CHAPTER – 5
Internet

Syllabus:
History of Internet. Introduction to internet browsers, URL.
Introduction to email and how to check and compose an email?
Important websites related to pharmaceutical information – like sites for information regarding drugs, medical literature, plants, adverse effects, clinical data, patent sites, FDA, WHO etc.

HISTORY OF INTERNET

The late 1960s was during the cold war between two opponents United States and USSR. At that time computer was invented but they were not connected. In late 1960 U.S. Department of Defense funded a project named ARPANET (**A**dvanced **R**esearch **P**rojects **A**gency **Net**work). The goal of this project was to connect a number of computing facilities around United States and share information and resources. It was a long distance computer network. Arpanet was designed to keep working even if someone dropped a bomb on part of the network. If some of the communication links were destroyed, the rest of the network would still function.

This project was tremendously successful and throughout 1970s and 1980s, the network expanded and became the internet. At first, the internet connected only military and university computers, but, gradually, more and more companies and then individuals joined this network.

The advantage of internet is that if any computer or link in this network breaks down or stopped the information will be circulated through another route. So internet can never be completely switched off. Some where in the world network will be running.

Who runs internet? Actually nobody runs internet. This s because the Internet is not one big network. It is a collection of many, many smaller networks, computers and data communication lines. Each of this smaller pieces of network is owned and run by some person or organization, but no one manages the Internet as a whole.

Definition of Internet
The Internet is a global network of computer networks, linking computers so they can "communicate." This communication can take several forms, such as electronic mail (email), discussion groups, and information retrieved via the World Wide Web.
The Internet is also called the "*information superhighway*," the "*world's largest library*," and just "*the Net*."

INTERNET BROWSERS OR WEB BROWSER

What is a Web browser?
Web browsers, are software programs that help us navigate the Web and access text, graphics, hyperlinks, audio, video, and other multimedia.
Browsers work by "translating" or "interpreting" hypertext markup language (HTML). The code embedded in Web pages is in HTML. Browsers read this code and display the Web page accordingly.
In the software several buttons are provide for going to previous webpage (back), go to forward webpage (forward), print button allows to print the webpage that is displayed by the browser.
By clicking on a **hyperlink** in a webpage the browser will jump to another webpage or another location.

Examples of some browser softwares

- Microsoft Internet Explorer
- Netscape Navigator
- Fire Fox Web Browser
- Opera

Uniform Resource Locator (URL)
URL stands for Uniform Resource Locator. A URL is the complete "address" of a website on the Internet. URL is made up of four components, each separated by a slash (/). These components are illustrated in the chart below.

1. The scheme http://	**2.** The name of the host computer	**3.** The directory path on the host computer	**4.** The filename itself
http://	**www.yahoo.com/**	**literature/**	**index.htm**
http stands for **hypertext transfer protocol**. Followed by a colon and two slashes, this scheme identifies the type of resource.	Also known as the **domain name**, this is the part of the address that identifies a specific computer on the Internet. The example identifies Yahoo company (.com) site on the World Wide Web (www).	The third component identifies the directory. Some URLs contain several subdirectories, separated by slashes. The example identifies a directory on the Yahoo website called **literature**.	This is the **desired file**, or **Web page**. The example calls up the page index.htm file..

Various example of URLs:
http://www.harley.com/
http://www.harley.com/25-things/index.html
http://www.ibm.com/
mailto:billg@microsoft.com
news:rec.pets.cats.anecdotes
ftp://ftp.microsoft.com/products/msmq/demos.zip

What is Email?

Electronic mail, or "email," is a letter that allows us to send messages almost instantly to other Internet users around the world.
For reading and composing email a special softwares is required. It is known as Mail Client. Examples of such Mail Client softwares are : Microsoft Outlook, Pegasus Mail, Netscape Messenger etc.
The following functions can be done by an email client software:
1. Sending an electronic letter to some ones address.
2. Receiving an electronic mail from another computer.
3. Send mails to more than one address at a time.
4. Attach some files (images, documents, etc.) with the letter.
5. Download some attachments sent by some one else.

Checking an email:
1. First an email client is opened. For example let us open **Outlook Express** from the start menu of windows.
2. Click on the send / receive button. The email will be downloaded from an email server. The email will be placed in side the **inbox** folder.
3. Clicking on an email displays the details.

Composing an email
1. First an email client is opened. For example let us open **Outlook Express** from the start menu of windows.
2. Click on the **New Mail/Messege** button. A window will be opened.
3. The address of the recepient-1 (e.g. email address of friend-1) is typed in the text box beside **To:**.
4. Many more email addresses may be written into the carbon copy text box **Cc**:
5. Subject of the letter is written in the **Subject** text box.
6. Body of the email is written in the large, white box below.
7. Click the **Send** button. The message will be sent to the addresses written in the **To:** and **Cc:** box.

What is a Search Engine?
Search Engines are special sites where we can type in a keyword or phrase, and within seconds the engine will search its database to find relevant sites. We can also browse through a search engine's subject directory, narrowing categories with each click. Examples of some popular search engines are
http://www.altavista.com http://www.google.com http://www.yahoo.com

Important websites related to pharmaceutical information – like sites for information regarding drugs, medical literature, plants, adverse effects, clinical data, patent sites, FDA, WHO etc.

IMPORTANT WEBSITES RELATED TO PHARMACEUTICAL INFORMATION

In the internet there are millions of websites containing huge amount of information. However, reliable information related to pharmaceutical science can be obtained from the following popular websites and search engines:

1. Information regarding a drug

URL of the website	Nature of information provided in the site
Rx List http://www.rxlist.com/	Drug Information for Prescription Drugs, Side Effects, Interactions, Warnings, Overdose, Patient Info, Pharmacology, Online Pharmacy.
MedLine Plus http://www.nlm.nih.gov/medlineplus/druginformation.html	Guide to prescription and over-the-counter medications provided by the United States Pharmacopeia. Information regarding a pharmaceutical product available in USA and Canada are provided.
Drug Interaction Checker http://www.drugs.com/drug_interactions.html	Check whether the drugs in the prescription have any interaction.
U.S. Food and Drug Administration • Center for Drug Evaluation and Research http://www.fda.gov/cder/drug/default.htm	Drug information required for research scientist along with references. A very authentic site for drug information.
Alcohol and drug information by US Dept. of Health and Human Sciences http://www.health.org/	Information of prevention of drug addiction. Includes databases, research and general information.

2. Various drug authorities

URL of the website	Nature of information provided in the site
World Health Organization http://www.who.int/en/	Information regarding latest outbreak of a disease and its prevention measures.
US Food and Drug Administration (FDA) http://www.fda.gov/	Monitors medical devices, food, drug, biological products, cosmetics, radiactive products etc.
United States Patent and Trademarks Office http://www.uspto.gov/	Patents regarding any new drug, dosage form or devices can be searched.
Australian Patent Site http://www.ipaustralia.gov.au/	Patents regarding any new drug, dosage form or devices can be searched.
Pharmacy Council of India http://www.pciindia.org/	Information regarding regulations of Pharmacy Council of India. Information of approved degree and diploma institutes. Information regarding Pharmacy Profession.
All India Council for Technical Education http://www.aicte.ernet.in/	Information regarding all rules of AICTE, office bearers, approved diploma, degree and post graduate institutes and many more information regarding technical institute all over India.

3. Various search engines

When we do not know where to search the information we must go to some search engine sites, type in the "*search string*" and click the "Search" button. Names of several sites will appear with small amount of information regarding the site. From this list the desired information can be obtained.

URL of the website	Nature of information provided in the site
Google search http://www.google.com/	Any information can be searched.
Yahoo search http://www.yahoo.com/	Any information can be searched. Provides free email service.
Altavista Search http://www.altavista.com/	Any information can be searched.

CHAPTER – 1
Computer Fundamentals

Syllabus:
History: Introduction to Computer, Computer classifications (According to generation, size and use).
Hardware: Introduction to hardware, CPU, motherboard, input devices, output devices, storage devices, and memory. Various ports and slots available with motherboard – ISA, PCI, Serial, Parallel, PS/2 and USB and their uses.
Networking: Introduction to networking, classification of networking like LAN, WAN, MAN, Hardware of networking – Modem, Hub, Cables.
Power devices used in various line conditions like CVT, UPS.
Number systems – Binary, Octal, hexadecimal and their uses in computer.
Software: Introduction to software, simple examples and uses of Machine Language, Assembly Language and Higher level languages. Operating Systems and classifications of application softwares according to their uses.

INTRODUCTION TO COMPUTER

The modern computer has its roots in mechanical aids to counting, developed as early as 450 BC - the abacus; and from then on the development is going and has reached upto the present days super computers. On its way of development computer has transformed from simple to complicated, form mechanical to analog to digital computers.
Babbage (1833) designed a mechanical computer in 19th century, England and that used punch cards for data input. He name it 'Analytical Engine'. It possessed an internal memory, an arithmetic logic unit and it was capable of printing its result - in short Babbage's machine had all the main peripherals required for the present day computers.

Abacus

One of the earliest digital calculating device is the abacus, used in India, China and other Asian countries for centuries. Different models were available. One model is grooved tablet form and another tablet was bead-on-shaft model, which uses movable beads strung on rods to perform calculations.

Napier's Bone

In 1615, John Napier, a Scottish mathematician and the inventor of logarithms, developed a primitive form of slide rule known as the Napier's "bones". It was a mechanical arrangement of strips of bones with numbers painted on them, which when brought into combination, would perform direct multiplication.

Slide rule

In 1620, Edmund Gunter, a Britisher, invented a straight logarithmic scale, which performs calculations by using a compass. William Oughtred, also a Britisher, used two of Gunter's scales to make the first slide rule in 1621. It performs multiplication and division by adding and subtracting.

Pascal's Calculator

In 1642, at the age of 19, the French mathematician, Blaise Pascal, inveted one of the first mechanical adding machine, called the Pascal's adder, which could perform both addition and substraction. The machine was of the size of a shoe box and used a system of gears and wheels with teeth numbered from 0 to 9.
The electrically activated desk calculators has came into use about 1920.

Classification of computers acording to generation

The advancement of electronic computer technology is generally classified into four chronological categories called the first, second, third and the fourth generations.

First generation

The computers those used vacuum tubes came to be known as the *first generation* or *electronic vacuum tube* computers.
Developed during : 1940 - 58
Disadvantages: (i) Uses great amount of electric power.
(ii) Generate a great deal of heat
(iii) Were not very reliable
Features (i) Can store programs internally
(ii) Can receive their input data in the range of 3000 to 5000 additions per second.

Second generation

In 1948, transistors was invented at he Bell Laboratories. Computers utilizing transistors can to be known as *second generation* or *transistorized* computers.
e.g. IBM 7090 (1958)
Advantages: (i) Smaller in size
(ii) Less expensive
(iii) Require less power

(iv) Generate less heat
(v) More reliable than 1st generation computers.

Features (i) Faster input/ outputs
(ii) Better programming ability
(iii) Can transmit data through telephone line to distant computer.
(iv) Computation speed ranging from 2000 to 50,000 additions per seconds.

3rd generation

In 1957, the first integrated circuit was produced which is a device that incorporates all the capabilities of many transistors and other circuits using the integrated circuits came to be called the *third generation* or *integrated circuit* computers.
e.g. IBM 360 (1964)

4th generation

1970s saw the introduction of microprocessors, which are programmable, very large-scale integrated circuit chips that contain all the elements to process the data. Integration of very large scale of transistor chips has made it possible to contain a large amount of a computer's electronic parts on one or few complex microchips; making way for the *fourth generation* or *very large scale integrated circuit* computers.

Types of computers according to size

There are many different types of computer - used for many different jobs. Here are some of the popular types of computers those are found in use today.

1. Pocket (palmtop) Computer

Pocket computers have been designed to allow people to keep lots of information close to hand. They have small, light batteries that last a long time so that the whole computer is light and small enough to be carried around in someone's pocket.
These computers have special operating systems suited to pocket computers.

One problem with small computers is that they don't have full-sized keyboards attached. Both of the computers in these pictures use special pens and touch-sensitive screens to enter data as well as a number of small buttons or keys.

