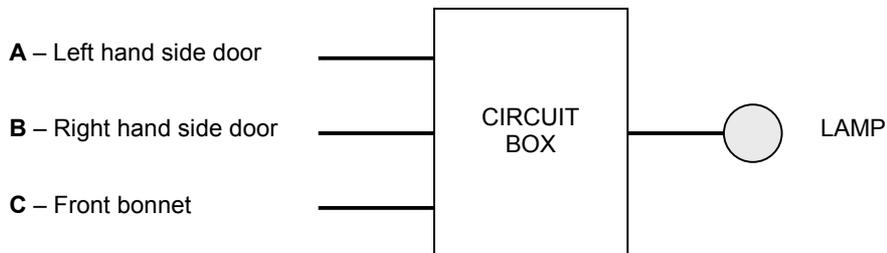
- (c) Observe the truth tables and design the alarm using any combination of NOT, AND, OR gates.

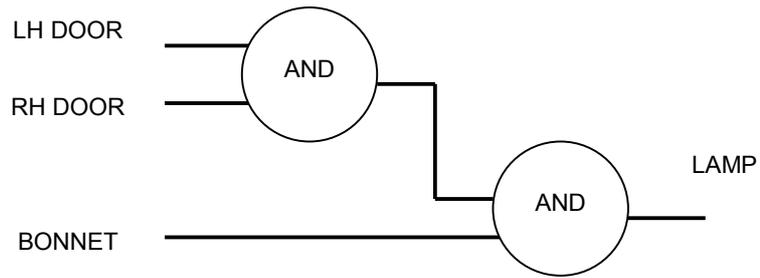
Assume that the binary value ‘1’ indicates door is closed.

The binary value ‘0’ indicates door/s open green light remains off.

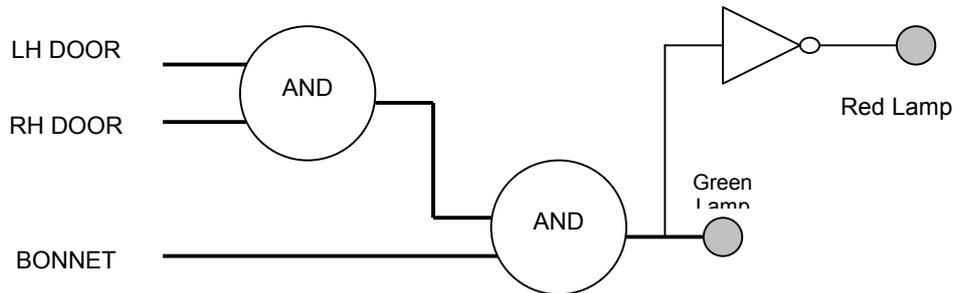
Also binary ‘1’ will switch on lamp, and vice versa.

A black box has been prepared to give you an idea of the input and output required.

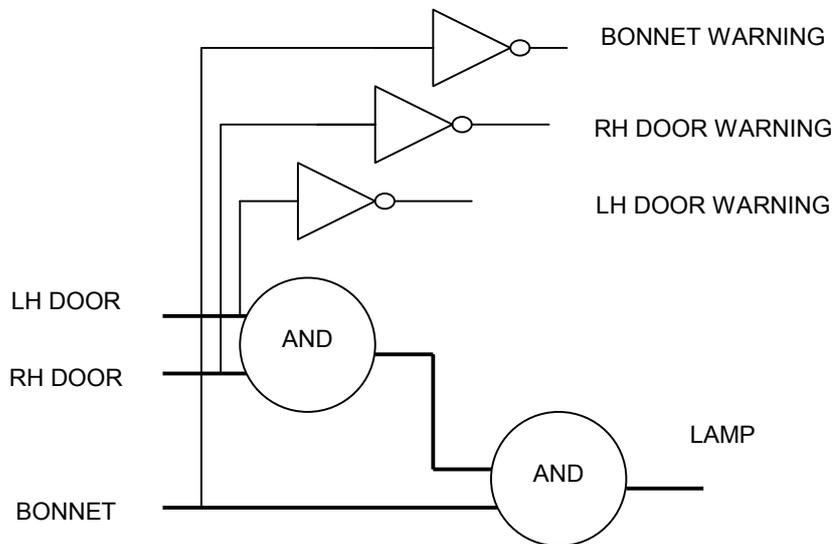




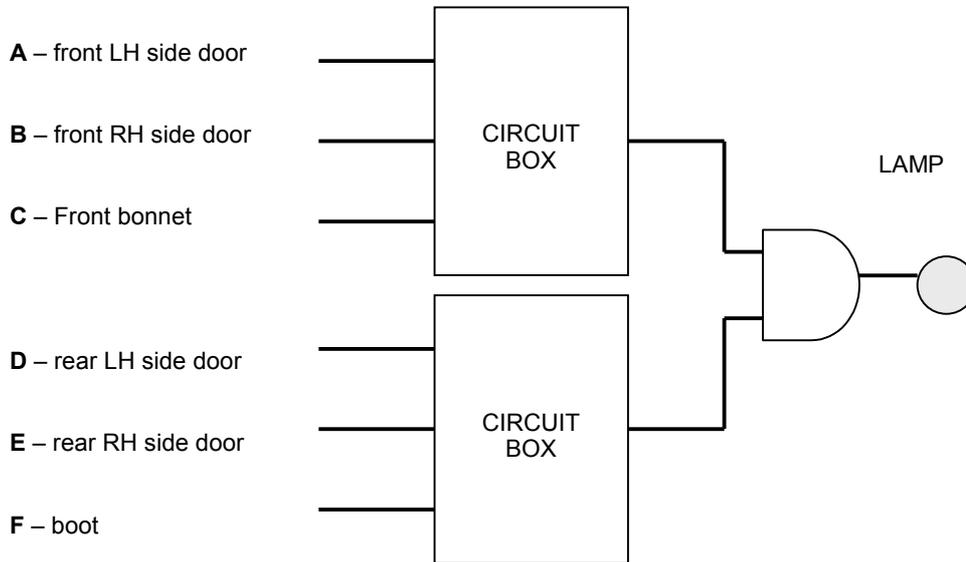
- (c) The green lamp is not enough that it should light, we must also warn the driver of the car that something is wrong. How would you modify your circuit to light a red lamp if one or more of the doors are not properly closed instead of the green lamp.



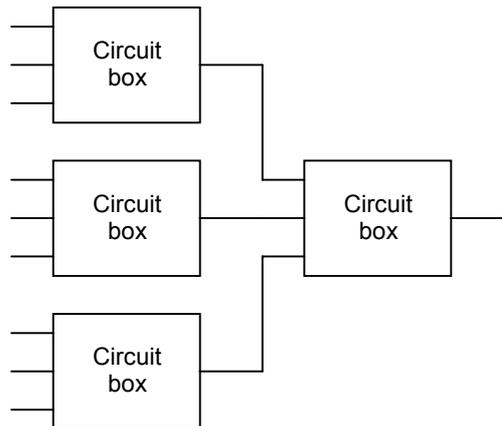
- (d) Is it possible to refine your circuit further to precisely indicate to the driver which door/s is/are not properly closed. Redraw your circuit with the new modifications.



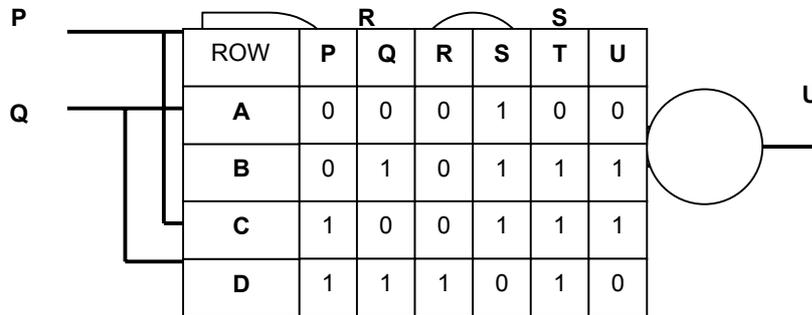
- (e) The car producer would like to introduce this system to a four door saloon to include, all four doors, the bonnet and the boot in the alarm system. How would you design this for him, there is no need to redesign the circuitry, use the circuit box diagram above plus any other necessary circuitry to implement this.



- (f) Keeping the same ideas in question (e) how would you design the inputs from 9 different items to be all '1' in order to signal to the driver that everything is in order and that he can start the car. There is no need to use other circuitry except the circuit box diagram shown above.



2. Below is a circuit diagram of which some logic gates are unmarked and others are shown with their correct electronic symbol.
Fill in the logic gates with proper labels and fill in the truth table.



Gate S = NOT,

Gate U = AND

- (b) By looking at the truth table now complete the following statements.

When A = '0' and B = '0' then	U = 0 and R = 0
When A = '0' and B = '1' then	U = 1 and R = 0
When A = '1' and B = '0' then	U = 1 and R = 0
When A = '1' and B = '1' then	U = 0 and R = 1

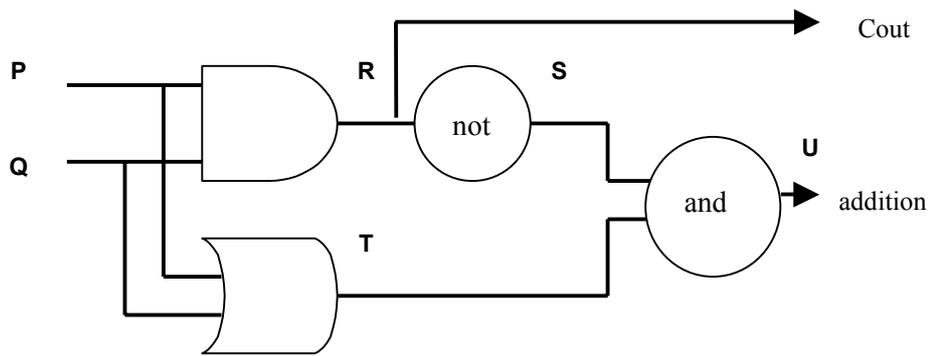
- (c) In binary addition we have the following:

$0 + 0 = 0$
 $1 + 0 = 1$
 $0 + 1 = 1$
 $1 + 1 = 0$ (carry 1 to the next column)
 such that,

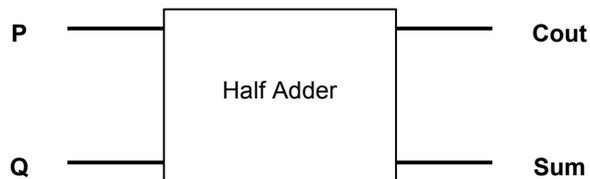
$ \begin{array}{r} 0^1 1^1 1 0 + \\ 0 0 1 1 \\ \hline 1 0 0 1 \\ \hline \end{array} $

Using the circuit above how would you be able to use it to add two binary values? That is at which points (i) do you introduce the values to add (ii) you take the result (iii) you take the carry value when you have '1' + '1'? [3]

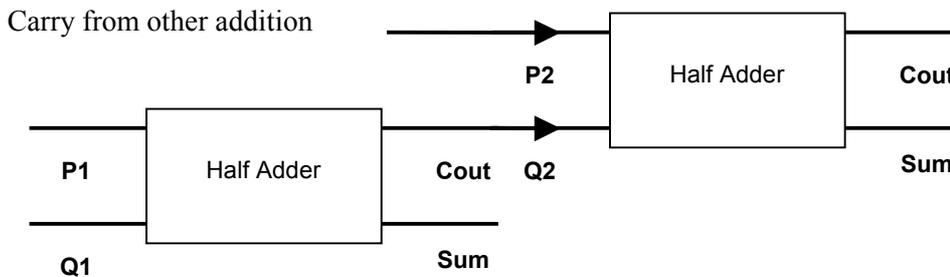
- (i) Introduce the values to add at P, Q.
- (ii) Take the addition result at U.
- (iii) Take the carry result at R.



Let us simplify the circuit above to give us the box below, where P and Q are the same inputs, Cout is the carry forward line and Sum is the addition of the result. In fact this circuit is also known as a half adder. Note, you should have already decided which are equivalent to Cout and Sum.



(e) The above circuit will only add two numbers to produce the Sum and Cout (carry forward) to the next stage. How would you connect two adders so that you will have three inputs, two inputs to add and one to cater for a carry forward from another addition.



Section 1 Chapter 4

PRACTICE EXERCISES – 4 PAPER 1

1. For the following, state with a reason, a suitable computer application for each device.

(a) direct data entry terminal:

application: remote construction site or fast food outlet

reason: special coded messages enable user to enter required data and transmit the order remotely

(b) visual display screen:

application: games

reason: nice colour graphics display

(c) mouse:

application: drawing packages

reason: easy navigation in x and y axis, click to enable commands easy to use.