2 Laptop Computer

A laptop is the compact and lighter version of a desktop computer. All the softwares that can be run on a desktop can be run also on laptop. Modern laptops have floppy drives, CD-ROM drives and CD re-writers, and even DVD drives, modem etc.. They often have a full-sized, or near-fullsize (88 keys) keyboard and a mouse or a touch-sensitive mouse pad. The screen is usually a large Liquid Crystal Display (LCD).
Laptops are usually much more expensive than desktop computers. They have expensive battery packs that have to power the hard-disk, CD drives and LCD screen. The batteries generally don't last as long as those in a pocket computer and may need recharging more than once a day.
The main advantage of a laptop is that the person using it can transfer all the programs and data from their desktop computer to a portable computer

3. Micro (desktop) Computer

At the moment there are two main types of desktop computer available: the Mac (made by Apple Computers) and the PC. Mac is short for Macintosh - it is usually distinguished by its stylish looks and bright colours. The latest operating system for the Apple Mac (in 2002) is OS X.
When people talk about PCs they usually mean an 'IBM-compatible' computer based on an Intel (or similar) microprocessor. The most common operating system for the PC is Microsoft Windows (latest version Windows XP) although other operating systems are available (eg Linux).
These are very popular computers. They are designed to be used on a desk or table with a separate keyboard and mouse for input.

4 *Mainframe/Supercomputer*

These computers are used for performing many millions of complex calculations in a short time. They are very large and expensive. Some examples of supercomputers are Cray (of USA), PARAM of India.
They are used to predict the weather, handle bank accounts, hold insurance details.
In between the mainframe and the microcomputer is the minicomputer.

HARDWARE

The physical components i.e. mechanical, electrical and electronic parts that can be seen, touched, repaired, replaced and that are essential to make a computer system work are known as hardware.

The components of a computer hardware: Normally a computer has the following components.

1. System unit contains
   Cabinet, CPU, Motherboard, RAM, Hard Disk Drive (HDD), Floppy Disc Drive (FDD) and CD-ROM Drive.
2. Monitor
3. Keyboard
4. Mouse
5. Printer

CABINET

It is a metal container in which all the components of a computer is accommodated except keyboard, mouse and printer. In the front panel several switches are present in which mainly the following switches and LEDs (Light emission display) are present:

1. *Power switch*: To power on and off the computer.
2. *Reset Button*: To restart the computer without switching off the power (that is called "warm booting").
3. *Turbo button*: Some computers offer a choice of speeds at which they can run. Pressing the turbo button "ON" will run the computer in its fastest speed.
4. *Light Emission Displays (LEDs)*: LEDs are small light indicators that informs the operator about the present state of the computer. LEDs indicates about "Power On/Off status" of the computer, "HDD-activity"– whether there is any activity inside the harddisk etc.
5. *System speaker*: Produces the "beep" sound that can produce various information during execution of a program.

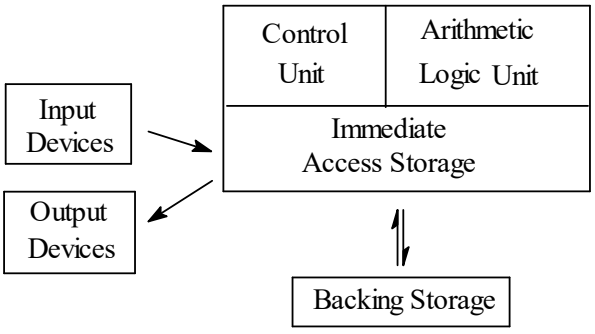
On the rear panel some ports are there. Generally three types of ports are there:

1. *Video / Monitor ports* (containing 9 pins) – These ports are used to connect to the display device i.e. monitor.
2. *Parallel Ports* (Containing 27 pins) – These ports are used to connect to printers (e.g LPT1 and LPT2)
3. *Serial Ports* (Containing 9 pins) – These are used to connect to communication devices like modems and mouse.
4. *Universal Serial Bus Port* (USB) – These ports are now-a-days used to connect to nearly all types of input devices.

On the rear panel some slots are provided for inserting expansion cards for various types of input devices like scanner.

THE CENTRAL PROCESSING UNIT

This part of the computer system collects the raw data from the input devices and converts it to useful information, which can then be used by the output devices. On some computers, the CPU can be a single microchip. On bigger systems, the CPU can be formed from a number of chips working together.



The CPU is made up of three main parts:
Control Unit this part controls the input and output devices
Arithmetic Logic Unit It does all the mathematics and makes the decisions
Immediate Access Store
This is the memory available for programmes and data. The more memory the CPU has – the more programs it can run at the same time; and the more data it can deal with in one go.
Data that is held in the Immediate Access Store (IAS) can be used immediately by the Arithmetic Logic Unit (ALU), if the data is held in the backing storage then it may take a little longer for the CPU to find it and copy it into memory. In 2004 a powerful microcomputer has minimum 128 Mb (megabytes) of memory (RAM).

The Control Unit is connected to a clock (a piece of crystal that 'ticks' very fast) and it issues a command at each 'tick' of the clock. If the computer has a very fast clock it means that the control unit is able to issue a lot of commands in a short time. So the computer carries out its jobs in a shorter time: it is faster.
In 2004 a fast microcomputer has a clock speed of 2.4 GHz (gigahertz) - this means that the control unit can issue about 2.4 billion instructions in a second.
Name of various brands of CPUs available in the market are:
Intel Pentium-I, Pentium-II, Pentium-III, Pentium-IV, Intel Celeron, Intel Centrino
AMD Athlon, Cyrix CPU etc.

MOTHERBOARD

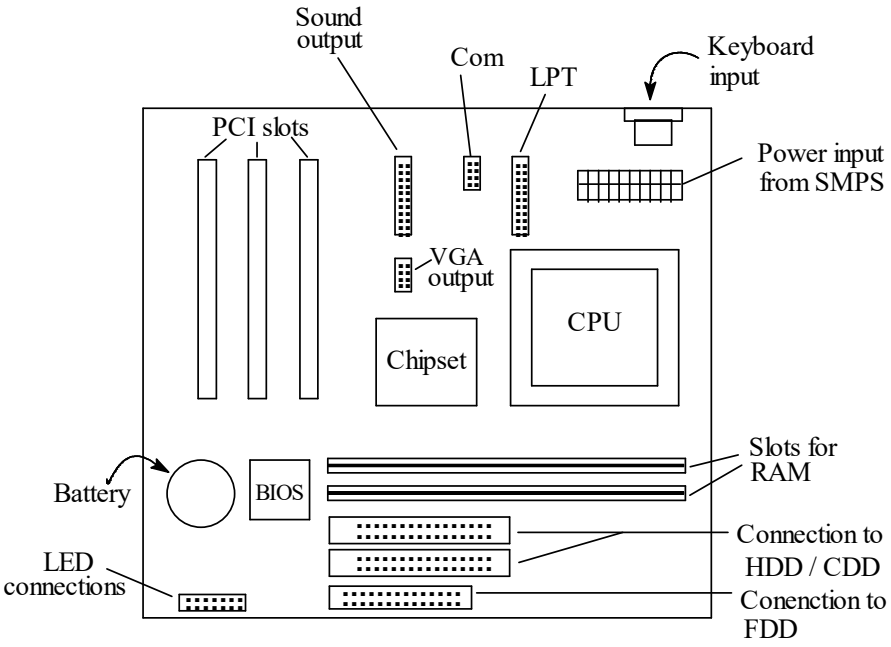
It is a circuit board that keeps the links between all the components of the system units like CPU, RAM, ROM, ports, and storage devices like hard disk drive, floppy disk drive, compact disk drive; and other computer peripherals like multimedia card, LAN card, modems, printers etc. It contains

SMPS:
(Switching Mode Power Supply)
Power supply comes as DC (direct current) from SMPS of the cabinet (in +12V, +5V and two Grounds) to the motherboard.

Keyboard Port:
Keyboard is attached to the motherboard through this port.

LPT: (Line Printer)
Printer is attached to the motherboard through this port.

COM: (Communication port)
Also called SERIAL PORT
Any communication device like modem can be attached to this port. Serial Mouse can be attached to this port also.



Schematic diagram of a Mother Board

PCI:
Through PCI slots various cards can be attached like PCI-Sound card, Scanner Card, Internal Modem card, LAN card etc.

PS/2: (Personal System / 2)
In present day PCs one separate mouse port and one keyboard port is provided through this PS/2 ports. Instead of attaching a serial mouse to COM port the COM port can be freed for other device like external modem.

USB Port: (Universal Serial Bus Port)
Through this port one or more Printers, Scanners, Webcameras and many other devices can be attached simultaneously.

CPU Socket:
Two types of CPU sockets are available one is square in shape; it is called "*Socket 7*" (actually called Socket 370) and another is long one called "*Socket 1*". Since the CPU gets heated when working so a heat sink (usually made of aluminium metal) is fitted on the CPU. A CPU fan again cools this heat sink.

VGA: Video Graphics Array
The video output is attached through this port i.e. the monitor is attached to this port.

Battery: A battery is provided on the mother board to keep a quartz clock ticking.
BIOS: (Basic Input / Output System)
This is a ROM (Read Only Memory) chip in which BIOS program is written that tells the computer how to 'talk' to the keyboard, monitor, etc. Without the BIOS even the Operating System can't load.
RAM Slots:
RAM (Random Access Memory) modules are fitted to the mother board thorough these slots. Generally two slots are provided in which either 8MB, 16MB, 32MB, 64MB, 128MB, 256MB or 256MB RAMs can be fitted.
Slots for connecting to Hard Disk Drives (Also called IDE Slots)
Two slots are provided on the motherboard for connecting either hard disk drives or compact disk drives by a IDE cable.
Slot for connecting to Floppy Disk Drive
One slot is provided for floppy disk drive. This one is shorter in size compared to IDE slots.

STORAGE DEVICES
Storage Devices are required to store data or program for immediate and later use.

Classification of storage devices
A. Primary Storage
    Read Only Memory (ROM)
    Random Access Memory (RAM)
B. Secondary Storage Devices
    Hard Disk, Floppy Disk, Compact Disk, Magnetic tape etc.

PRIMARY STORAGE DEVICES
Read only memory (ROM)
Read **O**nly **M**emory (or ROM) contains data that has been written once to the memory store but it can't be erased. Even if the computer is *switched off the data in the memory store remains safe*. The original data cannot be overwritten by new data.
The ROM chips contain programs that are built into the chips and direct the operations of the computer. ROM is not accessible to the user. In the ROM-BIOS the Basic Input/Output Systems instructions are written.
This BIOS (**B**asic **I**nput **O**utput **S**ystems) tells the computer how to 'talk' to the keyboard, monitor, etc. Without the BIOS even the Operating System cannot be loaded.

Random access memory (RAM)
Once the computer is running it needs a lot of extra memory which it can use as "thinking space" ... such as storing the results of calculations when a program is running. Data is deleted from these memories when the power switch is put off. This type of memory is called **R**andom **A**ccess **M**emory (or RAM). This types of memories are used for rapid access of data.

SECONDARY STORAGE DEVICES
Secondary storage devices provide a means of permanently storing the information contained in the temporary storage devices (RAM).

Magnetic Tape
It is made of a thin film of plastic with a magnetic covering on it. Data is stored in a sequence of magnetized and non-magnetized area on the tape (As video is stored on videotapes).
Disadvantage: To search any portion of the tape it is required to wind the whole tape to reach the site, which is time consuming.

Magnetic Disk (Floppy Disk)
It is a circular, plastic, flexible, and magnetic disk on which data is stored as magnetized and demagnetized areas. The magnetic disk is housed in a hard plastic cover.
Floppy disks are available in several sizes.
5.25 inches Floppy Disk: The magnetic disk is placed within a card sleeve and it allowed the disk to bend slightly – this is why they were called 'floppy' disks. Maximum memory 1.25 MB
3.5 inches Floppy Disk : The magnetic disk is placed within a hard plastic jacket that protects the disk. At a corner the write protection notch is there. If the shutter of the notch is in open condition, then no data can be copied into the disk – a condition called “write protected”. Maximum memory 1.44MB

Hard Disk (originally called Winchester Disks)
Floppy disks were useful but they couldn't store a lot of information.
Hard disk contains one or more metal disks coated with a magnetic material. The whole disk is enclosed in an airtight, dust-proof container. The data is read or written by read/write head placed on the access arm.
Advantages: The advantage of magnetic media (tapes, floppy disks, and hard disks) is that data can be saved easily. It can be deleted, and new data stored on top of it. Data access is very fast compared to other devices.
Optical Disks
In this disks the data is stored by burning tiny dots on shiny disks. The dots are burned onto the disk using a laser – and are read from the disk also with a laser. Because these disks use laser **light** to read and write to the disk they are often called **optical** disks.
Most optical disks can only be written to once and that data cannot be deleted or rewritten. This type of disk is called a WORM disk: **W**rite **O**nce **R**ead **M**any times. But mainly they are called CD-ROMs as they are a form of Read Only Memory.
Digital **V**ersatile **D**isks (or DVDs) are a type of optical disk that is able to hold a lot of data (enough for an entire film).
Some optical disks can be written to many times: they are more expensive. A common type of re-writable disk is the Minidisk: used for recording music and computer data.