(d) dot matrix printer:

application: for printing invoices

reason: no NLQ required cheap to run

(e) Why would a bar code on pieces of meat, which vary in weight, not include the price?

Since the weight will vary with every cut it is impossible to issue a standard barcode for such items.

2 (a) What is a SOFTCOPY and a HARDCOPY. Name a device that produces such a copy.

SOFTCOPY

A softcopy is a print out of results which is temporary and cannot be removed or carried around. Such an example is the display of results on a screen, the monitor produces a soft copy.

HARDCOPY

A hardcopy is a printed form of results such as on a sheet of paper. Hardcopies can be carried around and physically handled. A printer will produce a hardcopy.

- (b) Two main categories of printers exist: IMPACT and NON-IMPACT. Fill in the table to show to which category each of the following types of printer belong:

Thermal printer *dot matrix printer* *Laser printer*
Inkjet printer *line printer* *Daisywheel printer*

IMPACT	NON-IMPACT
Dot matrix	Thermal printer
Line printer	Inkjet printer
Daisy wheel	Laser printer

- (c) When choosing a printer one has to consider various factors. Mention FOUR factors, which you consider important:

1. The cost of the printer and the budget available.
2. If the printer is noisy.
3. The print quality.
4. The overall running costs.

- (d) Write down in the table below what each acronym stands for (Note: acronym = word formed from the initial letters of the name).

Device Number	Acronym for device	Full name of device
1	OCR	Optical character reader
2	MICR	Magnetic ink character reader
3	OMR	Optical mark reader

- (e) Put in each adjacent box the device number which:

1	Converts printed characters into electronic signals.
3	Senses a pencil line drawn in a specific position to indicate a particular choice out of several choices.
2	Reads magnetic printing.

PRACTICE EXERCISES – 4 PAPER 2B

1. (a) For each of the following computer hardware devices:

a light pen

an optical character reader (OCR)

a mouse

Explain what each device is used for and state, with a reason, one typical use.

A light pen is used for drawing on screen using light sensing pencil like stylus. A typical use is drawing as used by architects, designers and engineers.

It helps these professionals view exactly as they would draw on paper but the data is stored directly in electronic format.

An optical character reader is used in conjunction with a scanner to translate printed letters to electronic format.

A typical use is to transfer quickly documents from printer paper to electronic format for editing, storage and other forms of manipulation.

A mouse is used for navigating on a graphical user interface. One typical use is for drawing. The mouse is moved over graphical commands, and then you click on screen at the required coordinates to execute the command. A mouse is very easy to master especially with the younger generations.

- (b) For each of the following computer printers:

a line printer

a dot matrix printer

a laser printer

State, with a reason, a typical application for which each is more suitable than any of the others.

A line printer is a high-speed printer and costly. It is used for high-speed cheque printing and other jobs where a high volume of printing is required in a short time.

A dot matrix printer used to be a household type printer. It is slow. Rather cheap to run and rugged. It is now commonly found in retail stores and general merchant shops.

A laser printer provides the best printing quality along with speed. It is a page printer and very silent. It is costly to run when compared to other printer types.

It is found in offices where quality is required. Black and white laser printers are becoming cheaper but laser colour printers are still very expensive.

- (c) With reference to printers explain the following terms.

NLQ and DPI

NLQ means near letter quality and is the best printing mode available on any printer.

DPI means dots per inch, it means how many ink spots there are in one square inch. The higher the DPI the better the print quality, the more ink is used and the slower it prints.

2. Mention an input device and an output device that can be used with ease by a handicapped person.

i. Explain in detail how this is done.

Special software exists where a user can speak out commands. IBM first introduced the software. At first it was not so sophisticated but later improvements were made. The handicapped person first teaches the machine to recognise his voice and then starts building a database of the most required commands.

An output device that would help a handicapped person is the text reader. This special device enables a user to feed in a sheet of paper or a document from the computer and the device will then start to read the contents for the user.

Give one advantage and one disadvantage of the following types of printers:

ii. Dot matrix printer:

Relatively cheap to run, noisy and low print quality.

iii. Daisy wheel printer:

Very good letter quality printing, noisy and no graphics.

iv. Laser printer:

Top quality printing and fast, relatively costly to run.

Virtual reality is a fairly recent application of computer use.

v. Briefly explain what possible applications there are in this area.

Virtual reality means that the user is made to believe that what he is seeing is actually true when it is not. Virtual reality was first used for flight simulators to enable pilots test run planes and gain experience without endangering their lives or the lives of others.

Simulation is also possible to create virtual models of buildings, cities etc. The architects or engineer commission special programs to create a virtual 3D model of the building. The client then can 'walk through' his dream building and check for himself how the end product will look like.

Other application for virtual reality are in the space programs, war games and other scenarios which would be either expensive to construct or dangerous.

vi. State what kind of hardware /software is required.

Special hardware and software are required for virtual reality and are expensive and not readily available for all applications.

Special gear includes headgear, gloves, boots etc. This special gear contains embedded sensors and is connected to the computer. The headgear has a special screen within the helmet, which enables the user to view the program associated with the software.

These special gears are used for viewing models in 3D and are now also available for some games.

PRACTICE EXERCISES – 4 PAPER 2A

1. Many clubs produce illustrated magazine for their members. It is now possible to prepare and print such newsletters on most computer systems.

- (a) What kind of software would typically be used for preparing the newsletter and explain what will be the required features?

To prepare a near professional magazine or newsletter one would need a DTP a desktop publishing software. DTP allows the embedding of pictures within text, special formatting and special fonts, which would not normally be included with other software programs.

DTP programs also allow you to plan ahead if you would like to print a three folder leaflet etc, by having inbuilt templates.

Nowadays though word processors are too becoming sophisticated programs such that you cannot tell the difference between a DTP and a word processor. The trend is that they will end up both having the same identical features.

- (b) Name and explain what kind of software would be used for preparing the illustrations?

For the preparation of illustrations one can adopt a drawing packing or a good scanning software with all the features included. These will include a myriad of features such as cropping, resizing, change colour palette, morph, rotate, dither etc.

- (c) Why is a laser printer more suitable than a dot matrix printer for printing a newsletter?

A laser printer is necessary because in this case printing quality is crucial together with speed. A dot matrix is slow, very low printing quality and poor colour production. A laser printer is also noise free, and special paper is also available for higher quality.

- (d) Briefly explain why a SCANNER would be useful to somebody who is preparing the newsletter. What features would you look for when buying a scanner?

A scanner can provide direct access to many photos and enables other types of printed material to be transformed into electronic format. This can later be manipulated using special software already mentioned.

A scanner is also very useful to scan printed documents to store them on a computer by using OCR software.

- (e) Describe TWO other ideas the club can use the computer for, apart from preparing the magazine.

The computer can also be used to store member records with their membership number address, special interests etc. It can keep track of membership fees, catalogue a library and provide an electronic database also of related articles in electronic format.

The computer can also be used to house a web site, which all members can access from their homes, especially if the club has members who reside in remote areas.

Section 1 Chapter 5

PRACTICE EXERCISES – 5 PAPER 1

1. What is the difference between direct access and serial access?

Serial access means that to locate a record or file one must traverse all preceding records or files. That is, if you would like to locate record number 124 you must traverse all 123 records preceding it. In direct access if you would like to access record number 124 you home in directly on this record.

For the following methods of access involving magnetic storage, state a suitable device and, with a reason, an appropriate application:

- (a) serial access

device: reel to reel tape drive
application: printing pay cheques (payroll)
reason: all records are traversed one after another making sure that all payslips are processed

- (b) direct access

device: hardisk drive
application: searching a record in a database
reason: results are required fast, searching must be quick

2. The floppy disk is a convenient and probably the most commonly used storage medium on PC's although hard disks have now become cheap enough to be affordable.

- (a) Why is it necessary for a computer to have disk storage in addition to main memory?

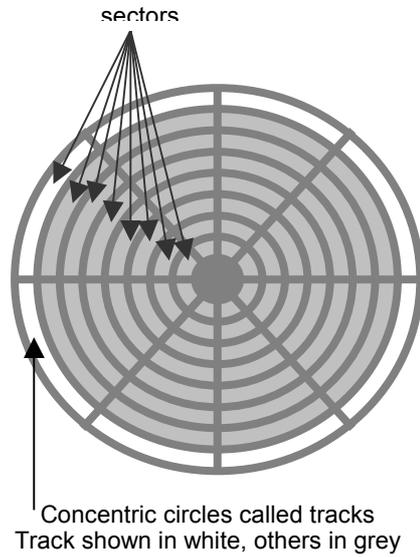
Main memory is volatile meaning all that is stored in main memory will be lost when the computer is switched off. To avoid losing programs or documents on switch off we require a permanent type of storage in the form of hard disk or tape drives and media.

- (b) Give TWO differences between floppy disks and hard disks.

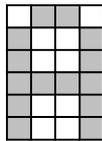
- i. Floppy disk is portable, hard disk is not.
- ii. Floppy disk stores small amounts of data (1.44Mb) whereas hard disks can store over 30Gb of data.

Draw diagrams to illustrate the following:

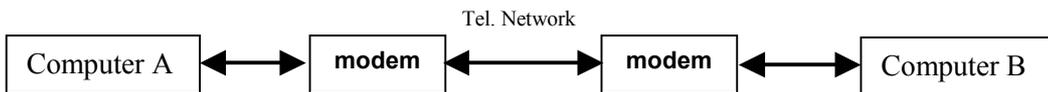
(c) The way a disk's surface is divided into tracks and sectors.



(d) The way a dot-matrix printer might form the letter A.



(e) The way two computers might communicate over telephone lines.



PRACTICE EXERCISES – 5 PAPER 2B

1. (a) Explain why each of the following hardware units are necessary in a computer system:
Input devices, output devices and backing storage.

Input devices are required to enable a user communicate with the computer by inputting data, input devices act as an interface between the user and the computer and will change a human tangible format to an electronic binary format the computer understands.

Output devices on the other hand enable a computed process to print out its results from an electronic format as processed by the computer into a human legible format.

Backing storage are vital since data will not be lost and is stored for future use, otherwise the user will have to re program from scratch every program, which would make it nearly impossible.

- (b) Give *one* input device, *two* output devices and *one* auxiliary storage device, which is typically found in a computer system.

Input device – mouse.

Output device – inkjet printer, monitor.

Auxiliary storage device – hard disk drive.

- (c) Explain how data is stored on a floppy disk.

The floppy disk is first formatted, meaning the surface is divided in imaginary concentric tracks, these are then further subdivided into sectors.

Each sector is now ready to accept data, data will be stored in binary form. Each sector will normally store 256 bytes of data. Each bit is either magnetised = '1' or left demagnetised = '0'.

- (d) Give a suitable application for the following devices/media and sustain your answer with a valid reason.

Reel to reel tape drive and reel to reel tape.

Cheque printing, all records are traversed one by one, no record is skipped.

Hard disk drive and hard disk.

Database of records, retrieval of records is fast because these enable direct access.

CD ROM drive and CD ROM

Multimedia application, such as encyclopaedias.

CD ROM is a read only medium, it can store up to 600Mb of data. This allows the storage of video clips, pictures, animations and text.

- (e) What is the difference between a DVD and a CD ROM

They are both optical media but a DVD can store a greater amount of data than the CD ROM. A DVD can store up to 17Gb of data compared to the 600Mb of the CD ROM

2. (a) Complete the following passage using each word shown only once.

Hard disk drive	tracks	128Mb	CD ROM
Serial	floppy disk drive	Direct	magnetic
20Gb	sectors	floppy disk	zero
optical	one	tape	cassette drive

The **HARDISK DRIVE** is a secondary storage device found in all computers. It stores large amounts of data around **20Gb** for later use. The hard disk has data organised in such a manner that allows **DIRECT** access. Another type of backing storage media is the **TAPE**. Formatting prepares the surface of magnetic disks into **TRACKS** which are then subdivided into **SECTORS** each of which can store a fixed amount of data. Other storage media are **OPTICAL** like for example the **CDROM**, these translate a reflecting surface to binary **ONE** and a non reflecting surface to binary **ZERO**.

PRACTICE EXERCISES – 5 PAPER 2A

- 1a. A floppy disk contains 40 tracks per side. Each track is subdivided into further ten sectors. Each sector can store 256 bytes.

A file of records is to be stored on the diskette. Each record is made up of 80 bytes of data. Each sector will store only whole records, such that a part of each sector may be wasted.

- i. How many records can fit in each sector?

$$256 / 80 = 3 \text{ Remainder } 16 \text{ bytes}$$

Answer 3 records.

- ii. How much is wasted in each sector?

Answer 16 bytes.