Units of memory

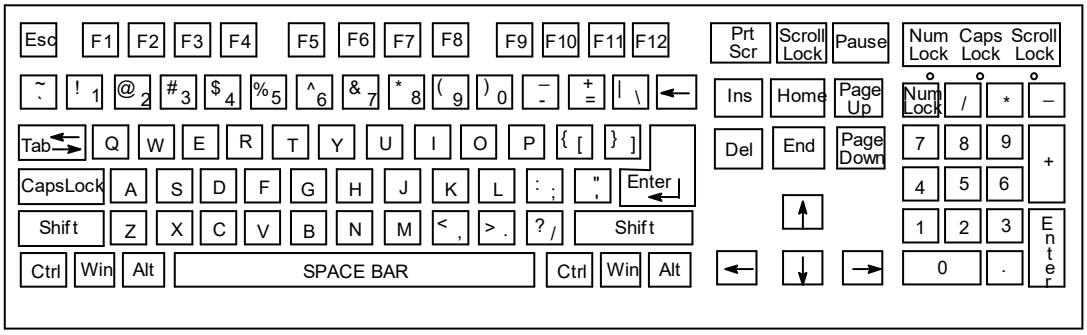
Data is stored in binary digit.
BIT: Each **BI**nary digi**T** (or BIT) of data.
BYTE: Eight bits in a group are called a BYTE
1 KILOBYTE (KB) = 1024 BYTES
1 MEGABYTE (MB) = 10^3 KB
1 GIGABYTE (GB) = 10^3 MB = 10^6 KB

INPUT DEVICES

Keyboard

Keyboard is the most common type of input device. It allows the user to communicate with the computer.
It contains four main areas:

1. The type writer keys
2. The function keys
3. The numeric keypad
4. The special –purpose keys



Enhanced Keyboard

Function keys:
The function keys are labeled as F1, F2, F3 …, F12 etc. They are used to send instructions to the software being used. Their use varies with the software program used. Frequently they are shortcuts for long commands. E.g. F1 key brings the help window in QBASIC or WINDOWS program.
The typewriter keys:
The central area of the keyboard contains the standard typing keys and the space bar.
A, B, C .., Z, 0, 1, 2, .., 9. As these keys are pressed characters (text and numbers) appear on the screen.
The cursor, also called *insertion point*, is a flashing bar that identifies the location of typing on the screen.

Other typing keys and their uses are described below:

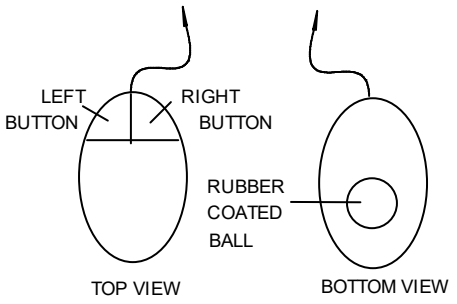
KEY	ACTION	KEY	ACTION
Shift + letter	Types uppercase letters	Shift + number	Types symbol shown above the number on that key.
Caps lock	Allows entry of all uppercase alphabetic characters without using "Shift"	Tab	Moves insertion point preset number of spaces to the right.
Enter	Moves the insertion point to the next line (same as type writer) or Allows the user to enter data or commands.	Shift + Tab	Moves insertion point preset number of spaces to the left.

The numeric keypad:
At the right side of the keyboard is the numeric keypad. It consists of nine keys with arrows and numbers on them. These keys can be used to enter numbers or to direct the movement of the insertion point on the screen.
To use the numbers in the numeric keypad the "Num Lock" key must be on. A LED will glow to display the status "On". Pressing the "Num Lock" key again will put off the LED.
When the "Num Lock" is off press the arrow keys to move the cursor in all directions.

Special purpose keys:
These keys have special meanings depending on the software used.
Generally they have the following uses:

KEY	ACTION	KEY	ACTION
"Esc" – Escape	Quits or goes back one step in a program; erases existing command	"Ctrl" – Control	Used in combination with another key to perform a special task
"Alt" – Alternate	Assigns another function to a given key	"Print Screen"	Prints a copy on what ever is on the screen
"Scroll Lock"	When on using the ↑ or ↓ keys moves document up or down on the screen, but insertion point does not move.	"Pause"	Let the program to stop for a short time
"Break"	Stops a command from completing execution.	"Insert" and "Ins"	Allows the user to insert characters between other characters.
"Delete" and "Del"	Erases character next to the cursor.	"Backspace"	Moves cursor to the left and erases a character. (some times it appears as ←)
"End"	Brings the cursor to the end of a line	"Home"	Carry the cursor to the beginning of a line.
"Page Up"	Moves up the document page	"Page Down"	Moves down the document page

MOUSE
The mouse is a hand held input device that controls the pointer (arrow is called mouse pointer) on the screen. When the mouse is moved on a flat surface a rubber coated ball on the bottom of the mouse rotates. The movement of the ball is translated into signals that tell the computer how to move the mouse pointer.
On the top of the mouse there are two buttons, called the Left button and Right button. Clicking on either of the buttons produces some actions or programs to run.
Some times a small wheel is attached to the mouse. It is called "scrollable mouse". It helps to scroll a document up and down without the help of scrolling keys on the keyboard.



OUTPUT DEVICES

Monitor

The computer screen is called a Monitor.
It displays the instruction that is input by a computer operator and after calculations displays result as text, graphs etc.
The computer screen can be either a monochrome or color screen.
Monochrome Monitor displays in one color only, usually black / white, and with different shades of gray.
Each point on the a color screen is called a *pixel*. The color of each pixel is determined by a notation of RGB, i.e. Red Green Blue. Each color may have 256 different shades. So a red color on the screen is expressed by RGB (255 , 0, 0).

Two types of monitors are found the cheaper quality is called Cathode Ray Tube (CRT) Monitor and the other is Liquid Crystal Display (LCD) Monitor. In the desktop computers CRT-Monitors are used and LCD Monitors are used in laptop computers.

Printers

The printer is an output device that produces a permanent copy of the work printed on paper. It is attached to the computer through LPT port. Printers can print both texts and graphics but the quality of graphics generally not good. To get a professional quality graphics output "Plotter" is used.
Plotter has different color pens and can draw presentation quality graphics.

Printers may be three types : Dot Matrix, Ink Jet or Laser Jet.

Dot Matrix Printer (DMP): Dot matrix printers are often used to print draft quality output. They produce characters by a series of pins that hits on the printer ribbon to produce dots on the paper.
The printers are fast and economical but produce noise.

Ink Jet Printers : Ink Jet printers spray inks from tiny nozzles on the ink-cartridges (ink container) in the pattern of characters. Color photographs can also be produced with color cartridges.

Laser Printers: The laser printer is a high resolution printer that produces high quality output. (N.B. the photocopier machine is a type of laser printer.) The characters are first created by a laser on an electrically charged drum. The paper is then get charges electrostatically. Inks in powder form sticks to the charged surface. A hot drum melts the ink particles to stick on the paper surface.

Various ports and slots available with motherboard

PCI, Serial, Parallel, PS/2 and USB and their uses.

PCI:
Through PCI slots various cards can be attached like PCI-Sound card, Scanner Card, Internal Modem card, LAN card etc.

PS/2 Port: (Personal System / 2)
Two PS/2 ports are present in the motherboard, one for mouse and the other for keyboard.

USB Port: (Universal Serial Bus Port)
Through this port one or more Printers, Scanners, Web-cameras and many other devices can be attached simultaneously.

Serial Port: This port is also called COM port. The serial port contains holes for 9 pins. Through this port either a mouse or a communication device (like modem) can be attached. Generally two serial ports are provided in the motherboard.

Parallel Port: The parallel port contains holes for 27 pins. Printer is attached to this port.

NETWORKING
What is a Network?

A network consists of two or more computers that are linked in order to share resources (such as printers and CD-ROMs), exchange files, or allow electronic communications. The computers on a network may be linked through cables, telephone lines, radio waves, satellites, or infrared light beams.
The three basic types of networks include:

- Local Area Network (LAN)
- Metropolitan Area Network (MAN)
- Wide Area Network (WAN)

Local Area Network (LAN)

A Local Area Network (LAN) is a network that is confined to a relatively small area. It is generally limited to a geographic area such as a laboratory, institute, or building. Rarely are LAN computers more than a mile apart.
In a typical LAN configuration, one computer is designated as the **file server**. It stores all of the software that controls the network, as well as the software that can be shared by the computers attached to the network. Computers connected to the file server are called **workstations**. The workstations can be less powerful than the file server, and they may have additional software on their hard drives.
On most LANs, cables are used to connect the **network interface cards** in each computer.

Metropolitan Area Network (MAN)

A Metropolitan Area Network (MAN) covers larger geographic areas, such as cities. By interconnecting smaller networks within a large geographic area, information is easily disseminated throughout the network. Local libraries and government agencies often use a MAN to connect to citizens and private industries.
It connects all **workstations** to a centralized **mainframe** at the district office by using dedicated phone lines, coaxial cabling, and wireless communications.

Wide Area Network (WAN)

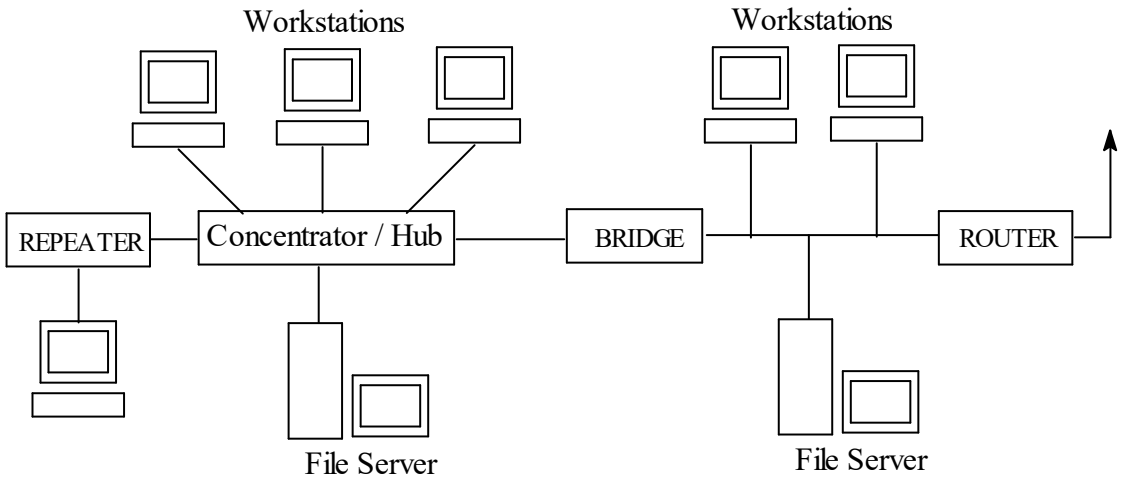
Wide Area Networks (WANs) connect larger geographic areas, such as a country or the world. Dedicated transoceanic cabling or satellite uplinks may be used to connect this type of network.

A WAN is complicated. It uses **multiplexers** to connect local and metropolitan networks to global communications networks like the Internet. To users, however, a WAN will not appear to be much different than a LAN or a MAN. In India the Railway reservation network is an example of WAN.

Networking Hardware

Networking hardware includes all computers, peripherals, interface cards and other equipment needed to perform data-processing and communications within the network. The equipment are as follows:

Servers
Workstations
Network Interface Cards
Concentrators/Hubs
Repeaters
Bridges
Routers



File Servers

A file server stands at the heart of most networks. It is a very fast computer with a large amount of RAM and storage space, along with a fast network interface card. The network operating system software (e.g. WINDOWS NT, WINDOWS SERVER 2003, UNIX) resides on this computer, along with any software applications (MSWord, MSExcel, etc.) and data files that need to be shared.

The file server controls the communication of information between the nodes on a network.

Workstations

All the computers connected to the file server on a network are called workstations. A typical workstation is a computer that is configured with a network interface card, networking software, and the appropriate cables. Workstations do not necessarily need floppy disk drives or hard drives because files can be saved on the file server. Almost any computer can serve as a network workstation.

Network Interface Cards

The network interface card (NIC) provides the physical connection between the network and the computer workstation. The NIC card is fitted on the mother board on PCI slot. Example of NIC card is Ethernet card.

Concentrators / Hubs

A concentrator is a device that provides a central connection point for cables from workstations, servers, and peripherals. In a star topology, twisted-pair wire (cable) is run from each workstation to a hub. Hubs have multiple slots (8ports, 16 ports etc.) into which the cable from all workstations are connected to the slot by RJ connector (similar to telephone cable connector).

Two types of hubs are available: (i) Passive and (ii) Active

(i) Passive hubs allow the signal to pass from one computer to another without any change.

(ii) Active hubs electrically amplify the signal as it moves from one computer to another.

Repeaters
Since a signal loses strength as it passes along a cable, it is often necessary to boost the signal with a device called a repeater. The repeater electrically amplifies the signal it receives and rebroadcasts it. Repeaters can be separate devices or they can be incorporated into a concentrator. They are used when the total length of your network cable exceeds the limit (say 100 meters).

Bridges
A bridge is a device that allows to segment a large network into two smaller, more efficient networks. A bridge monitors the information traffic on both sides of the network so that it can pass packets of information to the correct location.

Routers
A router translates information from one network to another; it is similar to a super-intelligent bridge. Routers select the best path to route a message, based on the destination address and origin address. For example a router translates the information on a LAN to Internet.

Network Cable
Cable is the medium through which information usually moves from one network device to another. There are several types of cable which are commonly used with LANs. The type of cable chosen for a network is related to the network's topology, protocol, and size. Examples of various types of cables are as follows:

- Unshielded Twisted Pair (UTP) Cable
- Shielded Twisted Pair (STP) Cable
- Coaxial Cable
- Fiber Optic Cable
- Wireless LANs

POWER DEVICES

The power to a computer can be supplied from AC supply or from DC batteries.
Low quality of AC supply may crash hard disk or may damage the mother board or other devices in a computer. So the quality of AC supply should be corrected.
Two instruments can be used to correct the line condition
(i) Constant Voltage Stabilizer (CVT) and (ii) Uninterrupted Power Supply (UPS).

Constant Voltage Stabilizer (CVT)
AC current input inside the CVT and the output voltage remains fixed within a narrow range of 230 volts. CVT can correct any low or high voltage input and supply a constant voltage to the computer but it cannot correct spikes or surge condition. If power cut occurs immediately the output also stops.

Uninterrupted Power Supply (UPS)
AC current input inside the UPS and the output voltage remains fixed at 230 volts. UPS can correct any low or high voltage input and supply a constant voltage to the computer and it can correct spikes or surge conditions. If power cut occurs then it can supply power to the computers for few minutes from the battery inside. This is called “power backup”. After a power cut the UPS gives enough time to close all the softwares and shut down the computer properly.

SOFTWARE

The group of computer programs that a computer needs to function is collectively known as software.

Operating system:

It is a complex group of programs that

(i) control the flow of programs and data through the computer
(ii) control input and output devices
(iii) manage the storage facilities of the computer by storing data and programs and retrieving them when required.

Examples : DOS, UNIX, WINDOWS, OS/2, LINUX, FREEBSD

Programming Languages

Examples of Programming languages:

FORTRAN : **F**ormula **T**ranslation
BASIC : **B**eginners **A**ll-purpose **S**ymbolic **I**nstruction **C**ode
COBOL : **C**ombined **B**usiness **O**riented **L**anguage
C, ADA, PASCAL etc.

Object oriented programming language

C++, Visual Basic

Machine language

Every computer has its own language:

- Programs written in machine language are machine dependent and are good for those particular type of machines only.
- Machine language instructions are usually represented by binary, octal or hexadecimal number system.
- *Disadvantage* :
  1. It is very difficult to make any change to a program written in machine language.
  2. It varies from machine to machine.

Assembly language

- In assembly language binary or octal or hexadecimal codes are replaced by symbols.
- Data-items are referred to by descriptive names (such as Gross or Tax).
- Operations are specified in symbolic codes called 'mnemonics", instead of numeric operation codes.
- Machine cannot understand an assembly language program directly. First the assembly language program is translated into machine language by a program named 'assembler', and then machine can understand it.

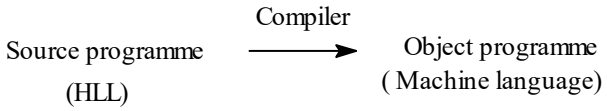
Disadvantages :

1. Assembly language is far from human languages (the English or other languages) . So it is difficult for a programmer to program in assembly language.
2. It varies from machine to machine.

Higher level language:

These languages allow the use of symbols and terminology that is familiar to human language.

- These are machine independent languages. Some programs executed in different machines with little or no alterations.
- The original program written in HLL is called 'source program". Source program is translated into 'object program'. This object program is carried out by a program known as 'compiler'. The method is known as 'compilation'. Object program is a machine language program.



Machine language version	Assembly language version				Higher-level language-version
013737 000016 000022	GO:	MOV	B	A	
063737 000020 000022		ADD	C	A	
000000		HALT			
000100	B	WORD	100		A =B+C
000150	C	WORD	150		
000000	A	WORD	0		
		.END	GO		

Application software

Application software is a set of program designed for specific uses or "*application*", such as word processing, graphics, or spreadsheet analysis.

To run an application-software, the programs are loaded into computer memory; execute (run) the programs, and then create, edit or update a file.

Application softwares are written in Higher Level Languages.

The operating system acts as a communication link between the application softwares and the hardware. It is responsible for loading the application software into the memory and then starting thew program. It also retrieves data files and save them to disk when directed. After using, the programmer "quit" the program.

Following are some categories of applications progarms:

Word processors:	Wordstar, MS Word
Desktop Publishing:	Adobe PageMaket, Aldus PageMaker
Database:	MS Access, FoxPro, Oracle, SQL, Sybase
Spreadsheet:	MS Excel, Lotus 1-2-3
Accounting:	Tally, Accord
Drawing:	Autocad, Corel Draw, ChemDraw
Photo manipulator:	Photofinish, Adobe Photoshop, Paintshop Pro
Animation:	Macromedia Flash
Multimedia:	Macromedia Director

Computer Applications

Chapter-1: Introduction to computer

Q1. Classify computers according to generations. [4]
Q2. Classify computers according to size. [4]
Q3. Write note on the following hardware components:
(a) CPU, (b) Storage Devices, (c) Input devices, (d) Output devices
Q4. Give a schematic functional diagram of CPU. [2]
Q5. What is ALU? [2]

CHAPTER 2.6

BASIC

Syllabus

Introduction to programming: Problem analysis, algorithm, coding, compilation, execution, debugging and testing, program documentation.
Constants, types of variables, array variables, arithmetic operations, hierarchy of operations, parentheses rules, logical operations, library functions.
Design of programs: Initialization, input, validation, processing, print, closing procedure.
Sequential structure of programming
Selective structure: IF statement, block IF, nested IF
Repetitive structure: FOR…NEXT loop, WHILE…WEND statement.
Function and Subroutine
Graphics: LINE, CIRCLE, and changing the color of background, drawing a graph with a set of data.
Formatting

PRINCIPLES OF PROGRAMMING

Principle behind programming with any programming language is nothing but computer-aided problem solving. It consists of the following processes:

1. Problem analysis.
2. Algorithm development.
3. Program coding.
4. Program compilation and execution
5. Program debugging and testing.
6. Program documentation.

Problem analysis

Whenever a problem is obtained it is first analyzed systematically to feed into a computer program. The following points are emphasized:

(i) Some information/data should be input, so input data is identified. Input information may be some data.
(ii) After program execution some information / data should come out as output, so output data is also identified. Output information may be data or graph or both.
(iii) The input data should be manipulated by some procedure, say by some calculation etc. to obtain the output information.

Example

Given three numbers x1, x2, and x3; find the largest of these numbers.

Problem analysis

Here the only information required as output in the largest of the three numbers x1, x2, and x3. The input information are the numbers x1, x2 and x3. In computer programming those three numbers are given three variable names e.g. X1, X2 and X3 to represent the memory cells containing these data items.

Initially, X1 is compared with X2. If X1 is larger than X2, it is compared with X3. If X1 is larger than X3, X1 is the largest number; otherwise X3 is the largest number. However, if X1 is not larger than X2, X2 is compared with X3. If X2 is larger than X3, X2 is the largest number; otherwise X3 is the largest number.

Algorithm development

Once the problem input and output data are identified and precise statement of the problem in terms of the input and output data is available, the problem is expressed by abbreviated statements in English (or other natural languages) that specify the steps to be performed in solving the problem are called **program design language (PDL) statements** or **pseudocode**.

The finite set of PDL statements, which describe the logic for solving a specific problem, is called an **algorithm** and the process of defining PDL statements is called the **algorithm development**.
An algorithm has the following features:

1. it has a finite number of inputs
2. it terminates after a finite number of steps
3. the action taken in each step are precise and unambiguous
4. all operations specific in each step can be done exactly and in a finite amount of time and
5. it has one or more outputs derived from the inputs by applying the algorithm.

Algorithm of the previous problem

1. Input X1, X2, X3.
2. Compare X1 with X2.
3. If X1 is larger than X2, compare X1 with X3.

If X1 is larger than X3, report X1 as the largest; otherwise report X3 as the largest.

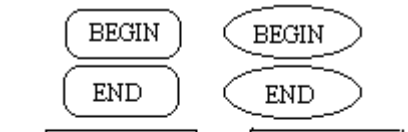
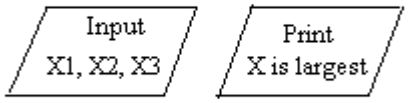
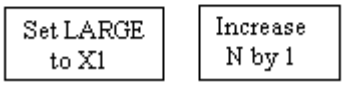
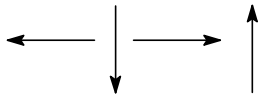
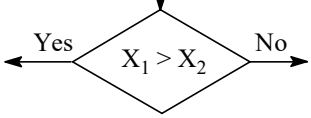
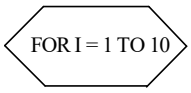
4. If X2 is not larger than X2, compare X2 with X3.

If X2 is larger than X3, report X2 as the largest; otherwise report X3 as the largest.

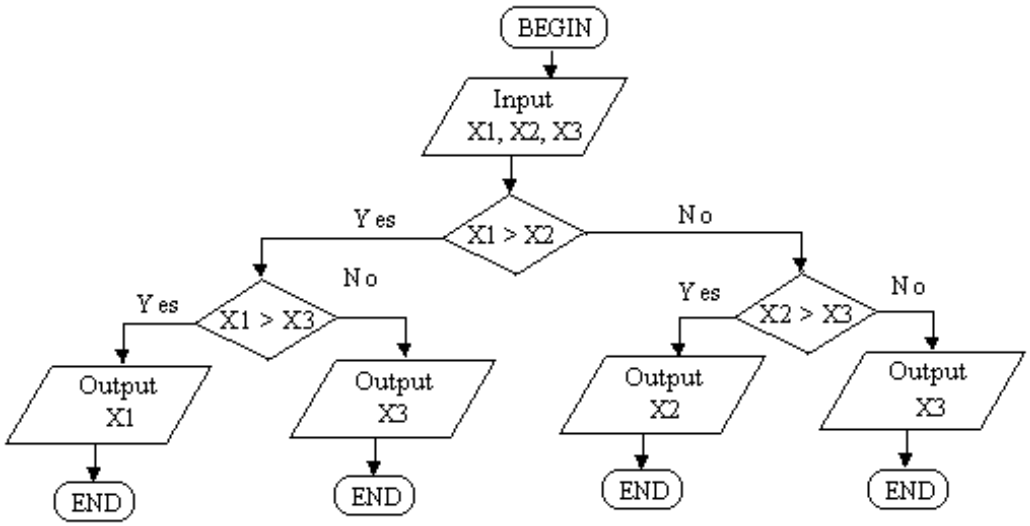
5. End.

Flowcharts: A diagram of the statements of an algorithm showing the relationships among these s**tatements is called** flowchart**.**

Symbols used in flowchart

Name of the symbol	**Symbol**	**Use**
1. Terminal Symbol		Used to indicate the beginning, or end, of a n algorithm.
2. Input / Output Symbol		A parallelogram is used to indicate the input and output, of information to or from the computer.
3. Computation or Process Symbol		A rectangle is used to indicate the assignment of values or computation.
4. Flowline Symbol		An arrow is used to indicate the order in which the steps of the algorithms are to be carried out.
5. Decision Symbol		A diamond is used to indicate a decision point. The question inside the diamond can have answers 'yes' or 'no'
6. Loop Structure Symbol		A hexagon indicates the beginning of a loop structure.