- iii. What is the maximum records that can be stored on the diskette?

3 records per sector,
40 tracks x 2 (top and bottom surfaces) = 80 tracks
each track = 10 sectors
therefore total records that can be stored is,
 $80 \times 10 \times 3 = 3,200$ records

Answer 2,400 records.

- iv. What is the total wasted space?

$$80 \times 10 \times 16 = 12,800 \text{ bytes}$$

Answer 12.5Kb.

Give a suitable application for the following types of backing storage media.

Reel to reel tape.

Backup.

Dvd.

Storage of full feature movie films.

Flash memory.

Digital cameras, laptops, digital data.

- 1b. CD-ROMs are becoming increasingly popular storage media.

- i. State TWO advantages and TWO disadvantages of CD-ROMs as compared to more conventional storage media such as magnetic discs.

Advantages of CDROM light weight and portable, stores large amounts of data.

Disadvantages of CDROM, cannot rewrite (read only format) requires special drive.

- ii. The biggest use of CD-ROMs is in the home. Comment.

CD ROMs can store large amounts of data, in the order of 600Mb. This makes it possible to store video clips, animations, photos, diagrams, audio clips and text. These were not possible portable magnetic media as they required large amounts of data storage.

These attributes make the CDROM ideal for storing whole libraries of books, encyclopaedias, specialised teaching aids.

The use of graphics, animations and audio clips attract the attention of the younger generation.

Being in electronic format a CDROM enables quick searching and listing of topics, an ideal attribute for any reference work.

Section 2 Chapter 1

PRACTICE EXERCISES – 1

1. (a) An application package performs tasks such as accounts, stock control, word processing, etc. Mention TWO legal ways such software may be acquired.

First Way: Purchase of software off the shelf.

Second Way Pay a programmer/software house to write the software.

- (b) From your experience in using an application package, give ONE advantage and ONE disadvantage of such a package.

Advantage Using Word it is extremely easy to prepare documents

Disadvantage Does not have the ability to spell/grammar check in Maltese.

2. (a) Give TWO examples of application packages.

Word Processor, Spreadsheet

- (c) What happens when you install a program?

The software is transferred on to the hard disk (from a storage media), and configured to my system.

- (d) What is technical support?

This is support offered by the supplier (normally of software) in order to make it more user friendly.

- (e) What is software upgrade?

The ability to purchase a newer version of the application at a reduced (r even no) cost

- (f) What is piracy?

The copying of software without paying the appropriate licence fees.

Section 2 Chapter 2

PRACTICE EXERCISES – 2

1. You have a choice of buying either an electronic typewriter, or a computer with word processing software. State your choice and give three reasons:

Your choice: Word Processor
First reason Spelling checker
Second reason Automatic margins and word wrap
Third reason Font formatting

2. Use the following list of words to fill in:

*calculations; numbers; cell; columns; automatically; rows;
accountants; cost accounting; text; formula*

A spreadsheet is a sheet of paper divided into **columns** and **rows**. These sheets are used by **accountants** for carrying out complex **calculations**. The advantage of an electronic spreadsheet is that **numbers**, **text** and formulae can be entered into each **cell**. If a **formula** is entered, the spreadsheet will carry out any calculations specified. This ability to **automatically** calculate figures makes spreadsheets ideal for planning applications, such as **cost accounting**. [5]

3. Which FIVE of the following software packages would be most suitable for each of applications listed below?

*Wordprocessor, Spreadsheet, Database, Graphics Package, Desktop Publisher,
Presentation Package, Communications Package*

Application	Software Package
Preparing a school magazine	Desktop Publisher
Writing your CV	Word Processor
Preparing a Loan Repayment Programme	Spreadsheet
Address Book	Database
Writing a Pascal Program	Word Processor

[5×1]

Paper 2 A Questions

1. Software must not only work correctly, it must also be easy to use – a quality sometimes called **user-friendliness**.

(a) Mention THREE features, which make a program user-friendly.

- i. Comprehensive help and well explained error messages
- ii. Well laid out screen with a graphic interface
- iii. Stable and error free application

(b) What is **on-line help**, and what are its advantages?

This is help which is included in the software itself. It is typically activated by pressing the F1 key. This help is normally searchable and in most cases is context sensitive.

(c) Why is a **Graphical User Interface** (GUI) generally considered user-friendlier than a **Command Line Interface** (CLI)?

A GUI is normally more pleasing to the eye. The features available to the user are normally all displayed on the screen and all the user has to do is to select the option that is required.

(d) Mention FOUR things which make a program hard to use, and which are therefore to be avoided when you are writing a program.

- i. Cryptic error messages
- ii. Inadequate help facilities
- iii. Frequent crashing
- iv. Bad screen layout

(e) Describe how modern input/output devices can make computers easier to use for disabled people.

Some modern devices have features, which make them especially adapted for disabled people. For example the screen can be configured to display highly contrasting colours and large fonts for people with visual problems. Soundcard/speakers may be use to sound out prompts for those who cannot see well. The volume of the speakers may be increase for the hard of hearing. The microphone may be used as an input device for those used who cannot use a keyboard.

3. The owner of a small printing press is looking for a computer system to be used for DTP (Desktop Publishing).

(a) Briefly describe the different software packages, which are required for desktop publishing.

Page layout software
Drawing Package
Scanning Software
Word Processor

- (b)** Besides a keyboard and a VDU, what input and output devices are most suitable for use with a DTP system? Explain the use of each.

Input Devices: Scanner Which will be used to scan images directly into the magazine
Digital Camera to be able to shoot photos and download them into the PC

Output Devices: High Resolution Printer in order to print out the master copy for the
printers

- (c)** Explain why it is best to choose the software before deciding on the hardware.

The hardware specification depends very much on the software that you are going to use. There is no need for extremely high-end specifications if the software that you are going to use does not require it. On the other hand, if the hardware is purchased first and then the user finds out that the software does not run on the specifications, he might have to upgrade the software, or else change the software to an inferior package.

- (d)** Explain why businessmen, such as the printing press owner, may find it necessary to computerise their business.

It would automate many of the manual processes, such as account keeping, chasing debtors, and preparing draft printouts

- 3.** A club secretary a word processing package in order to prepare circulars for the club's members. Apart from using the application to write out the circular, she uses it to personalise each circular for each member by putting the member's name, address and membership number on the circular.

- (a)** What features of the word processor make it ideal to prepare the circular
Spelling checker, Page layout automation, Mail merge

- (b)** Describe how the clerk uses the word processor to personalise each circular.
The clerk associates the circular to a database containing the name, address and membership number of each member. When typing out the letter, instead of typing in each members' name, the clerk keys in the relative fields in the database. When the document is ready for printing, the mail merge feature is activated, and a personalised copy of the letter is prepared for each member.

- (c)** What type of hardware is required for the clerk to work efficiently?
Apart from the computer, which would not require a high specification, the clerk would need a laser printer if there were a large number of members.

Paper 2 B Questions

1. Many youth clubs periodically produce illustrated newsletters for their members. Nowadays, it is possible to prepare and print such newsletters even on cheap computer systems.

- (a) What kind of software would typically be used for preparing the newsletter?

A Desktop Publisher would typically be used, but nowadays, since word processors are becoming more and more advanced, one might consider using a top end word processing package.

- (b) What kind of software would be used for preparing the illustrations?

A drawing package will be used. Although both the Desktop Publisher and the Word Processing Package do have some sort of graphic tools, these do not really allow you to edit the pictures.

- (c) Why is a laser printer more suitable than a dot matrix printer for printing a newsletter?

The quality of the laser printer is far superior to that of a dot matrix printer, and since the printing process inevitably loses some of the quality, the master copy should be as sharp as possible.

- (d) Briefly explain why a SCANNER would be useful to somebody who is preparing the newsletter?

The scanner is useful in order to capture graphics into the magazine without having to redraw them on the computer.

- (e) Describe TWO other uses a youth club might find for a computer, besides that of preparing the newsletter.

Two other uses would be to maintain the membership records and to keep the accounts of the club.

3. Word processing has become a very popular computer application. Write brief notes about each of the following common word-processing features:

- (a) Text formatting

Text formatting is the ability to enhance the appearance of the actual text characters, for example by changing the font type (Times Roman, Courier), size (8pt, 12pt) and style (Italics, Bold)

- (b) Text Justification.

This type of formatting works on the way a paragraph is aligned to the margins. Left justification means that the left of the paragraph is aligned to the margin with the right normally ending up jagged. Right justification is the same but with respect to the right. Full justification is when both the left and the right of the paragraph is aligned to the margins. This results in the spacing between the words being of variable length. Centre justification is when the lines of the paragraph are centred between the margins.

(c) Spell Checking

The facility to check the spelling of the document. Words which do not feature in the dictionary are highlighted and similar words offered as replacements.

(d) Print Preview

The facility to view how the document will appear when printed. The amount of magnification can allow part of a page to a number of pages to be viewed at one go.

(e) Cutting and Pasting

These options allow the user to delete text or graphics from one part of the document and place them in another part.

Section 2 Chapter 3

PRACTICE EXERCISES – 3

Paper 1 Questions

1. Three types of software that may be used to convert source code to object code are Assemblers, Compilers and Interpreters.
 - (a) What is the difference between source code and object code?
Source code is the code written by the programmer and object code is that code that is generated by the compiler. [2]
 - (b) What is the difference between an Assembler and a Compiler?
An assembler converts a low level source code program to machine code whilst a compiler converts a high level source code to machine code. [2]
 - (c) What is the difference between a Compiler and an Interpreter?
A compiler generates a machine code executable program whilst an interpreter translates the source code each time the program is executed. [2]
 - (d) Why do programmers prefer High Level Languages to Low Level Languages?
High level languages are easier to learn as they are not as machine oriented as low level languages are. [2]
 - (c) Why do programmers still feel the need to use Low Level programming Languages?
Low level languages offer more control on the actual hardware of the computer, therefore there may be situations where it is needed. [2]

Section 2 Chapter 4

PRACTICE EXERCISES – 4

Section 1 Questions

1. (a) Explain the terms:

WAN. Short for Wide Area Network. A network of computers spread over a large area.

LAN. Short for Local Area Network. A network of computers spread over a small area

- (b) A school has 3 offices. All computers in the offices are connected to each other on the schools LAN. What advantages are there by connecting the PCs together rather than having set up as standalone PCs.

First advantage Easier to share documents

Second advantage Can communicate via email

Third advantage Can share devices such as hard disks, laser printers, CD rewriters etc

2. All computers require an Operating System in order to function.

- (a) What do you understand by the term Operating System?

The operating system is that software that interfaces between the user or the application and the actual hardware. It allows the easy operation of hardware devices. This software also has a number of disk utilities such as formatting, scanning and defragmenting.

- (b) i. What is **Batch Processing**?

Is the process of first collecting the input data, processing it and finally extracting the output.

- ii. Mention ONE application of batch processing.

Point of Sale/Stock control in a retail outlet.

- (c) i. What is a **Real-Time Operating System**?

An operating system that facilitates the immediate updating of records.

- ii. Describe briefly ONE situation where a real-time operating system is required.

A flight booking application

Section 2 Questions

1. Although the Batch Processing system is a very old way of processing, many modern companies today still use the technology. There are other companies (however) that have no option but to go for a real time system. Discuss.

The technology, although old, is still valid today. Many applications, such as stock control, payroll etc do not really require real time processing. Since real time processing is more expensive (hardware and software wise) companies only go for real time systems if they really require it.

2. Write Short Notes on the following:

(i) Spooler

The spooler is that part of the operating system that is responsible for transferring data that has been stored in a spool file to the printer.

(ii) Queuing

Queuing is the practice of storing files prior to an operation (eg printing). This way applications do not have to wait for the previous job to finish the operation prior to being released to the user.

(iii) Time Sharing

Is the process of allocating various terminals to a processor. Each terminal gets a few split seconds of attention prior to the “attention span” being moved onto the next terminal. This way all the terminals perceive to have the attention of the CPU at all times.

(iv) WAN

A Wide Area Network is created when a number of computers are linked together over a large distance (even globally). These are linked together in order to share devices, software, data or simply to allow communication between the users.

(v) LAN

A Local Area Network is similar to a Wide Area Network, but the PCs are located within a short distance of each other. For example within one building, or office.

(vi) GUI

A Graphical User Interface is used to describe the front end of an application. A GUI normally interfaces with the user via icons and/or drop down menus. The application is therefore normally very user friendly as the user can operate it simply by selecting options on the screen.

(vii) CLI

Command Line Interface is also used to describe the front end of an application. This type of front end is not so user friendly as the user normally has no indication of what to do on the screen. Instead the user interfaces with the software by issuing commands at the terminal by typing them in.