Flowchart for the above problem



Program coding
Program coding is the process of translating the solution of the problem, as worked out in algorithm, or flowchart, into the exact instructions that will make up the program.
The algorithm must be written using the vocabulary of the a programming language (ForTran, BASIC, C etc.) and must conform to the syntax of that programming language.

```
REM
REM     QBASIC Code for Problem 1
REM
DIM  X1 AS SINGLE, X2 AS SINGLE, X3 AS SINGLE
INPUT X1, X2, X3
IF (X1 > X2) THEN
     IF (X1 > X3) THEN
          PRINT X1
     ELSE
          PRINT X3
          ENDIF
ELSE
          IF (X2 > X3) THEN
                    PRINT X2
          ELSE
                    PRINT X3
          ENDIF
ENDIF
STOP
END
```

Program processing
Statements are entered in the computer through a key board the program is compiled by the ForTran compiler and then the program is executed.
Program debugging and testing
Very often, no matter how simple, or complex, the programs do not execute successful for the first time; the programs contain at least a few errors or bugs. The programs are therefore, **debugged**, i.e. errors or bugs are eliminated, till they execute successfully.
Program documentation
Before putting the program into use, the documentation for the program is provided and this is the last step in the programming process. Program documentation provides a written description of each step of the program.

PROGRAMMING IN QBASIC

BASIC stands for **B**eginner's **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode
QBASIC is also a version of BASIC language, called Quick BASIC. It was developed by Bill Gates and Paul Allen way back in 1975.
Advantages of QBASIC
1. It is a very easy to use language because many of its commands are pure English like LET, PRINT, etc. .
2. It has a simple structure for its programs: its lines may be numbered as 10, 20, 30, etc. and are executed in order.
3. It has structural similarity with many other higher level languages.
4. Simple graphics can be drawn with this program.
5. Easy helps are given readily.

How to start QBASIC program
First of all search the program QBASIC.EXE in the hard disk or floppy disk. Copy it in the floppy. Go to
Start → Click Program → Click MSDOS.
A DOS screen will be opened within the Windows. Go to A:\>QBASIC ↵
QBASIC screen will open. Go on typing the codes in the QBASIC window. Press F5 to run the program.

CONSTANTS

A **constant** is a quantity whose value does not change during program execution. It may be of **numeric, character** or **logical** type.

1. Numeric constants

Any string of digits, preceded by a single algebraic sign, is called a numeric constant. The general rules for forming numeric type constants are given below:

(a) The decimal digits 0,1,2,3,….,9 are used.
(b) The *minus* sign must precede a negative constant; a plus sigh is optional and an unsigned constant is considered positive.
(c) No commas are to be included in a string of digits.
(d) The number of digits of a constant is limited by either a maximum number of digits or a maximum magnitude.
(e) A space within a numeric constant is not allowed.

Numeric constants are subdivided into (i) *Integer* and (ii) *Real* constants.

(i) **Integer constant** (also called *fixed point co*nstant) is a constant, which does not include a decimal point.

Thus

25 0 -7 +15274

are valid integer constants, whereas the following are not for the reasons indicated beside:

Incorrect	Reasons	Correct
18.0	Contains a decimal point	18
-284.	Contains a decimal point	-284
10,235	Contains a comma.	10235
3-	Minus sign cannot be after the number.	– 3
– – 7	Two minus signs cannot precede a number	– 7

(ii) **Real constants** (also called *floating point constant*) is a constant with a **decimal** point and may have a fractional part.

Thus

18.3 – 163.0 42. + .0125 + 0.256

are valid real constants, whereas the following are not valid real constants,

Incorrect	Reasons	Correct
1,465.3	Contains a comma.	1465.3
– 56	Contains no decimal point	– 56.

Scientific notation

Very small and very large numbers are expressed conveniently by scientific notation.

Standard decimal form	Scientific notation	BASIC notation
0.0000567	5.67×10^{-5}	5.67E–5
0.0000000679	6.79×10^{-8}	6.79E–8
20000000000	2.0×10^{10}	2.0E+10
– 0.000000076	-7.6×10^{-8}	–7.6E-8

So the general form of a scientific notation is

$$(\text{co-efficient}) \times 10^{(\text{integer})} = (\text{co-efficient})\ E\ (\text{integer})$$

The part appearing before E is called **mantissa** and the part following the E is called **exponent**.

2. Character constant

Any sequence of acceptable characters in the BASIC character set, enclosed within double quote (“ “), is called a **character constant** or **string constant**.

The number of characters in a character constant is the **length** of the constant.

If an apostrophe sign is to be one of the characters of the constants, it must be entered as a single quote.

Examples:

Character / String constant	Length of the string	Remarks
“ UNIX–PC ”	7	‘–‘ is a character
“ I.P.T.”	6	‘ .’ dots are also characters
“Don’t”	5	‘ single quote is a character
“More Pages”	10	‘ ‘ space is also a character

VARIABLES

A **variable** is a name used to identify the data stored in a memory location whose content may change during program execution. The rules for naming a variable are given below:

1. A variable name can contain letters A to Z (i.e. alphabet characters) and digits 0 to 9 (i.e. numerical characters) but no special characters such as +, $, *, etc.
2. The first character of a variable **must** be a letter (A to Z).
3. Maximum length of a variable name may be 40 character long.
4. No space is allowed in the variable name.

The following are valid variable names

A2X3 ITEM PAY_DAY123

The following are not valid variable names

Variable Name	Reasons
2AB	The first character is not a letter
BETA–3	(–) is a special character
X18.2	(.) is a special character
ABCDEFGHIJKLMNOPQRSTUVWZYZ12	More than 40 characters

- Certain words called reserve words such as PRINT, INPUT, CLS etc. are part of the BASIC language and hence are not valid variable names.

TYPE SPECIFICATION STATEMENTS

A variable can store a certain type of data after specifying the type of that variable. Following are the type specification statement of variables:

Type specification	Basic Keywords	Remarks	Data type suffix
Integer	INTEGER	A 16-bit signed integer variable.	%
Long-integer	LONG	A 32-bit signed integer variable.	&
Single-precision	SINGLE	A single-precision 32-bit floating-point variable.	!
Double-precision	DOUBLE	A double-precision 64-bit floating-point variable.	#
String	STRING * n%	A fixed-length string variable n% bytes long.	
String	STRING	A variable-length string variable.	$

Type specification statements may be declared **explicitly** in the following syntax

DIM variablename AS variabletype

e.g.

```
DIM NUM AS INTEGER
DIM NUM1 AS INTEGER, NUM2 AS LONG, NUM3 AS SINGLE, NUM4 AS DOUBLE
DIM FNAME AS STRING*15   –FNAME is a variable that contains maximum length of 15 characters
DIM LNAME AS STRING      – LNAME contains any length of characters.
```

A variable can also be declared implicitly by within the program by *Variablename suffix* For example

NUM% NUM2& NUM4#
NUM1% NUM3! FNAME$
LNAME$

Limits of QBASIC environment

	Maximum limit	Minimum limit
Variable name length	40 characters	1 character
String length	32,767 characters	0 characters
Integers	32,767	– 32,767
Long Integers	2,147,483,647	-2,147,483,648
Single-precision numbers		
Positive number	3.402823E+38	2.802597E-45
Negative number	-2.802597E-45	-3.402823E+38
Double-precision numbers		
Positive number	1.79769313486231D+308	4.940656458412465D-324
Negative number	-4.940656458412465D-324	-1.79769313486231D+308

Undeclared variables

If a variable is not explicitly declared by *DIM variablename AS variabletype* then QBASIC automatically assigns the variable as **Single precision.**

ARRAY VARIABLES

An array is a collection of values, or data, that are related in some way. The data may be stored in consecutive memory locations, each of which can be accessed directly. Such a *collection of values*, or data, is called an ***array*** and the *data item* its ***element***.

e.g. The test scores of 100 students are S_1, S_2, S_3, ….., S_{100}. Where S is the single group name for all the student's score and the subscript (1,2,3, .. .100) identifies the test scores of a specific student. Thus S_1 is the scores of the first student, S_{10} is the score of the 10^{th} student and so on.

In BASIC language the above data set is represented as follows: S(1), S(2), S(3),…., S(100).

The number of subscripts in a pair of parentheses may be more than one.

e.g. S(23,34,2)

The name and the range of subscript of an array may be declared in a DIM statement of the form:

DIM variable (m_1 TO n_1, m_2 TO n_2, .. , m_k TO n_k) AS type

- Each pair of integer constants, usually called **parameters.**
  m_1 is the lower bound of the array, minimum value is 0.
  n_1 is the upper bound of the array.
- AS type Declares the data type of the array or variable (INTEGER, LONG, SINGLE, DOUBLE, STRING)

e.g. DIM PUPIL (0 TO 10) AS INTEGER, MARKS (10 TO 50) AS LONG

ARITHMETIC OPERATIONS

The numeric data can be manipulated using *arithmetic operations*. A combination of variables and constants together with operation symbols, we use the phrase *expression*.

For numeric data there are six arithmetic operations:

Operation	Arithmetic Operator	BASIC operator	Examples
Addition	+	+ (plus sign)	P + Q
Subtraction	–	– (minus sign)	P – Q
Multiplication	x	* (asterisk sign)	P * Q
Division	÷	/ (slash sign)	P / Q
Exponentiation	P^Q	^ (caret sign)	P ^ Q
Negation	– P	– P	– P

Rules

1. Single mode operation

When two constants or variables of the same type are combined, using one of the four arithmetic operations (+, –, *, /) the result will be same *type* as the constants or the variables.

e.g. 8 + 2 = 10 8 – 2 = 6 8 * 2 = 16 8 / 2 = 4

e.g. 8. + 2. = 10. 8.0 – 2. = 6.0 8. * 2. = 16. 8. / 2. = 4.

If both the constants are integers and after division a fraction appears then the result will be the integer portion of the fraction

e.g. 5 / 2 = 2 (not 2.5) 1 / 2 = 0 (not 0.5)

If both the constants are real (i.e. either *single* or *double*) and after division a fraction appears then the result will be the real.

e.g. 5./2. = 2.5 1. / 2. = 0.5

2. Mixed mode operation

When one integer and another real constants or variables are involved in an operation then the integer is automatically converted to its real equivalent, and the result is a real type.

e.g. 5 / 2.0 = 2.5

3. Exponentiation

(a) If the *exponent* is an integer value the QBASIC performs the operation by repeated multiplication.
e.g. – 4 ^ 2 = (–4) * (–4) = 16

(b) If the *exponent* is a real (either single or double) then QBASIC performs the operation by logarithmic method. E.g.
– 4.0 ^ 2.0 = exp (2.0 log (–4.0)) = Undefined
Because logarithm of negative value is *undefined*.

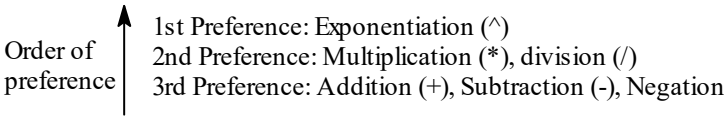
N.B.

$(–4.0)^{2.0} = x$ (let) Taking logarithm on both sides yields: 2.0 log (–4.0) = log x

Therefore, $x = 10^{2.0 \log (-4.0)}$. log (–4.0) is undefined.

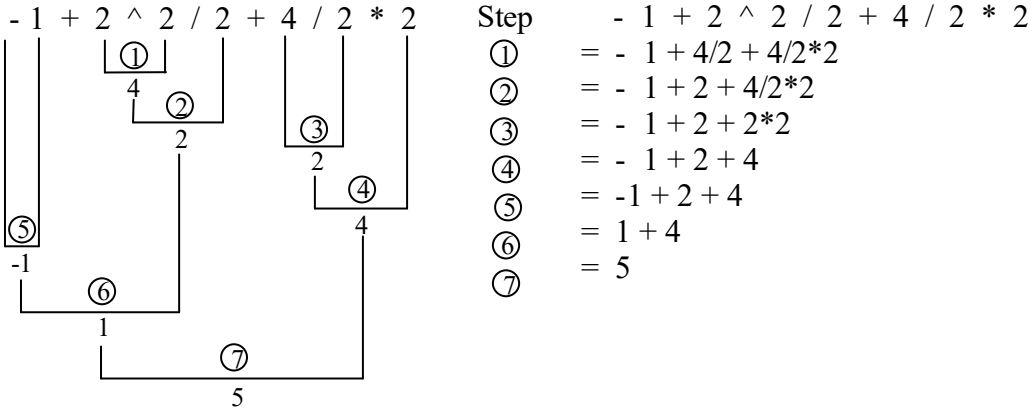
PRECEDENCE RULE

Precedence rule states that the operations will be performed in the following descending order of precedence:

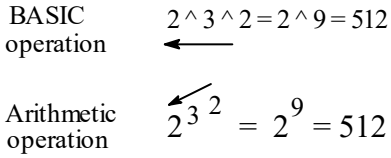


- When the order of precedence of operators is same, such as in multiplication and division, the operations will be performed in order from **left to right**.

e.g.