Section 2 Chapter 5

PRACTICE EXERCISES – 5

Section 1 Questions

1. Briefly explain what is meant by each of the following two data access methods, and for each suggest one application for which the method may be used, giving a reason for your choice of application.

Serial Data Access: With this mode of access, to retrieve a record, all the previous records must be read. That is to read record 100; the previous 99 records must first be read.

Application: An application for this mode of access would be a backup operation

Reason: When you would want to restore the file, the whole file must be restored, and therefore, the entire file would have to be processed anyway.

Random Data Access: This mode of access allows us to directly access any record. That is to access record 100, we can go directly to it.

Application: A telephone directory system

Reason: Since in most cases you would only want one telephone number or address, it would be extremely time consuming to go through the whole database each time you need a number

Section 2 Question

1. A video shop has commissioned a membership application from a software house. The owner was told that there will be two main files used, a members file and videos file.

- (a) What fields would you expect to find in the members' file (data type and field length must be specified).

Membership Number	Integer	
Member Name	Text	25
Member Surname	Text	25
House Name	Text	25
Street Name	Text	25
City	Text	25
Telephone Number	Text	16
ID Card	Text	9
Books currently on Loan	Integer	
Date of Birth	Date	
Date of Membership	Date	

- (b) What fields would you expect to find in the books' file (data type and field length must be specified).

Book Code	Integer	
ISBN Number	Text	20
Shelf No	Text	2
Title	Text	30
Author	Text	
Date of Publication	Date	
Available	Text	1

- (c) The system analyst also informed the owner that a loans file will be needed. What do you think

the loans file will be used for.

The Loans file would contain the details of the books loaned out to members.

- (d) What fields would you expect to find in the loans' file (data type and field length must be specified).

Loan No	Long Integer
Book Code	Integer
Membership Number	Integer
Date Loaned Out	Date
Date Expected Back	Date

- (e) What is a key field and why is it required?

A key field is a field that would contain a unique value for every record in the file.

- (f) Mention the key field in each of the three files described above.

Member File:	Membership Number
Book File	Book Code
Loan File	Loan No.

Section 3 Chapter 1

Generic

1. The major steps of the system life cycle:

- i) Detailed definition of the problem
- ii) A feasibility study
- iii) Collecting information about the proposed system
- iv) Analysis
- v) Design of the system
- vi) Programming
- vii) Implementation
- viii) Maintenance

2. a) **Feasibility study.** The initial suggestions must be examined in some detail to check if a computer system is practical and feasible. The analyst has to determine the input format and where the data is gathered; the output required and the method of deriving the output from the input. The job has to be looked at as a whole because any oversights may be costly to rectify at a later stage. An estimate of the cost of the whole project is also indicated so that management would be in a better position to decide in favour or against continuing with the project.

b) Test data is designed to test the valid conditions, to test the error conditions and to check for volume handling.

Paper 1

1. **Systems analysis** is the process of analysing particular systems to see if **computerisation** would be useful, **productive** and more profitable. It also includes the specifying, setting up, **testing** and evaluating of the system. The latter is done to see if the new system has achieved the desired aims and **objectives** that were laid down when the **specifications** for the **system** were drawn up. These procedures have been split up into more manageable sections that together are called the **system life cycle**.

2. Tasks that are carried out during particular stages:

a) Establish the security and recovery procedures	<i>Detailed Design</i>
b) Trap invalid data	<i>Testing</i>
c) Estimate the approximate cost of the hardware	<i>Feasibility study</i>
d) Produce specifications for each module in the system	<i>Detailed Design</i>
e) Formal practice on the job	<i>Training</i>

Paper 2B

1. a) The various stages of a system life cycle are shown in the following list. Put the list in order by writing the numbers from 1 to 8 in front of each stage as shown:

8	Maintenance
7	Implementation
6	Programming
1	Very detailed definition of the problem
3	Collecting information about the proposed system
4	Analysis
5	Design of the system
2	A feasibility study

- b) Four aims of the testing stage:

- the program modules work together correctly
- invalid data is trapped
- the expected results are obtained
- abnormally large volumes of data are acceptable

- c) Types of test data are designed to test the valid conditions, to test the error conditions and to check for volume handling.

2. **Parallel implementation**

The computerised system is implemented while the manual system is still running. Its advantage is that if the computerised system fails, the manual system is still working as usual. The disadvantage is that for some time there is an amount of duplication of work. At some point in time, the manual system is phased out.

Direct implementation

The manual system stops and the new computerised system is used. The disadvantage of this implementation is that it is hard to revert to the manual system if the computerised system fails. Thus the computerised system has to be very sound.

Pilot run and **Staggered** implementation methods are also used.

Paper 2A

1. Mention 4 methods of implementation and describe each one of them.

Parallel implementation

The computerised system is implemented while the manual system is still running. Its advantage is that if the computerised system fails, the manual system is still working as usual. The disadvantage is that for some time there is redundancy of work. At some point in time, the manual system is phased out.

Direct implementation

The manual system stops and then starts the implementation of the computerised system. The disadvantage of this implementation is that it is hard to revert to the manual system if the computerised system fails. Thus the computerised system has to be very sound.

Pilot run

A pilot run is used to test thoroughly a system and gather information about the success of the computerised system before it is implemented. This is aimed to minimise errors when a system is implemented after the manual system is phased out.

Staggered implementation

This way of implementing a system is another way of minimising failure. Various sub-systems are identified and these are implemented one by one (not all at once).

2. Systems are created to solve problems. Once a problem is defined, a system is developed to try to solve it. Identify and describe the main stages of a system life cycle.

Definition of the Problem. A system is created to solve a problem. Once the problem is defined, one can analyse whether it can be solved or not. If it can be solved, then the next stage follows.

Feasibility Study. The initial suggestions must be examined in some detail to check if a computer system is practical and feasible. The analyst has to determine the input format and where the data is gathered; the output required and the method of deriving the output from the input. The job has to be looked at as a whole because any oversights may be costly at a later stage. An estimate of the cost of the whole project is also indicated so that management would be in a better position to decide in favour or against continuing with the project.

Project Approval. The original proposal is presented to management for approval, before which, work on the system is not started.

Detailed Design Stage. After approval, the systems analyst starts to:

- draw the system flowchart to show the flow of information
- specify the file organisation, record length and media
- the list of programs which access which files
- describe the input and output formats and contents
- produce a specification for each program in the system
- list general routines and standards for use within the system
- establish the security and recovery procedures
- produce time-scale and budget information for the remaining work

Testing. The aims of systems testing are to ensure that:

- the programs work together correctly
- the controls agree
- invalid data is trapped
- security and recovery techniques are safe
- the expected results are obtained
- obscure points are catered for
- abnormally large volumes of data are acceptable

Training. People have to be trained to use the new system to use it more effectively. Personal explanation is far better than studying a user's manual because the user can ask questions when in doubt. It is up to the IT Trainer to answer any queries and reassure the users so that they gain confidence in the system. A formal presentation and on the job training is required.

Documentation. Normally there will be three levels of documentation - the **operator's manual**, the **program documentation** and the **user's manual**.

Implementation. Timing of the implementation must be agreed upon by the users. At this stage, testing, documentation and training of users should have been completed. There are two attitudes to implementation. Either **direct** implementation or working in **parallel** for a specified period of time. Sometimes implementation on very large systems is **staggered** over a longer period of time.

Section 3 Chapter 2

Generic

1. **Program Documentation:** This documentation defines all variables and how the program works. In this way it makes it easier to modify and maintain the program.
User Documentation: This documentation provides the normal running information together with typical examples of input and the system's response (output). It also explains the validation techniques and the error messages issued.
2. The program documentation includes
 - i) List of variables used, their type and purpose
 - ii) Explanation of what the next group of instructions is to achieve.

The user documentation includes

- i) instructions how to install the program and how to use it usually illustrated with screen dumps of typical inputs and outputs.
- ii) Help on error messages.

Paper 2B

1.
 - a) **In-line documentation** is “comment” statements written in a spoken language within the program instructions.
 - b) We can do this in Pascal by writing comments in English within curly brackets. For example:

```
Tot:=0; {Initialisation of the variable Tot}
```
 - c) The in-line documentation should include:
 - **General information:** author, date, version, target computer, system requirements, language used.
 - A **list of variables** used, their type and purpose.
 - An **explanation** of what the next group of instructions is to achieve.
2. A well-documented program should have a user manual.
 - a) Identify three sections that could be found in this manual and describe them.
 1. **Overview.** This section goes through all the options in the program explaining what they are for and how to carry out certain tasks.
 2. **Tutorial.** Here a case is described and the user is “demonstrated” step by step how to carry out the scenario on the system.
 3. **Troubleshooting.** This section lists all the possible error messages that can be displayed on the screen, how they occurred and how to rectify it.

- b) User manual is also referred to as a non-technical manual. On the other hand, an operator's manual is technical in nature. In what environment should an operator's manual be used?

Unlike systems working on personal computers, multi-user systems are not run by the user himself. The services of an operator are necessary in multi-user environment and system developers compile a manual with the necessary instruction and guidelines to monitor the progress of a job.

- b) What is the main purpose of an operator's manual?

It is used as a permanent reference document to inform the computer operator of the system to be implemented, the work to be done in its routine operation, and any special features.

The contents should be clear and the requirements of operations by sections should be illustrated (e.g. data capture, data control, data preparation, operations, etc.) It should be designed to enable problems of operation to be solved without continual reference to programmers or systems analysts.

Paper 1

1. The sections *Program documentation*, *User manual* and *Operator manual* were included as follows:

A guide to variable names used in the program	<i>Program documentation</i>
A listing of all programs making up the system	<i>Program documentation</i>
Suitable test data showing the corresponding output.	<i>Program documentation</i>
A detailed explanation in layman terms of what the system can do.	<i>User manual</i>
Indication of what is the cause of an error message.	<i>Operator manual</i>
List of peripheral devices involved in the system and their consumables.	<i>Operator manual</i>

2. In-line documentation makes code more **readable** and thus easier to **maintain**. In-line documentation is achieved by **inserting** “comment” statements within the **program** instructions. We can do this in Pascal by writing **comments** in English within curly brackets. The in-line documentation should include **general information**, a **list of variables** used, their type and their purpose, and an **explanation** of what the next group of **instructions** is to achieve.

Chapter 3. Error and Validation Techniques

Paper 2B

- 1 (a) **Validation** techniques involve small program routines that ensure that data being input is correct. **Verification** involves the input of data twice by different data entry clerks. The system compares the last input with the first and alerts the clerk if there is a mismatch.
 - (b) Range check on the date:
 - Cannot be less than 1
 - If the month is Jan / Mar / May / Jul / Aug / Oct / Dec then max date allowed is 31.
 - If the month is Apr / Jun / Sep / Nov then max date allowed is 30
 - If the month is Feb then if the Year is LEAP, then max date allowed is 29 else 28.Range check on month:
 - Has to be one from {Jan, Feb, Mar, Dec}Check for Leap Year
 - If $\text{year mod } 4 = 0$ and $\text{year mod } 400 \neq 0$ then it is a leap year.
 - (c) **Check digits** are complex validation systems which ensure that a key data item is valid (modulo 11) . A **range check** validates the input by comparing it to a minimum and maximum value allowed. Can be used while entering exam results: results less than 0 or greater than 100 are not allowed.
2. (a) A **syntax error** is an error in the use of the programming language reserved words or structure. In Pascal, typing *writlen* instead of *writeln* generates an error.
 - (b) A **logic error** is an error in the logic of the program. Calling a subroutine before another can cause this error which is difficult to trap. Another example is typing a formula wrongly which still generates a result but not the correct one. (Area = Length + Width instead of Area = Length * Width)
 - (c) A **run-time error** occurs while a program is running. May be caused by trying to access data from a floppy disk and the disk is not there. Another one is the division by zero error which can be caused by invalid data input.

Paper 1

1. a) Transposition b) Substitution c) Omission.
2. a. **Reason 1** – The program should be tested to check whether it works or not.
Sometimes a program can stop and display an error message.
Reason 2 – The program should be tested to find out whether it does what it was intended to do.
- b. A run-time error.

Division by zero error. This may occur when trying to find the average of a set of numbers and the number of marks entered is 0. The program tries to divide the total by 0 and this causes a runtime error.

Test data can be used to check that correct results are output by the program. A range of input values should be used to correct this run-time error.

- (a) **Syntax Errors:** These errors are simple errors in the use of the language reserved words and structure. Each language has its own grammar and a programmer may use the wrong construction. This would be identified during the translation of the source code into object code. Compilers and Interpreters have the capability of checking for and identifying syntax errors. Considering some examples in Pascal:

	Correct Syntax	Wrong Syntax
a)	WRITELN;	WRITENL;
b)	Z := X + Y;	X + Y := Z;
c)	T := X * Y - 3;	T = X x Y - 3;

Logic Errors: These are errors in the logic of the program. It is not possible for the computer to detect the presence of a logical error.