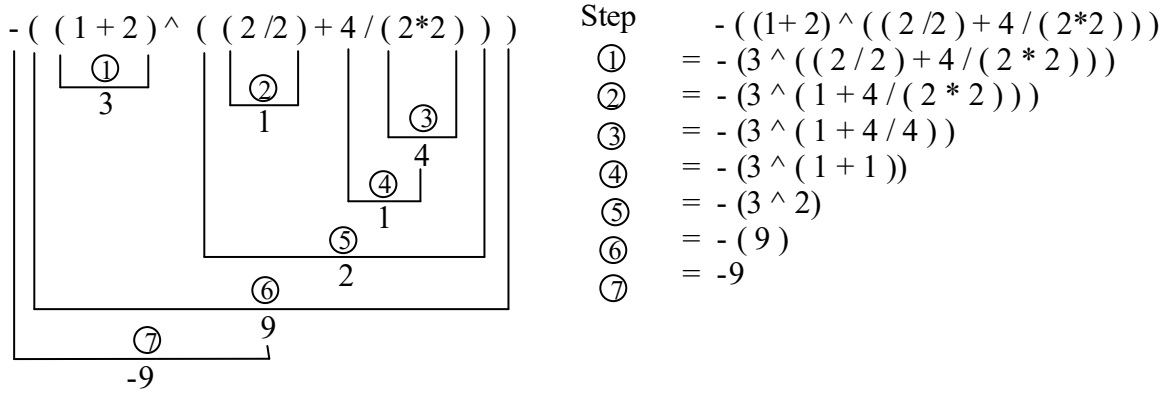


- If exponentiation operator appears multiple times in succession, the operation is performed from **right to left**.



PARENTHESES RULE

Parentheses rule states that operation will be performed in the innermost set of parentheses and then in the next outer set, etc., until all the operations inside the parentheses have been performed.



LOGICAL OPERATIONS OR BOOLEAN OPERATIONS

Boolean operators perform bit manipulations, Boolean operations, or tests on multiple relations. They return a true (nonzero) or false (zero) value to be used in making a decision.

Boolean operators	**Description**	**Syntax**	**Examples**
NOT	Used to perform logical negation on an expression.	*result* = **Not** *expression*	Dim A, B, C, D, MyCheck A = 10: B = 8: C=5 MyCheck = **Not (**A > B**)** ' Returns False.
AND	Used to perform a logical conjunction on two expressions.	*result* = *expression1* **And** *expression2*	Dim A, B, C, D, MyCheck A = 10: B = 8: C=5 MyCheck = A > B **And** B > C ' Returns True
OR	Used to perform a logical disjunction on two expressions.	*result* = *expression1* **Or** *expression2*	Dim A, B, C, D, MyCheck A = 10: B = 8 : C=5 MyCheck = A > B **Or** B > C ' Returns True.

Other logical operators are XOR (exclusive or), EQV (equivalence) and IMP (impliactions).

LIBRARY FUNCTIONS

There are a number of computational processes that require multiple steps to obtain the desired result and are used again and again by different programmers. Examples include computing the square root of a given number, determining the absolute value of an expression, finding the largest value from a set of numbers and so on.

Instead of programming them again and again QBASIC provides a number of **built-in** functions. This buit-in functions are called **intrinsic** functions.

Function	**Description**	**Syntax**	**Example**
ABS	Returns the absolute value of a number.	ABS(numeric-expression)	PRINT ABS(45.5 – 100.0) 'Output is: 54.5
CINT	rounds a numeric expression to an integer	CINT(numeric-expression)	PRINT CINT(12.49), CINT(12.51) 'Output is: 12 13
CLNG	rounds a numeric expression to a long (4-byte) integer	CLNG(numeric-expression)	PRINT CLNG(338457.8) 'Output is: 338458
CSNG	converts a numeric expression to a single-precision value	CSNG(numeric-expression)	CSNG(975.3421515) 'Output is: 975.3422
CDBL	converts a numeric expression to a double-precision value	CDBL(numeric-expression)	CDBL(1 / 3) 'Output is: .3333333333333333
FIX	truncates a floating-point expression to its integer portion	FIX(numeric-expression)	PRINT FIX(12.49), FIX(12.54) 'Output is: 12 12
INT	returns the largest integer less than or equal to a numeric expression	INT(numeric-expression)	PRINT INT(12.54), INT(-99.4) 'Output is: 12 -100
ATN	returns the arctangent of a specified numeric expression	ATN(numeric-expression)	CONST PI=3.141592654 PRINT ATN(TAN(PI/4.0)), PI/4.0 'Output is: .7853981635 .7853981635 PRINT (COS(180 * (PI / 180))) 'Output is: -1 N.B. To convert from degrees to radians, multiply degrees by (PI / 180).
SIN	return the sine of a specified angle in radian	SIN(angle)	
COS	return the cosine of a specified angle in radian	COS(angle)	
TAN	return the tangent of a specified angle in radian	TAN(angle)	

Function	Description	Syntax	Example
EXP	returns e raised to a specified power, where e is the base of natural	*EXP(numeric-expression)* N.B. For EXP, the numeric expression is a number less than or equal to 88.02969.	PRINT EXP(0), EXP(1) 'Output is: 1 2.718282
LOG	returns the natural logarithm of a numeric expression	*LOG(numeric-expression)* N.B. For LOG, any positive numeric expression.	PRINT LOG(1), LOG(EXP(1)) 'Output is: 0 1
MOD	Divides one number by another and returns the remainder.	*numeric-expression1 MOD numeric-expression2* **numeric-expression1, numeric-expression2** – Any numeric expressions. Real numbers are rounded to integers.	PRINT 19 MOD 6.7 'QBasic rounds 6.7 to 7, then divides. 'Output is: 5
SQR	Returns the square root of a numeric expression.	*SQR(numeric-expression)* **numeric-**expression – A value greater than or equal to zero.	PRINT SQR(25), SQR(2) 'Output is: 5 1.414214

STRING MANIPULATIONS

Function	Description	Syntax	Example
INSTR	Returns the position of the first occurrence of a string in another string	*INSTR([start%,]stringexpression1$,stringexpression2$)* N.B. **start%** – Sets the character position where the search begins. If start% is omitted, INSTR starts at position 1. **stringexpression1$ –** The string to search **stringexpression2$** –The string to look for.	a$ = "Microsoft QBasic" PRINT INSTR(1, a$, "QBasic") 'Output is 11
LEFT$ RIGHT$	Return a specified number of leftmost or rightmost characters in a string.	LEFT$(stringexpression$,n%) RIGHT$(stringexpression$,n%) N.B. **stringexpression$** – Any string expression. **n%** – The number of characters to return, beginning with the leftmost or rightmost string character.	a$ = "Microsoft QBasic" PRINT LEFT$(a$, 5) 'Output is: Micro PRINT RIGHT$(a$, 5) 'Output is: Basic
MID$	The MID$ function returns part of a string (a substring). The MID$ statement replaces part of a string variable with another string	*MID$ (stringexpression$, start% [,length%])* *MID$ (stringvariable$, start% [,length%])=stringexpression$* N.B. **stringexpression$** – The string from which the MID$ function returns substring, or the replacement string used by the MID$ statement. It can be any string expression. **start%** – The position of the first character in the substring being returned or replaced. **length%** – The number of characters in the substring. If the length is omitted, MID$ returns or replaces all characters to the right of the start position. **stringvariable$** – The string variable being modified by the MID$ statement.	a$ = "Where is Paris?" PRINT MID$(a$, 10, 5) 'Output is: Paris Text$ = "Paris, France" PRINT Text$ 'Output is: Paris, France MID$(Text$, 8) = "Texas " PRINT Text$ 'Output is: Paris, Texas
LEN	Returns the number of characters in a string or the number of bytes required to store a variable.	*LEN(stringexpression$)* *LEN(variable)* N.B. **stringexpression$** – Any string expression. **Variable** – Any nonstring variable.	a$ = "Microsoft QBasic" PRINT LEN(a$) 'Ouput is 16

Exercise
Write the QBASIC equivalent of the following expressions:

Arithmetic Expression	QBASIC Expression	Arithmetic Expression	QBASIC Expression
a + b / c – d	A + B/C – D	$x = \sqrt{y}$	X = SQR (Y)
$\frac{a + b}{c + d}$	(A + B) / (C + D)	x = | y – a |	X = ABS (Y)
$a^2 - b^2$.	A ^ 2 – B ^ 2	$x = e^{y + a}$	X = EXP(Y + A)
$\frac{ab}{c^2 - d}$	A*B / (C^2 – D)	$\log_{10} (2a - 3b)^2$.	LOG((2*A – 3*B)^2)/LOG (10)

Do the following exercises
Convert the following arithmetic expressions to QBASIC expressions
(a) $a^z - 2.0$ (b) $\frac{ab}{d} - c^2$ (c) $(a^n)^m + a^n a^m$. (d) $a + \frac{1}{1 + \frac{1}{1 + a}}$
(e) $\sqrt{3a^2 + 6b^2}$ (f) $\sqrt{x^2 / (y + z)}$ (g) $\log_e (x + 3y)^3$. (h) $\cos(2x - y) + |x2 + y^2| + e^{xy}$.
(i) $\sqrt{|\sin (a - |b|)|}$ (j) $\left| \sqrt{a - b^2} - \frac{c^2}{\sin (x + y)} \right|$

Convert the following QBASIC expressions to arithmetic expressions
(a) SQR (3*A^2 + * B ^ 2)
(b) SQR (X ^ 2 / (Y + Z))
(c) COS (2*X–Y) + ABS (X^2 + Y^2) + EXP (X * Y)
(d) LOG ((2 * A – 3 * B) ^ 2
(e) A + B / ABS (M – N)
(f) EXP (ABS (Z)) – Y ^ 3 / ABS (X)
(g) COS (LOG (3 * X + Y))
(h) SQR (ABS (SIN (A – ABS (B)))))
(i) ABS (SQR (A – B ^ 2) – C ^ 2 / SIN (X + Y))

Exercise
Calculate the values of each of the following expressions as per precedence rules:

(a) 9 – 6 + 3	(b) 3 ^ 2 + 4 / 5	(c) 2.0 / 4	(d) 3 + 2 ^ 3	(e) (3 + 2) ^ 3
(f) – 4.0 ^ 4	(g) – (4.0 ^ 4)	(h) 12.0 / 3.0 * 2.0	(i) 12 / (3.0 * 2.0)	(j) (2 ^ 3) ^ 2
(k) 2 ^ 3 ^ 2	(l) ((3 + 2) ^ 3) / 5	(m) (3 + 2 ^ 3) / 5	(n) (3. + 2 ^ 3) / 5	

The values are calculated as follows:
(a) 9 – 6 + 3 = 3 + 3 = 6
(b) 3 ^ 2 + 4 / 5 = 9 + 4/5 = 9 + 0 = 9
(c) 2.0 / 4 = .5
(d) 3 + 2 ^ 3 = 3 + 8 = 11
(e) (3 + 2) ^ 3 = 5 ^ 3 = 125
(f) – 4.0 ^ 4 = – (4.0)^4 = – (4.0) (4.0) (4.0) (4.0) = – (256.0) = -256.0
(g) – (4.0 ^ 4) = – (4.0 * 4.0 * 4.0 *4.0) = – (256.0) = – 256.0
Rest of the problems do yourself.

DESIGN OF PROGRAMS

Although computer program differ greatly in purpose and processing, they can all be organized into the following functional modules:

1. **Initialization** establishes initial values for some variables, prints headings, messages etc.
2. **Input** performs input of data required by the program.
3. **Validation** performs validation of input data to detect errors, omissions etc.
4. **Processing** performs computation, or data manipulation.
5. **Output** performs output of data to be provided by the program like printing the output.
6. **Closing procedure** performs procedure to end the execution of the program.

STRUCTURE OF PROGRAM

All computer program can be coded by using only three logical structures (patterns) given below, or a combination of these structures:

1. **Sequential structure**: It consists of one action followed by another. In other words performs operation A and then operation B and so on.
2. **Selective structure**: It consists for a test for a condition followed by two or more alternative paths for the program to follow The program selects one of the paths depending on the test of the condition.
3. **Repetitive or Iterative structure**: Here an operation, or a set of operations, is repeated as long as some condition satisfied.

SEQUENTIAL STRUCTURE

1. **Assignment statement**

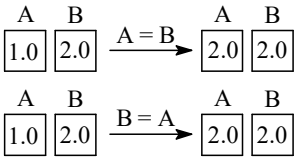
The assignment statement is used to assign values to variables and has the form of *variable = expression.*

Where expression may be a

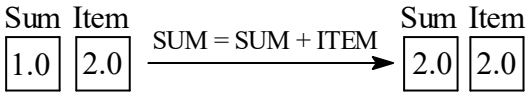
(a) constant (b) a variable to which a value has been assigned

(c) a formula which the computer can evaluate.

e.g.



e.g.



Examples of Rules of assignments

1. In the left-hand side of the '=' symbol a variable should be there.
2. No value, constant, or expression can be on the left-hand side of the '=' symbol.
3. In the right-hand side any value, constant or valid expression can be placed.
4. The type of variable and the expression or constants in both the sides should be same.

e.g X = 2 Y = (N + 4) / 3 K = 1. / 3. + 1./3 IPT = "Institute"

Hence, the following are not valid QBASIC assignment statements for the reasons indicated:

15 = N	Variable N is on the right instead on the left of the = sign.
X + 4.3 = 3.14	Numeric expression should not appear to the left-hand side of = sign.
STRING = 4 & 7	4 & 7 is not a legal expression.
A = B = 1	B = 1 is not a valid expression
DIM A AS INTEGER A = "abcd"	When A is declared as integer in the first line a string constant "abcd" cannot be assigned to A.

2. **Input and output statements**

The computer can manipulate the data only it is available to its memory. Data can be input in the memory either within the program or by the user during program execution.

Data can be input in the memory by the following statements and the result is shown beside:

Sl. No.	Syntax	Example	Displayed result during program execution
1	INPUT variablename	INPUT ipt	_
2	INPUT "prompt"; variablename	INPUT "First Data:"; X	First Data: ?_
3	INPUT "prompt", variablename	INPUT "First Data:", X	First Data: _

Sl. No.	Syntax	Example	Displayed result during program execution
4	INPUT variable1, variable2	INPUT X, Y	?12 , 20 Press enter
5	INPUT "prompt", variable1, variable2	INPUT "X=",X, Y	X= 12, 20
6	INPUT "prompt"; variable1, variable2	INPUT "X=";X, Y	X= ? 12, 20

Data can be output from the memory by the following statements and the result is shown beside:
PRINT writes data to the screen or to a file.
LPRINT prints data on the printer LPT1.
The general syntax is
PRINT [expressionlist] [{; | ,}]
LPRINT [expressionlist] [{; | ,}]
_ expressionlist A list of one or more numeric or string expressions to print.
_ {; | ,} Determines where the next output begins:; means print immediately after the last value., means print at the start of the next print zone. Print zones are 14 characters wide.

Sl. No.	Syntax	Example	Displayed result during program execution
1	PRINT variable1, variable2	PRINT A, X, Z	25 10 12
2	PRINT variable1; variable2	PRINT A; X; Z	25 10 12
3	PRINT string constant	PRINT "Institute"	Institute
4	PRINT string1 + string2	PRINT "ab" + "cd"	Abcd
5	PRINT string, variable	PRINT "X=",X	X= 20
6	PRINT string; variable	PRINT "X=";X	X= 20

STOP
The STOP statement is used to halt the program execution. But the program execution will not terminate here and the control will not be given to the user. Press F5 to continue.

END
This statement terminates a program and give the control to the user.
END must be the last statement of a program.

Example: Write the algorithm, flow chart and the program code.

1. Write a program that accepts input temperature in degree Farenheit, converts it to degree Centigrade, and outputs the temperature in degree Centigrade.
2. Write a program that reads values for the three sides of a triangle, calculates its perimeter and its area, and outputs these values.
3. Write a program that inputs the radius of a circle, computes and outputs (a) the area of the circle, (b) the area of the largest square contained within th ecircle, and (c) the ratio of (a) to that of (b). Use the constant, PI, the value 3.1416.
4. Write a program to find the surface area (SUR), volume (VOL) of a box with dimensions a,b,c where SUR = 2(ab +bc + ca) and VOL =abc.
5. Assuming $a_1b_2 - a_2b_1 \neq 0$, the solution of the linear equations $a_1x + b_1y = c_1$ and $a_2x + b_2y = c_2$ is given by

   $$x = \frac{b_2c_1 - b_1c_2}{a_1b_2 - a_2b_1} \qquad y = \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1}$$

   Write a program that reads values of a1, b1, c1, a2, b2, c2 and calculates and prints the solution pair x and y.
6. Suppose that a car starts from rest and has constant acceleration for t seconds. The final velocity v and the distance traveled d by the car is given by the formulae:
   v = ft and d = ½ ft^2.
   Write a program that reads f and t, and prints t, d, and v.
7. Write a program to convert a given measurement in feet to an equivalent one in (i) yards, (ii) inches, (iii) centimeters, (iv) meters. (1yard = 3 ft, 1 foot = 12 inches, 1 inch = 2.54 centimeter, 1 meter = 100 centimeter)

2. SELECTIVE STRUCTURE

The selective structure consists of a test for a condition followed by alternative paths which the program can follow. Selection among alternative paths is programmed with the IF statements.

Relational expression

If A and B represents two *numeric or string constants* they can be combined by the following six *relational operators*:

Relational operator	Application in QBASIC	Meaning
<	A < B	A is less than B
< =	A < = B	A is less than or equal to B
=	A = B	A is equal to B
< >	A < > B	A is not equal to B
>	A > B	A is greater than B
> =	A > = B	A is greater than or equal to B

- The relational operator can be used to compare numerical values. If a relational operator is present in between two numerical expression then the numerical expression will be executed first.
  e.g. Assuming the two *single* values X = 20.0, Y = 4.0 and the integer variable M = 5 then
  X + 5 > M * Y → 20.0 + 5 > 5*4.0 → 25.0 > 20.0 → Answer is "True"
- The relational operators can be used to compare two string expressions.
  e.g. Let us assume three string variables X = "a", Y = "b", Z = "A"
  X > Y → True because ASCII code of "a" is 97 and "b" is 98
  X > Z → True because ASCII code of "a" is 97and "A" is 65

Logical expression

Logical operators used in QBASIC

Symbol	Expression	Meaning	Logical value
NOT	NOT A	Negation	**True** if and only if A is **False**.
AND	A AND B	Conjunction	**True** if and only if, A and B are both **True**.
OR	A OR B	Disjunction	**True** if and only if, at least one is **True**.

The following two truth tables summarizes the logical values for these expressions:

Truth Table - I

A	NOT A
T	F
F	T

Truth Table -II

A	B	A AND B	A OR B
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

Precedence of relational and logical operations

Precedence 1: Parentheses rules
Precedence 2: Numeric expressions
Precedence 3: Relational operations
Precedence 4: Logical operations

If more than one logical operations are in one expression then the execution will start from the left hand side.

e.g. Say A =3, B =4 and C =12 and the expression will be executed as follows:

A > B AND (A < C OR B < > C) OR NOT (A = B)
⇒ A > B AND (True OR B < > C) OR NOT (A = B)
⇒ A > B AND (True OR True) OR NOT (A = B)
⇒ A > B AND True OR NOT (A = B)
⇒ A > B AND True OR NOT (False)
⇒ A > B AND True OR True
⇒ False AND True OR True
⇒ False OR True
⇒ True

GOTO statement

The GOTO statement is used to jump from one line to a another line.
Syntax: GOTO *line*
_ *line* The label or number of the line to execute next.

Precaution: Indiscriminate use of GOTO statement may make a program difficult to debug and understand therefore, GOTO should be used to caution.
e.g.

```
10  LARGER = V2
    SMALLER = V3
    GOTO 10
```

IF Statements
Executes a statement or statement block depending on specified conditions.
Syntax I (Logical IF Statement)

```
IF condition THEN statements [ELSE statements]
```

Syntax II (Block IF statement)

```
IF condition1 THEN
   [statementblock-1]
[ELSEIF condition2 THEN
   [statementblock-2]]...
[ELSE
   [statementblock-n]]
END IF
```

condition1	Any expression that can be evaluated as
condition2	true (nonzero) or false (zero).
statementblock-1	One or more statements on one or more lines.
statementblock-2	
statementblock-n	
statements	One or more statements, separated by colons.

Example of Logical IF :

```
IF  (num  <  100 ) PRINT  num
IF (num  < 100 ) PRINT  num  ELSE  PRINT "No number"
```

Example of block IF:

```
DIM  v1 AS INTEGER, v2 AS INTEGER, larger AS INTEGER
INPUT  v1,v2
IF (v1 > v2) THEN
        larger = v1
ELSE
        larger = v2
END IF
PRINT larger
END
```

Example of nested block IF:

```
DIM v1 AS INTEGER, v2 AS INTEGER
INPUT v1, v2
IF  (v1  >  v2) THEN
        PRINT "Semester 1"
ELSE
        IF (v2 > v1) THEN
                PRINT "Semester 2"
        ELSE
                PRINT "Semester 3"
        END IF
END IF
END
```

Nest

IF Block

Exercises

Problem1. Write a program to find the largest of the three numbers.
Solution:

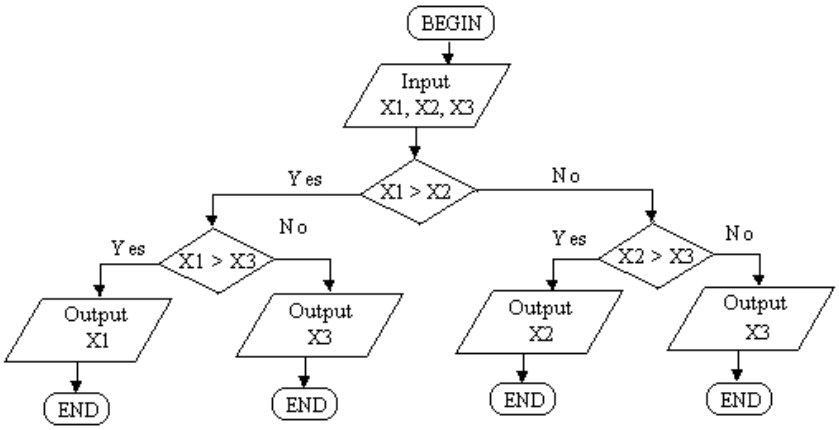


Fig. Flow chart

Program coding:

```
REM Program to find the largest of 3 numbers
DIM x1 AS SINGLE, x2 AS SINGLE, x3 AS SINGLE, largest AS SINGLE
INPUT x1,  x2,  x3
IF (x1 > x2) THEN
        IF (x1 > x3) THEN
                largest = x1
        ELSE
                largest = x3
        END IF
ELSE
        IF (x2 > x3) THEN
                largest = x2
        ELSE
                largest = x3
        END IF
END IF
PRINT largest
END
```

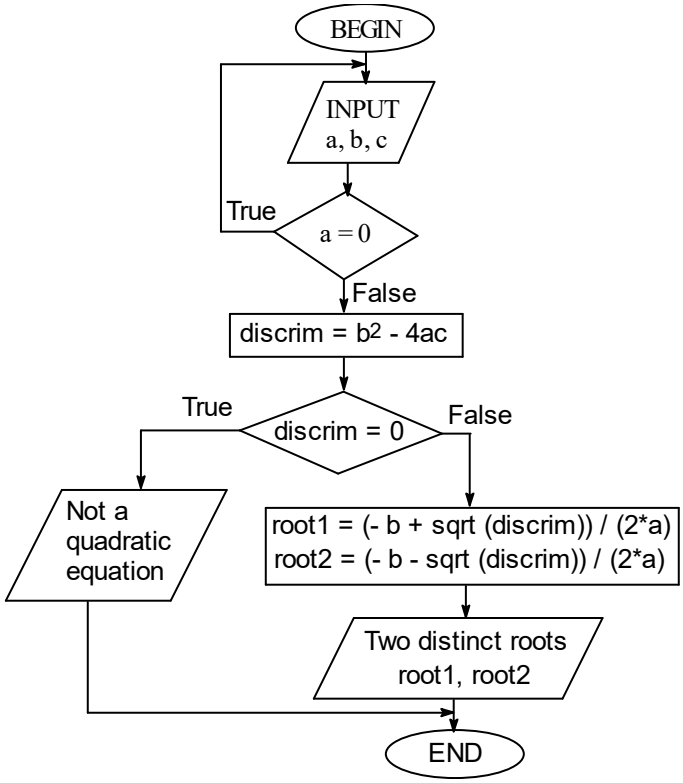
*Problem 2: Write a program that reads the coefficients **a**, **b** and **c** (**a** ≠ 0) of the quadratic equation $ax^2 + bx + c = 0$ and computes its real roots. The roots of the quadratic equation are given by*

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

N.B. When the discriminant $(b^2 - 4ac) > 0 \Rightarrow$ two real roots are present
When the discriminant $(b^2 - 4ac) = 0 \Rightarrow$ two identical real roots are present and are equal to $(-b/2a)$
When the discriminant $(b^2 - 4ac) < 0 \Rightarrow$ no real roots are present

Algorithm

1. Begin : Declare a, b, c, discrim, root, root1, root2 as SINGLE
2. Read a, b and c.
3. If a = 0 then give the message "Not a quadratic equation" and GOTO statement 2 to read data again.
4. Calculate discriminant from the formula discrim = b^2 – 4*a*c
5. If discrim < 0 then give the message " No real roots" and END the program here.
6. If discrim = 0 then give the message "Two identical roots" and give the value by using root = – b /(2*a) and end the program here.
7. If discrim > 0 then give the message "Two distinct roots"
   root1 = (– b + SQR(discrim)) / (2*a) root2 = (– b – SQR(discrim)) / (2*a)
   end the program here.