Example 1: The correct formula to calculate the area of a rectangle is: $AREA := L * B$;
But if instead " * " one types " + ", the result given is not the area, but the total of L and B.

Example 2: Another common error that yields a logic error is that of forgetting to initialise a variable. In the case where the sum of a list of numbers is to be found, the formula $TOTAL := TOTAL + X$ can be used. But, if TOTAL is not initialised to the value 0, then the final output is incorrect.

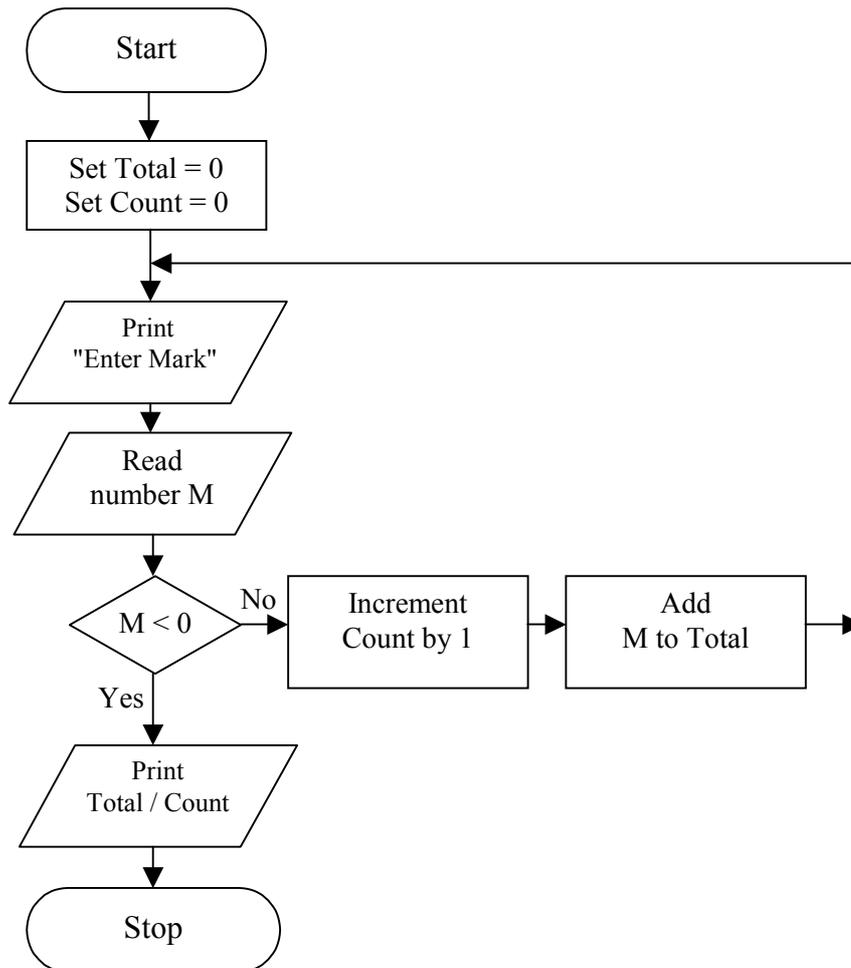
The first line of defence against logical errors should be the programmer sitting down and checking through the logic of the program. It is often better if a programmer other than the original author of the program performs this task. It involves sitting down with a pencil and paper and going laboriously through each stage writing intermediate results. This process is often called a **dry run**. If a structured programming technique has been adopted then it is relatively easy to test each module independently.

- (b) **Debuggers** are special software that helps you to debug programs. They often allow you to single-step through a program and see the effects on each variable. Some logical errors are very difficult to find and trace facilities offered by debuggers indicate the exact path taken by the computer. This may be significantly different from the path that the programmer originally intended, and therefore the source of the error is indicated.
- (c) **Data validation** is a process where data is checked according to pre-established rules before it is allowed to be entered. If the user is asked to type in the age of a student then only positive numbers (perhaps up to a maximum of 99) ought to be allowed. Similarly no punctuation should occur in this age field.
- (d) Although equipped with a visual display unit, data entry clerks still input data blindly. Most probably due to extensive usage, they will input data without visually checking the validity. To minimise errors, the **data verification** system is sometimes applied. This system involves the inputting of data twice by different data entry clerks. **Data validation** involves a process where data is checked before it is processed, including range checks, and use of check digits.

Section 3 Chapter 4

Paper 1

1. a) Correct, Wrong, Wrong, Correct



Paper 2A

1. a) 3, 1, 4, 4, 5.
- b) Finds the highest common factor of the two numbers entered
- c) M takes the value of N and N takes the value of M
- d) Run-time error (Division by zero error)

2. a) A **flowchart** is a graphical representation of the operations involved in a data processing system. Symbols are used to represent particular operations or data, and flow lines indicate the sequence of operation or the flow of data. **Pseudo code** is writing an algorithm as close as possible to the programming language, but without taking the extra effort to ensure that the syntax is correct.
- b) An **integer** is a whole number that can be negative or positive. A **real** is a number that can represent fractions.
- c) 20 marks
- d) Increment means increase the value by 1
- e)

```

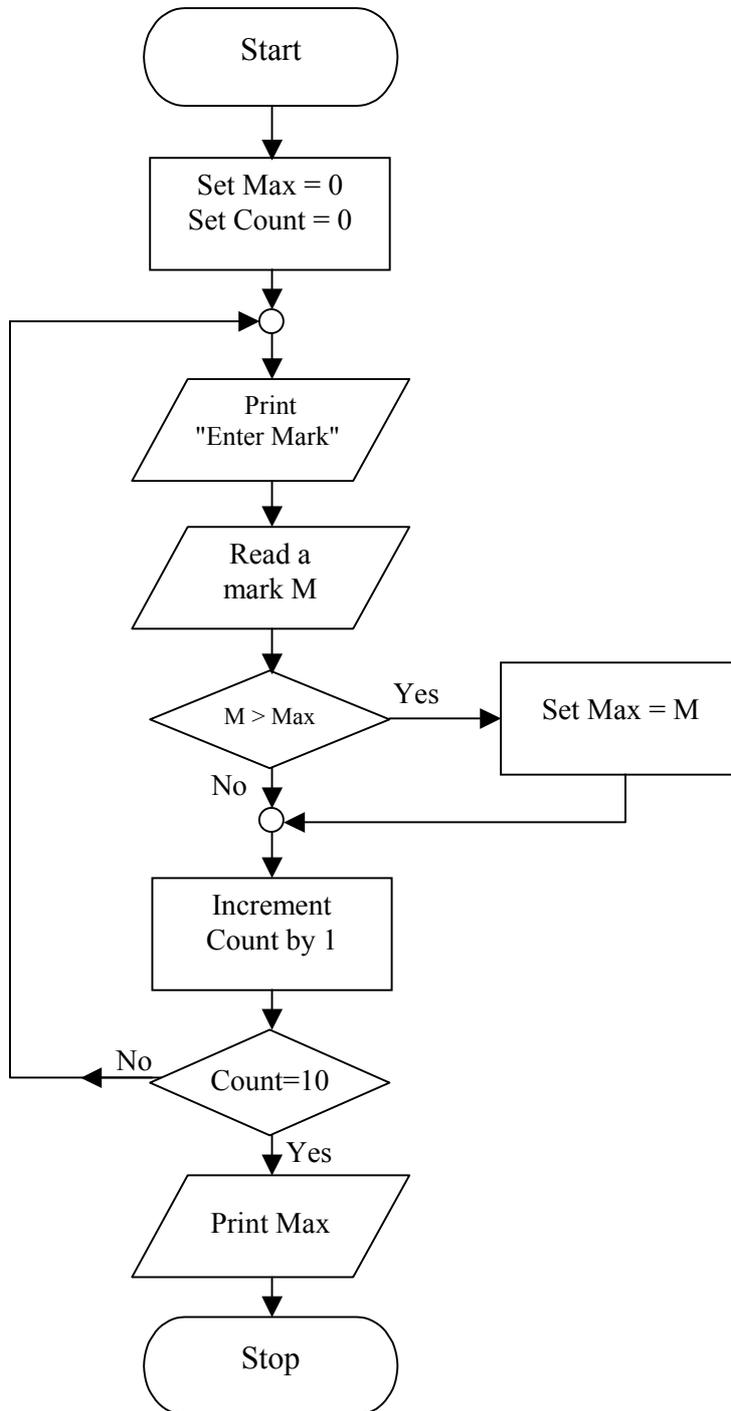
Start
F ← 0
P ← 0
Repeat
    Enter M
    If M < 50 then Increment F else Increment
P
Until (F+P) = 20
Output F and P

```

Paper 2B

1. a) An algorithm is a finite set of rules giving a sequence of operations for solving a specific type of problem.
- b) It calculates and prints the average of 10 numbers.
- c) The flowchart symbol in step 3 represents an input,
- d) The one in step 7 represents an output.
- e) A conditional jump
- f) Total, Count, N.
- g) Steps 1 and 2
- h) Steps 3, 4, 5, 6.

The flowchart representing the algorithm to find the highest mark out of the 10 entered marks.



Section 3 Chapter 5

Practice Exercise 1

1.	<table border="1"><tr><td>2</td><td>10</td></tr><tr><td>7</td><td>111</td></tr><tr><td>8</td><td>1000</td></tr><tr><td>16</td><td>10000</td></tr></table>	2	10	7	111	8	1000	16	10000	<table border="1"><tr><td>25</td><td>11001</td></tr><tr><td>27</td><td>11011</td></tr><tr><td>40</td><td>101000</td></tr><tr><td>100</td><td>1100100</td></tr></table>	25	11001	27	11011	40	101000	100	1100100
2	10																	
7	111																	
8	1000																	
16	10000																	
25	11001																	
27	11011																	
40	101000																	
100	1100100																	

2.	<table border="1"><tr><td>11</td><td>3</td></tr><tr><td>101</td><td>5</td></tr><tr><td>110</td><td>6</td></tr><tr><td>1111</td><td>15</td></tr><tr><td>11000</td><td>24</td></tr></table>	11	3	101	5	110	6	1111	15	11000	24
11	3										
101	5										
110	6										
1111	15										
11000	24										

3.	(i) 36, 214, 506	(ii) 25, 43, DC
----	------------------	-----------------

4.	(a) $90_{10} = 1011010_2$	(b) $00011010_2 = 1A_{16}$	(c) $64_{10} = 40_{16}$
----	---------------------------	----------------------------	-------------------------

Practice Exercise 2

- $67 \text{ bin} = 1000011 \text{ bin}$
 - $CE9A \text{ hex} = 1100111010011010 \text{ bin}$
 - $327 \text{ dec} = 147 \text{ hex}$
 - $10001111 \text{ bin} = 143 \text{ dec}$
 - $10010011100 \text{ bin} = 49C \text{ hex}$
 - $CA \text{ hex} = 202 \text{ dec}$
- binary digit, either 1 or 0
 - a byte is a group of 8 bits
 - ASCII code: a standard code for information interchange where each character is assigned a code which is standard throughout IBM compatible machines
- There are 8 bits in a byte
 - 0 to 255
 - is not enough to fit 48. $2^6 = 64$ is enough. Hence, the least number of bits is 6.
- Six bits

Practice Exercise 3

1. $195_{10} = 11000011_2$ in 8-bit register

2. (i) 1000000

(ii) 100101

(iii) 111001

3. (i) 100011

(ii) 10000111

(iii) 1011010

4.

	Number	1's Comp	2's Comp
i)	1100	0011	0100
ii)	0001	1110	1111
iii)	111111	000000	000001
iv)	101010	010101	010110
v)	10011110	01100001	01100010
vi)	0011011111	1100100000	1100100001

- 5.
- | | |
|-------------|-----------|
| i) 1001 | ii) 10100 |
| iii) 110111 | iv) 11001 |
| v) 111011 | |

Practice Exercise 4

1. SAM: (i) -13 (ii) -65 (iii) +127
 One's Comp: (i) -43 (ii) +117 (iii) -0
 Two's Comp: (i) -15 (ii) -118 (iii) +99

2. (i) (a) 10101101 (b) 11010010 (c) 11010011
 (ii) (a) -53 (b) +106 (c) -60

3. Unsigned: 0 to 255
 Signed: SAM and One's Complement: -127 to 127
 Two's Complement: -128 to 127

Paper 1

1. (a) A=1010101, B=1000101
 Two's complement of B = 0111011

$$\begin{array}{r}
 A = \quad 1010101 \quad + \\
 -B = \quad 0111011 \\
 \hline
 C = \quad \mathbf{1} \ 0010000 \quad C = 0010000 = 10000
 \end{array}$$

- (b) A = 85, B = 59, C = 16 which is equal to 85-59

2. We need 62 unique patterns in all. The minimum number of bits is 6 since $2^6 = 64$.

3. Let A = 01101000, Let B = 00100100. The Two's complement of B is 11011100.
 Hence A - B = 01101000 + 11011100 = **01000100**.

$$\begin{array}{r}
 01101000 \quad + \\
 11011100 \\
 \hline
 \mathbf{1} \ 01000100
 \end{array}$$

4. (a) 4 bits are available for functions. So there are $2^4 = 16$ functions
 (b) 8 bits for address. So there are $2^8 = 256$ directly addressable locations.

5. (a) The largest 8-bit binary number in two's complement is 01111111
 (b) +127

(c) Let $A=37_{10}$ and let $B=121_{10}$

Converting 37_{10} in binary we get $A = 00100101_2$

Converting 121_{10} in binary we get $B = 01111001_2$

Two's complement of $B = 10000111$

Let $C = A - B$

$$\begin{array}{r} 00100101 \quad + \\ \underline{10000111} \\ 10101100 \end{array} \quad C = 10101100$$

C is a negative number so we must find the two's complement: $01010100 = 84_{10}$

Therefore $C = -84_{10}$

7. Complete each row of the table to show the same number in binary, hexadecimal and decimal.

Binary	Hexadecimal	Decimal
1 0 1 1 0 1 0 1	B 5	181
1 1 1 1 1 0 0 0	F 8	248
0 1 0 0 1 0 1 0	4 A	74

8. Positive numbers are represented as 8-bit quantities, hence $2^8 = 256$ unique codes.

(a) The biggest number that can be stored in this way is 255

A **Numeric Overflow error** in this computer can occur if a number greater than 255 is to be stored. For example adding 200 with 100 gives 300 which cannot be stored since 300 in binary is the number 100101100 (it has more than 8 bits)

Paper 2A

1. (a) A 16-bit computer uses a 20-bit address register.

2.

(i) size of the addressable memory is $2^{20} = 2^{10} \times 2^{10} = 1024 \times 1024$ unique locations

(ii) this computer has a 16-bit wordlength

(iii) Four bytes = Two 16-bit words. This computer needs 2 memory cycles

(b) A character code uses 7 bits for the character and 1 bit for the parity

(i) To store a single character we 8 bits are required.

(ii) The purpose of the parity bit is to check that the data was transmitted and received correctly. The parity bit is calculated from the data bits and transmitted together with them. When the data is received, the parity bit is again calculated and compared with the one received. If there is a mismatch an error occurs. An odd parity bit is set to 1 when the total number of high bits in the data code is not odd.

(iii) 7 bits are used for the character itself. $2^7 = 128$ different characters can be coded.

(iv) A printing character can be seen on a monitor or can be printed such as all the letters of the alphabet, numeric characters, punctuation characters, even the space. A control character is meant to be used to controls a process, like the enter character, Escape character, delete character, cursor keys which can control the movement of a cursor on the screen.

(v) The word COMPUTERS has 9 characters. A 16-bit computer uses 5 words

1	C	O
2	M	P
3	U	T
4	E	R
5	S	

3. (a) (i) Sign and Magnitude $10011010 = -26$

(ii) Two's Complement 10011010 is a negative number. We need to find its two's complement

Two's complement of 10011010 is 01100110 .

$01100110 = 102$

Therefore $10011010 = -102_{10}$

(b) The arithmetic shift left multiplies the number by 2

Section 3 Chapter 6

Practice Exercise

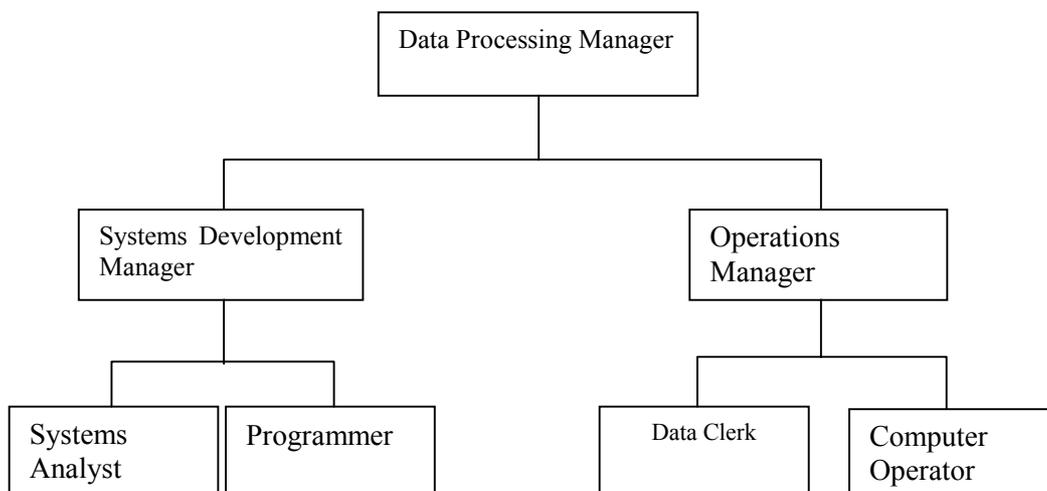
1. Introducing new information systems and planning for future data processing needs
2. (a) Systems Analyst
 - (b) Four major abilities of a systems analyst:
 - Communications Skills
 - Problem solving skills
 - Environment Expertise
 - Technical Knowledge
3. Programmer is required to maintain, existing programs as well as to create new ones.
4. COBOL, C
5. A 4GL is a system of standard templates, which simplify the development of an application and enhance the programmers' productivity.
6. Operators are required to supervise the running of the computer installation and to operate the individual pieces of equipment.
7. A maintenance engineer is called upon for advise when new equipment is to be purchased and performs regular maintenance together with the team of technicians to the computer hardware.
8. The role of a webmaster involves the designing and maintenance of a web site.

Paper 1

Job responsibilities

Systems Analyst	Responsible for the design and implementation of the system
Operations Manager	Manages the resources, deal with other users at the site, standard administration, disaster mitigation, hardware and software upgrading.
Programmer	Responsible for the planning, writing and testing of the programs.
Computer Engineer	Performs regular maintenance to the computer installation's hardware.
Computer Operator	Supervise the running of the computer system installation and to operate the individual pieces of equipment.
Data Processing Manager	To introduce new information systems. To ensure that the data processing needs of the organization are met. Planning for future data processing needs.
Webmaster	To create and manage the information content (words and pictures) and organization of a Web site To manage the computer server and technical programming aspects of a Web site

2.



Paper 2B

The **Data Processing Manager** is the person with overall responsibility for running the department. He/she is responsible for

1. the technical operations (making sure the jobs are being done on time and to the required standards) and
- 2.
3. personnel within the department (recruiting suitable personnel to be programmers etc).

The **Systems development Manager**:

1. delegates jobs to the different members of the team,
2. monitors progress according to predetermined time schedules
3. enforces quality standards and
4. acts as liaison between the team and other interested parties.

The **Operations Manager's** job would be to supervise the day-to-day operations and ensure that the entire backup team to the analysts and programmers works efficiently. He/she should ensure that the right jobs are done in the correct order and make sure that staff are available to get top priority work through the system in time.

Systems Analyst is the person who is responsible for the design and implementation of a computer system.

Programmer is responsible for planning, writing and testing of programs, and in general works plans provided by the systems analysts.

Computer Operator is required to supervise the running of the computer installation and to operate where necessary, the individual pieces of equipment.

Data entry operator is responsible to type in the data correctly.

I.T. Trainer The role of the trainer is to impart new skills in a clear way and to build up user confidence. The trainer must have good communication powers and be able to help the user in his difficulties. The trainer is usually an expert in a particular area and therefore users are appointed to many trainers.

Paper 2A

1. a) A **systems analyst** is the person who is responsible for the design and implementation of a computer system. The systems analyst gives meaning and direction to the development process. The systems analyst must be good at problem solving abilities, be knowledgeable about business operations and have a strong background in relevant areas of computer science.

The **technical support staff** consists from a team of technicians and engineers. They are in charge of regular maintenance to the computer installation's hardware. When new equipment is purchased, they are responsible to install it.

Data entry operators enter the data into a computer system. They are usually skilled touch typists.

- b) The technician may check whether the printer is online, whether the printer is loaded with paper and that the connection cable is connected properly.
- c) The system analyst reviews periodically the computerised systems of the company to analyse the throughput of the system. Bottlenecks can be identified and solutions to make the system better are studied and designed. The system may have errors in some modules, and these can be rectified periodically to minimise mistakes and make the system more reliable. At each review, the systems analyst may decide to maintain the system, or phase it out to implement a new, better system.

Appendix

Questions on Chapter 1 - Systems analysis and design

Generic

1. Identify and list the major steps of the system life cycle
2. What is meant by the following terms:
 - a) Feasibility study
 - b) Test data

Paper 1

1. Fill in the blanks with 8 words from the following table:

<i>objectives</i>	<i>Systems analysis</i>	<i>Feasibility study</i>	<i>specifications</i>	<i>implementation</i>
<i>System</i>	<i>system life cycle</i>	<i>productive</i>	<i>testing</i>	<i>computerisation</i>

..... is the process of analysing particular systems to see if would be useful, and more profitable. It also includes the specifying, setting up, and evaluating of the system. The latter is done to see if the new system has achieved the desired aims and that were laid down when the for the were drawn up. These procedures have been split up into more manageable sections that together are called the

2. In which stage the following tasks are carried out during a systems analysis:

Feasibility study
Training

Testing
Implementation

Detailed Design
Problem Definition

a) Establish the security and recovery procedures	
b) Trap invalid data	
c) Estimate the approximate cost of the hardware	
d) Produce specifications for each module in the system	
e) Formal practice on the job	

Paper 2B

1. a) The various stages of a system life cycle are shown in the following list. Put the list in order by writing the numbers from 1 to 8 in front of each stage as shown:

	Maintenance
7	Implementation
	Programming
1	Very detailed definition of the problem
	Collecting information about the proposed system
	Analysis
	Design of the system
	A feasibility study

- b) List four aims of the testing stage
c) Suitable data should be prepared to test various conditions. List three types of test data that you might use to ensure the correct operation of the program.
2. a) Implementation of a system must be carried out carefully by a systems analyst.
b) Describe what is meant by Parallel implementation and Direct implementation
c) Imagine you are a systems analyst, mention two other methods of implementation apart from Parallel implementation and Direct implementation.

Paper 2A

1. Mention 4 methods of implementation and describe each one of them.
2. Systems are created to solve problems. Once a problem is defined, a system is developed to try to solve it. Identify and describe the main stages of a system life cycle.

Questions on Chapter 2 - Documentation

Generic

1. Briefly explain the importance of the program documentation and user documentation (user manual):

Program Documentation

.....

User Documentation

.....

2. Mention two main sections you might expect to find in:

Program Documentation

.....

User Documentation

.....

Paper 2B

1. A programmer finds **in-line documentation** very helpful to understand the program and be able to maintain it.
 - a) What is in-line documentation?
 - b) Give an example of in-line documentation.
 - c) In-line documentation should include relevant information. Identify three different types of in-line remarks that you may include while writing a program.
2. A well documented program should have a user manual.
 - a) Identify three sections that could be found in this manual and describe them.
 - b) User manual is also referred to as a non-technical manual. On the other hand, an operator's manual is technical in nature. In what environment should an operator's manual be used?
 - c) What is the main purpose of an operator's manual?

Paper 1

1. The following sections should be included in either of the following documentation.

Program documentation

User manual

Operator manual

Identify in which documentation/manual you can find the following:

A guide to variable names used in the program	
A listing of all programs making up the system	
Suitable test data showing the corresponding output.	
A detailed explanation in layman terms of what the system can do.	
Indication of what is the cause of an error message.	
List of peripheral devices involved in the system and their consumables.	

2. Fill in the blanks by using the following 9 terms

<i>comments</i>	<i>general information</i>	<i>inserting</i>
<i>readable</i>	<i>maintain</i>	<i>explanation</i>
<i>instructions</i>	<i>list of variables</i>	<i>program</i>

In-line documentation makes code more, and thus easier to In-line documentation is achieved by “comment” statements within the instructions. We can do this in Pascal by writing in English within curly brackets. The in-line documentation should include, a used, their type and their purpose, and an of what the next group of is to achieve.

Questions Section 3 Chapter 3

Paper 2B

1. (a) Explain the terms **validation** and **verification**.
(b) A date can be validated as well as verified. A date is to be input in the form 25 APR 2001 (a two-digit day, followed by a three-letter month, followed by a four-digit year). Describe the validation checks that could be carried out on dates of this form.
(c) Explain, with examples, the terms **check digit** and **range checks**.

2. Programs may contain syntax, logical and execution (run-time) errors. Explain, with an example, the meaning of each.
 - (a) Syntax error
 - (b) Logical error
 - (c) Execution (run-time) error

Paper 1

1. One type of error frequently made when typing data into a computer system is a transcription error. Substitution, omission and transposition are three types of transcription errors. Using the word HORSE state which type of error has been committed for each of the following:
 - (a) HROSE
 - (b) TORSE
 - (c) HORS

2. Once a program is ready it should undergo testing. Give two reasons why testing at this stage is important.

Reason 1

Reason 2

Name a typical programming error, which occurs when a program is running.

Give an example of such an error.

.....
.....
.....

How would you test for such an error?

.....
.....
.....

Paper 2A

1. Bugs in programs, or programming errors, can sometimes be difficult to trace and correct.
 - (a) Using an example of your choice, explain the difference between a **syntax error** and a **logic error**.
 - (b) Most compilers and interpreters nowadays include a utility called a **debugger**. What is a debugger?
 - (c) A well-written program should always perform **data validation**. Explain what is meant by data validation, giving examples.
 - (d) Explain the difference between **data verification** and **data validation**.

Paper 1

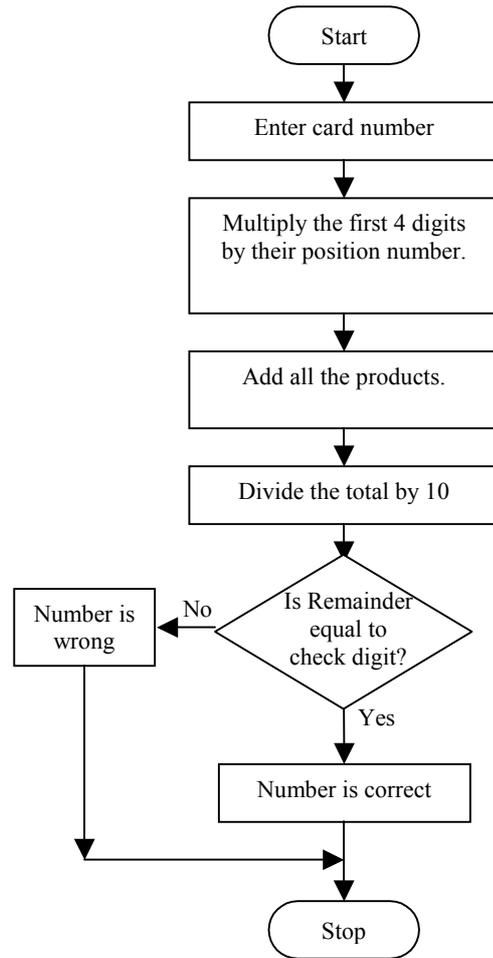
1. A Golf Club system uses membership card numbers consisting of 4 digits, together with a check digit. An example of such a card number 32655 as shown below:

Position	5	4	3	2	Check Digit
Card Number	3	2	6	5	5

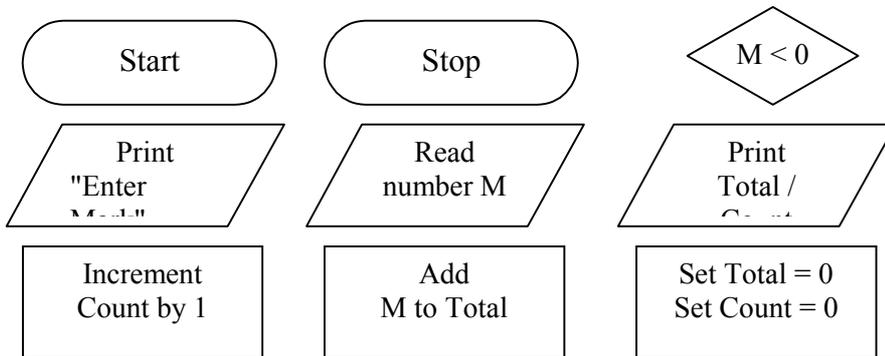
The flowchart describes the algorithm used to verify whether an account number is correct or not.

a. Using the flowchart, determine whether each of these card numbers is correct or wrong:

Account Number	Write CORRECT or WRONG
35464	
93625	
23562	
35649	

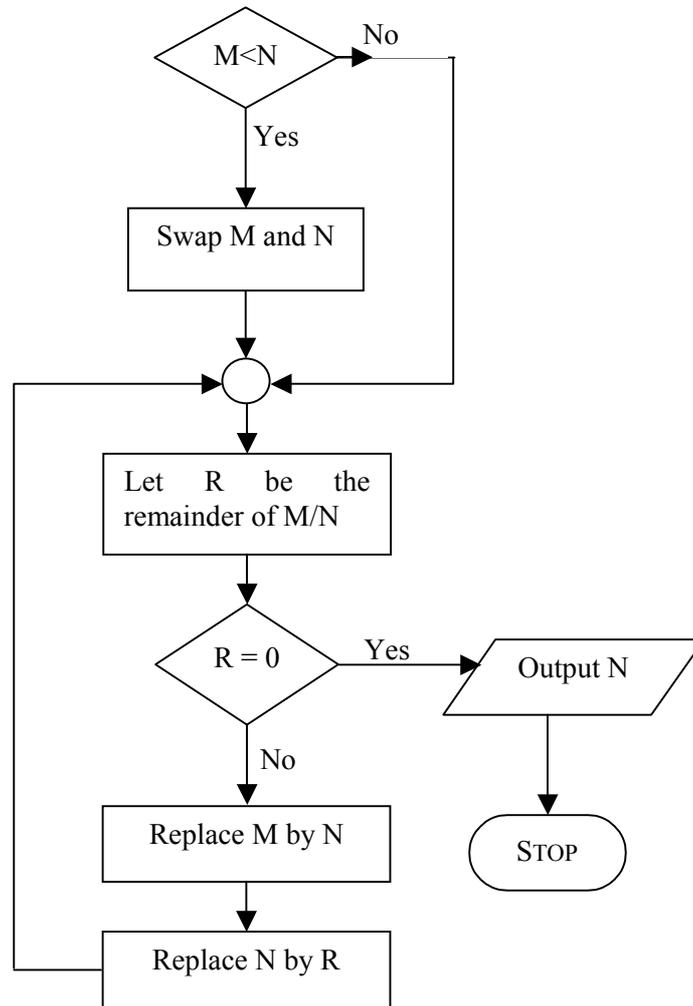


2. Use the flowchart symbols given to draw a program flowchart which lets the user enter a list of marks and then prints their average. The user signals the end of the list by entering a negative number, which is not considered as one of the marks.



Paper 2A

1. The flowchart describes an algorithm that takes two integer numbers, M and N as input, **neither** of which must be 0.



(a) Copy the following table, and fill in the OUTPUT column to show the output of the algorithm for each pair of input values of M and N:

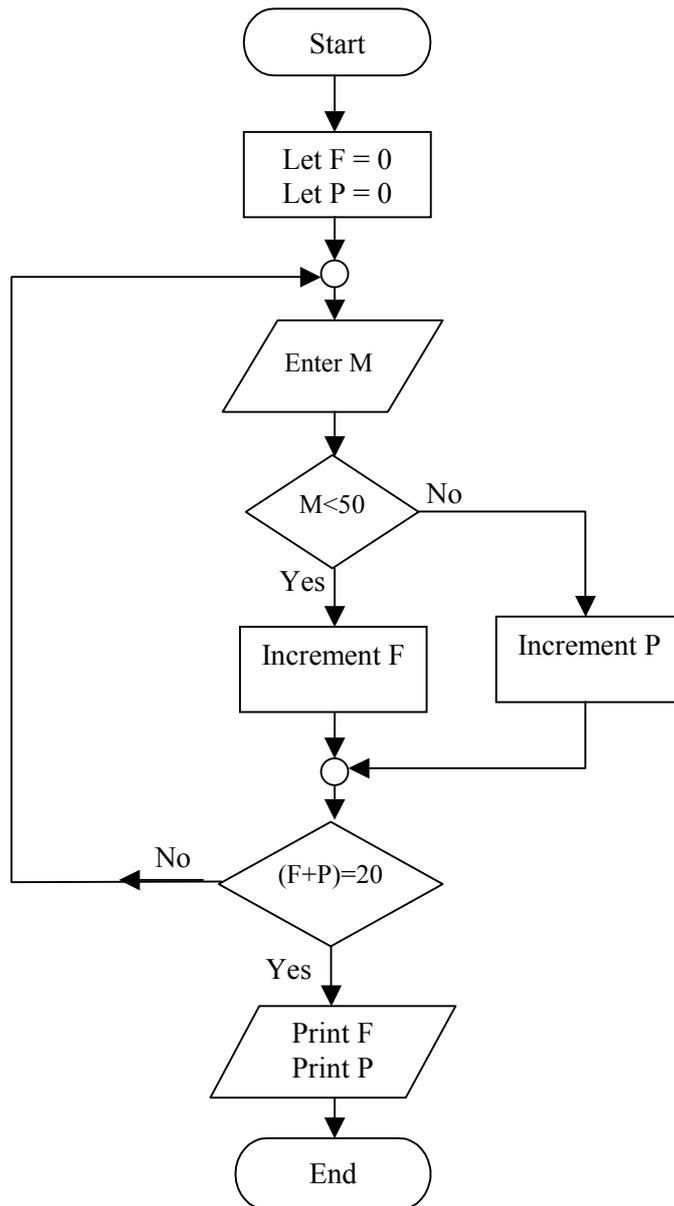
M	N	Output
6	9	
7	3	
8	4	
12	8	
20	15	

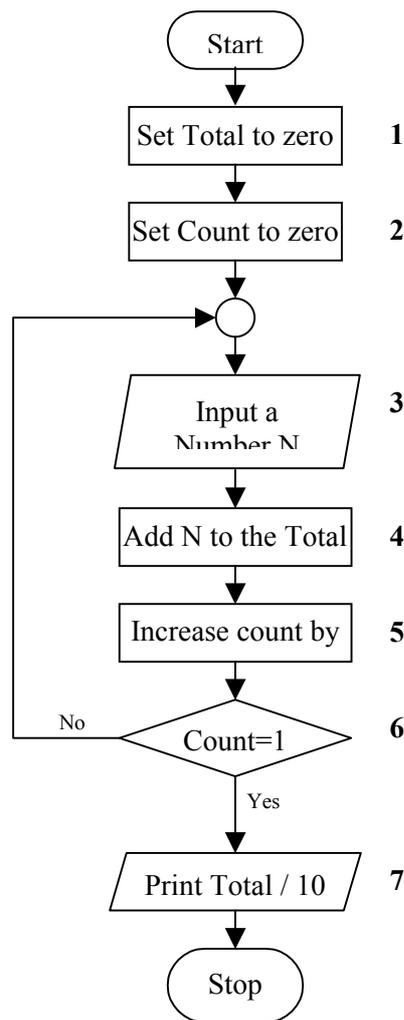
(b) What does this algorithm do?

(c) What is meant by *SWAP M and N*?

(d) The algorithm only works if neither M nor N is 0. What would happen if one of them were 0?

2. The algorithm shown in the flowchart reads in a list of marks and outputs the number of passes (P) and the number of fails (F). The pass mark is set to 50.
- What is the difference between a **flowchart** and **pseudo code**?
 - The variables M, P and F are numeric variables. All of them are of type integer. Explain what is an **integer** and a **real** number.
 - How many marks are read in by the algorithm?
 - What is the meaning of the term **increment** as used in the flowchart.
 - Write a program in pseudo code for the algorithm shown in the flowchart.

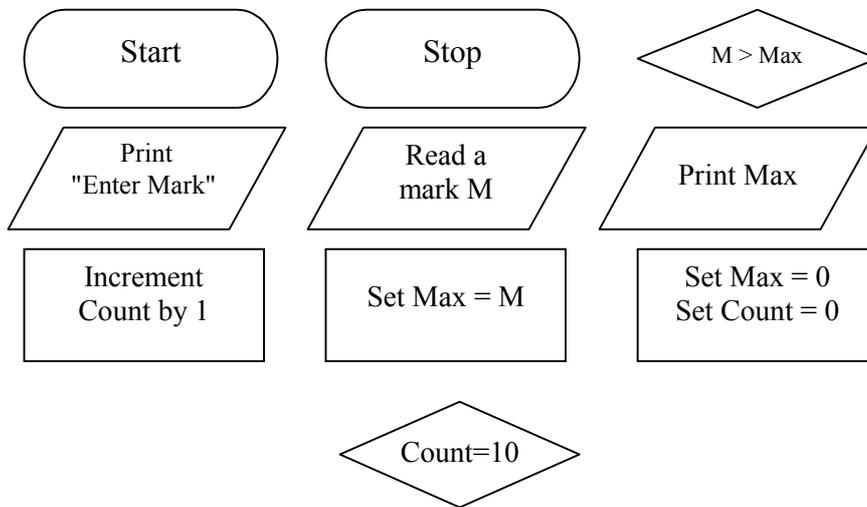




The flowchart above describes a simple algorithm. The main steps have been numbered.

- (a) What is an algorithm?
- (b) What does this algorithm do?
- (c) What does the flowchart symbol used in step 3 represent?
- (d) What does the flowchart symbol used in step 7 represent?
- (e) What does the flowchart symbol used in step 6 represent?
- (f) List all the variables used in this algorithm.
- (g) Which steps perform initialisation?
- (h) Which steps form a loop?

2. Use the flowchart symbols given to draw a program flowchart which lets the user enter a list of 10 marks (between 0 and 100 inclusive) and then prints the highest mark.



Questions - Section 3 - Chapter 5

Generic Questions

PRACTICE EXERCISE 1

1. Express the following decimal numbers in binary:
2, 7, 8, 16, 25, 27, 40, 100. [8]
2. Express the following binary (base 2) numbers in decimal:
11, 101, 110, 1111, 11000. [5]
3. Convert the following as indicated:
 - (a) Hexadecimal to Decimal: 24, D6, 1FA. [3]
 - (b) Decimal to Hexadecimal: 37, 67, 220. [3]
4. Show your working for the following:
 - (a) Convert the decimal number 90 to binary. [2]
 - (b) Convert the binary number 00011010 to Hexadecimal [2]
 - (c) Convert the decimal number 64 to Hexadecimal. [2]

PRACTICE EXERCISE 2

1. Convert the following using the short methods:
 - (a) 6710 to binary
 - (b) CE9A16 to binary
 - (c) 32710 to Hexadecimal
 - (d) 100011112 to decimal
 - (e) 100100111002 to Hexadecimal
 - (f) CA16 to decimal [1×6]
2. Explain the following terms:
 - (a) bit [2]
 - (b) byte..... [2]
 - (c) ASCII code..... [3]
3.
 - (a) How many bits are there in a byte? [1]
 - (b) Complete the following to show what range of positive numbers can be represented in a byte.
The range of positive numbers in a byte is from 0 to _____ [2]
 - (c) What would be the minimum number of bits required to store uniquely in binary a set of 48 characters?
..... [2]
4. A set consisting of the capital letters A to Z (26 characters), the small letters a to z (26 characters), the digits 0 to 9 (10 characters), the decimal point and the space character. What is the minimum number of bits required to represent a character on this computer? [2]

PRACTICE EXERCISE 3

1. Convert the positive decimal integer 195 to binary and store it in an 8-bit register. [2]

--	--	--	--	--	--	--	--

2. All the numbers being binary, calculate the following:

(a) $111011 + 101$

(b) $100 + 100001$

(c) $101010 + 1001 + 110$

[1×3]

3. All the numbers being binary, calculate the following:

(a) 111×101

(b) 1111×1001

(c) 1001×1010

[1×3]

4. Find the One's Complement and the Two's Complement of the following (give your answer according to the size of the given binary number): [6, 6]

Binary Number	One's Complement	Two's Complement
1100		
0001		
111111		
101010		
10011110		
0011011111		

5. All the numbers being binary find the following: [5]

Binary Subtraction	Result
1100 – 11	
11001 – 101	
111101 – 110	
11101 – 100	
111110 – 11	

PRACTICE EXERCISE 4

1. Convert the following into decimal: [3,3,3]

- (a) Sign & Magnitude: (i) 10001101 (ii) 11000001 (iii) 01111111
 (b) One's Complement: (i) 11010100 (ii) 01110101 (iii) 11111111
 (c) Two's Complement: (i) 11110001 (ii) 10001010 (iii) 01100011

2. Three methods of representing Signed Integers exist. Keeping in mind the rules, convert the following as indicated. (1 byte = 8 bits)

- (a) -4510 into binary using:
 (i) Sign & Magnitude
 (ii) One's Complement
 (iii) Two's Complement [2, 2, 2]
 (b) 110010112 (two's complement) into decimal [2]
 (c) 011010102 (sign & magnitude) into decimal [2]
 (d) 110000112 (one's complement) into decimal [2]

3. What range of positive and negative numbers can be stored in an 8-bit word, if signed and unsigned representation is used?

- (a) Range for signed representation (Sign & Magnitude) _____
 (b) Range for signed representation (One's Complement) _____
 (c) Range for signed representation (Two's Complement) _____
 (d) Range for unsigned representation _____

[4]

PAPER 1

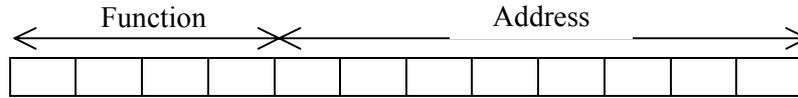
1. Given two binary number A = 1010101 and B = 1000101.

- (a) Use two's complement to find the value of C, given that $C = A - B$. [1]
 (b) Show that your answer is correct by changing the numbers A, B and C to decimal numbers and performing the subtraction. [2]

2. A computer has a character set consisting of the capital letters A to Z (26 characters), the small letters a to z (26 characters) and the digits 0 to 9 (10 characters). What is the minimum number of bits required to represent a character on this computer? [1]

3. Perform the following subtraction by the two's complementation and addition method.
 01101000 – 00100100 [1]

4. The diagram below shows the format of a program instruction for a simple computer with a word length of only 12 bits:



- (a) How many different functions is it possible to have on this computer? Explain. [2]
 (b) How many directly addressable locations are available? Explain. [2]
5. A computer uses 8-bit two's complement numbers.
 (a) In the space below fill in the largest positive binary number that can be represented in this computer. [1]
- (b) Give the value to base 10 of the binary number you wrote above: _____ [1]
 (c) Show how an 8-bit computer using two's complement arithmetic calculates $3710 - 12110$ and get -8410 as an answer. [3]
6. Complete each row of the table to show the same number in binary, hexadecimal and decimal. [2,2,2]

Binary	=	Hexadecimal	=	Decimal
1 0 1 1 0 1 0 1	=		=	
	=	F 8	=	
	=		=	74

7. In a particular computer, positive numbers are represented as 8-bit quantities.
 (a) What is the biggest number that can be stored in this way? _____ [1]
 (b) Give an example of an arithmetic operation that would cause a **Numeric Overflow error** in this computer. [2]

PAPER 2A

1. (a) A 16-bit computer uses a 20-bit address register.
- (i) What is the size of the addressable memory for this computer?
 - (ii) What is the word length for this computer?
 - (iii) How many memory cycles does the CPU need to fetch a four-byte instruction using this computer? [2,2,2]
- (b) A character code uses 7 bits for the actual character and 1 bit for odd parity.
- (i) How many bits are required to store a single character? [1]
 - (ii) What is the purpose of a parity bit? What is meant by odd parity? [2,2]
 - (iii) How many characters can be coded using this character set? [2]
 - (v) How many words does the 16-bit computer use to store the word COMPUTERS using this character code? [2]
2. (a) Consider the 8-bit pattern 10011010. What decimal number does this 8-bit pattern represent if
- (i) It represents a sign and magnitude number
 - (ii) It represents a two's complement number [2,2]
- (b) Assuming sign and magnitude representation, an arithmetic shift left (ASL) operation moves each bit, except the sign bit, one position to the left. Hence
10011010 after an arithmetic shift left becomes
10110100
- What arithmetic operation does an arithmetic shift left do to the number? [2]

Questions Section 3 - Chapter 6

Generic Questions

1. List two responsibilities of a data processing manager
2. (a) Who is responsible for the design and implementation of a computer system?
(b) List four major abilities that a person requires to take such a responsibility
 - (i)
 - (ii)
 - (iii)
 - (iv)
3. What is the role of a programmer in a data processing department?
4. Mention two typical high-level languages that application programmers use.
5. Outline the features of a fourth generation language (4GL).
6. List two major duties of an operator in a data processing department.
7. Give two situations when a maintenance engineer is called to give assistance.
8. What is the role of a webmaster?

Paper 1

1. From the following list of personnel commonly found in large data processing departments, choose any THREE and briefly describe their job responsibilities:

Systems analyst	Programmer	Computer operator	Data processing manager
Operations manager	Computer engineer	Webmaster	

- (a)

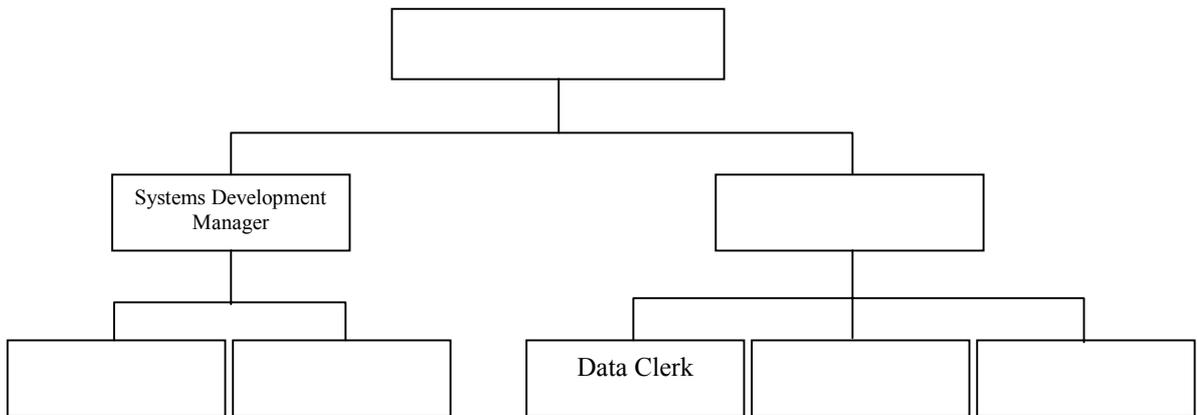
- (b)

- (c)

[3, 3, 3]

2. The chart below shows how the personnel of data processing department are organised. Insert the following job titles in the remaining blank boxes.

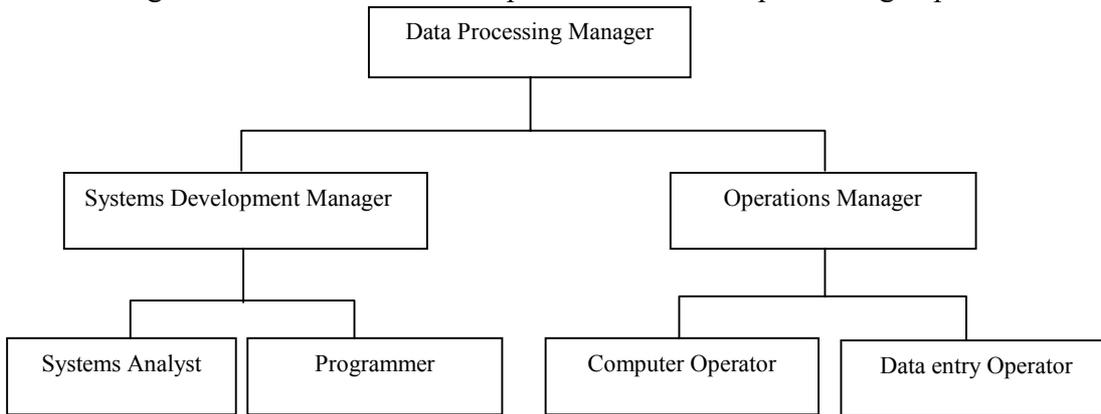
Systems Analyst	Operations Manager	Programmer
Data Processing manager	Computer operator	Technician



[6]

Paper 2B

1. The diagram below shows how the personnel of a data processing department are organised:



- (a) Briefly describe the duties performed by each of the seen personnel shown in the diagram. [2x7]
- (b) Mention ONE OTHER type of personnel who might be employed in a data processing department, and describe his/her duties. [1,2]

Paper 2A

1. In its data processing department a company employs system analysts, technical support staff and data entry operators.

- (a) Define the work of the three types of jobs mentioned above.
- (b) During the course of work, a data entry operator calls on the technical support staff complaining that the printing facility is not working. Mention three basic things that a technician will check.
- (c) The system analyst reviews periodically the computerised systems of the company
 - (i) Why is this routine job carried out?
 - (ii) What possible decisions might be taken by the systems analyst as a result of this task?