Program coding

```
        DIM  a AS SINGLE, b AS SINGLE, c AS SINGLE
        DIM  discrim AS SINGLE, root AS SINGLE, root1 AS SINGLE, root2 AS SINGLE

10      INPUT a, b, c

        REM Checking for A <> 0
        IF  ( a = 0 ) THEN
                PRINT "Not a quadratic equation and  A cannot be zero. Input data again."
                GOTO 10          ' Line 10 is the input line.
        END IF

REM Calculate the discriminant
        discrim = b^2 – 4*a*c
REM  Discriminant is less than zero
        IF discrim < 0 THEN
                PRINT "No real root. The program ends here"
                GOTO 99          ' Line 99 is the last line of this program.
        END IF
REM Discriminant is equal to zero
        root = – b / (2 * a)
        IF discrim = 0 THEN
                PRINT "Two identical roots", root
                GOTO 99          ' Line 99 is the last line of this program.
        END IF
REM Discriminant is greater than zero
        root1= (– b + SQR (discrim) ) / (2 * a)
        root2= (– b – SQR (discrim) ) / (2 * a)
        PRINT "Two distinct roots :", "Root 1="; root1, "Root2="; root2

99      END
```

EXERCISES

1. If I = 2, J = 3, K = 6 then what values do the following expressions have?

(a) I > J AND I * J <= K
(b) I * J >= K AND I > J
(c) NOT I > J AND I * J >= K
(d) I > J OR I * J >= K
(e) I > J AND (I <= K OR I * J <= K)

2. If I = 1 and J = –1 then what values the following logical expression have?

(a) ((I > 0) AND (J < 0) OR (NOT (I > 0) AND NOT (J < 0)
(b) ((I > 0) AND NOT (J < 0) OR (NOT (I > 0) AND (J < 0)

3. Write a logical IF statement that prints YES if FOOD is between –1 and 1 (i.e. $-1 \leq FOOD \leq 1$)

4. What will be the final value of NERD at the end of the each program fragment?

(a)
```
NERD = 5 : JOCK = 10
IF (3*NERD < JOCK ) THEN NERD = NERD + 2
NERD = NERD + 3
```

(b)
```
    NERD = 5 : JOCK = 10
    IF (2* JOCK <= 3* NERD) THEN GOTO 10
    NERD = NERD +1
    GOTO 20
10  NERD = JOCK
20  NERF = NERD + JOCK
```

5. Write a program to convert to Fahrenheit if the temperature input is in Centigrade and convert to Centigrade if the input temperature is in Fahrenheit scale.

6. Write a program to calculate the parts of alcohols (one of higher and other of lower strength) required to prepare a alcohol of desired strength. (Hints. Use aligation method.)

7. Write a program that calculates the Reynold's number from diameter (DIA), velocity (VEL), density of liquid (DENSITY) and viscosity of the liquid (VISCOSITY) from the formula

REYNUM = (DIA*VEL*DENSITY)/VISCOSITY

If REYNUM ≤ 2000 give output
"The liquid is flowing in a stream line flow. Reynold's number is:" REYNUM
If REYNUM ≥ 4000 give output
"The liquid is flowing in a turbulent flow. Reynold's number is:" REYNUM
If 2000 ≤ REYNUM ≤ 4000 give output
"The liquid is flowing in transition state. Reynold's number is:" REYNUM

3. REPETITIVE STRUCTURE

In a *repetitive* structure a set of statements are executed many times, but that set of statements will appear only once in the program.

Any repetitive structure contains an **entry point**, a **repetition** or **loop body**, and an **exit point**, as illustrate in the figure:

The number of repetitions in a repetitive structure can be either **condition-controlled** or **counter-controlled**.

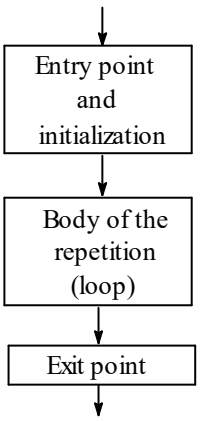


Fig. Repetitive structure

IF Loop

In the IF loop the number of repetition can be condition controlled.

Program fragment

```
100     INPUT  A, B
        SUM = A + B
        PRINT A, B, SUM
        GOTO 100
        STOP
        END
```

In the above program fragment there is no termination point. So this program will iterate (repeat) for an infinite time. To stop this loop a condition may be provided.

For example if the A = 9999 then the program will stop. This A = 9999 is a termination condition.

The same program fragment can be written in the following manner:

```
100     INPUT  A, B
        IF ( A = 9999) THEN GOTO 99
        SUM = A + B
        PRINT A, B, SUM
        GOTO 100
99      STOP
        END
```

Counter controlled by IF loop

To control the number of times a loop is executed, a **counter** may be used whose value is changed by 1 every time then loop is executed. When the counter attains a predetermined value, the loop execution is terminated.

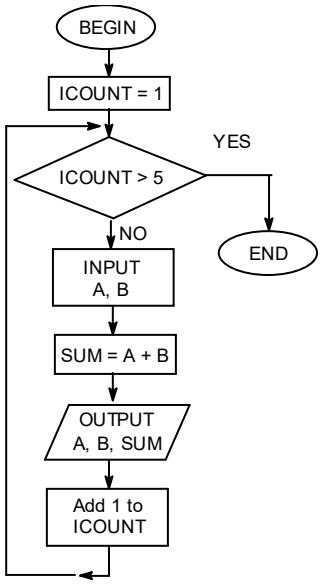
Example:

```
        DIM ICOUNT AS INTEGER
        DIM A AS INTEGER, B AS INTEGER
        DIM SUM AS INTEGER

100     IF ( ICOUNT  > 5 ) THEN GOTO 99
        INPUT  A, B
        SUM = A + B
        PRINT A, B, SUM

        ICOUNT  = ICOUNT + 1
        GOTO 100

99      END
```



Counter controlled by FOR … NEXT loop

QBASIC can handle counter-controlled repetitive structures, or program loops, by a single statement called the FOR .. NEXT loop. It repeats a block of statements a specified number of times.

Syntax

```
FOR counter = start TO end [STEP increment]
   [statementblock]
NEXT [counter [,counter]...]
```

_ counter — A numeric variable used as the loop counter.
_ start and end — The initial and final values of the counter.
_ increment — The amount the counter is changed each time through the loop.

Example:

```
FOR i = 1 TO 15
        PRINT i
NEXT i
FOR i = 2 to – 6 STEP – 3
        PRINT i
NEXT i
```

In the first loop the loop will be executed 15 times.
In the second loop the loop will be executed thrice
1st loop: i = 2
2nd loop: i = 2 + (–3) = – 1
3rd loop: I = – 1 + (– 3) = – 4

- Where no *increment* value is given the increment is taken as 1. [(i.e. FOR i = 1 TO 15 STEP 1]
- When the value of '**i**' will exceed the **end** value the loop will stop.
- **Start**, **end**, **increment** values may be integer or real value or variables.

Nested FOR … NEXT loop

Several FOR .. NEXT loop may be used within the same program. They may follow one another, or they may be nested, i.e. one loop may remain within another.

The inner most loop will be executed first, then the next outer loop, and so on.

Example

```
DIM PRIN AS INTEGER, RATE AS INTEGER, YEAR AS INTEGER
DIM INTEREST AS SINGLE
FOR PRIN = 1000 TO 10000 STEP 1000
        FOR RATE = 10 TO 15
                FOR YEAR = 1 TO 3
                        INTEREST = PRIN * (RATE / 100.) * YEAR
                        PRINT PRIN, RATE, YEAR, INTEREST
                NEXT YEAR
        NEXT RATE
NEXT PRIN
END
```

Inner loop | Middle loop | Outer most loop

EXERCISES

Q1. Write a program that reads an integer N and prints the sum of the following:

(a) $1 + 2 + 3 + \ldots..+ N$
(b) $1^2 + 2^2 + 3^2 + \ldots + N^2$.
(c) $1^3 + 2^3 + 3^3 + \ldots + N^3$.
(d) $2 + 4 + 6 + 8 + \ldots + 2N$
(e) $1 + 3 + 5 + 7 + \ldots + (2N+1)$

Solution of (a)

```
DIM N AS INTEGER, SUM AS INTEGER
INPUT N
SUM = 0
FOR I = 1 TO N
        SUM = SUM + I
NEXT I
PRINT SUM
END
```

Q2. Write a program that reads an integer N and computes factorial of N (i.e. N!)
[Hints: PROD = 1*2*3*….* N]
Q3. To fit a straight line Y = mX + C through a set of n points (x_1,y_1) (x_2, y_2), (x_3, y_3),…,(x_n, y_n), the following formulae are used to determine the values of m and c :

$$m = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} y_i - \left(\sum_{i=1}^{n} x_i\right)^2} \qquad c = \frac{1}{n}\left[\sum_{i=1}^{n} y_i - m\sum_{i=1}^{n} x_i\right]$$

Write a program that first reads the number of points n, and then reads n pairs of values (x_i,y_i), and compares m and c.

FUNCTIONS AND SUBROUTINES
The functions and subroutines subprograms with which a program can be broken up into parts.
Function Subprogram
The function subprogram is the method by which a programmer can define function whose value cannot be specified by a single expression. The function will be called from the main program. Only one value will be returned by the function.
Syntax

```
FUNCTION name [(parameterlist)]
        [statementblock
        name = expression
        [statementblock]
END FUNCTION
```

_ *name* The name of the function and the data type it returns, specified by a data-type suffix (%, &, !, #, or $).
_ *parameterlist* One or more variables that specify parameters to be passed to the function when it is called

Example

Main program **PROG25.BAS**	Function **PROG25.BAS:MAX**
DIM X1 AS SINGLE, X2 AS SINGLE DIM X3 AS SINGLE INPUT X1, X2, X3 PRINT **MAX**(X1, X2, X3) END	FUNCTION **MAX** (A AS SINGLE, B AS SINGLE, C AS SINGLE) DIM LARGE AS SINGLE LARGE = A IF B > LARGE THEN LARGE = B IF C > LARGE THEN LARGE = C MAX = LARGE END FUNCTION

Subroutine
When more than one value is required to be returned to the main program then subroutine statement is writte.
Syntax

```
SUB name[(parameterlist)]
  [statementblock]
END SUB
```

_ name The name of the SUB procedure, up to 40 characters long, with no data type suffix.
_ parameterlist One or more variables that specify parameters to be passed to the SUB procedure when it is called:

Main program **PROG25.BAS**	Function **PROG25.BAS:LARGEST**
DIM X1 AS SINGLE, X2 AS SINGLE DIM X3 AS SINGLE, MAX AS SINGLE INPUT X1, X2, X3 CALL **LARGEST** (X1, X2, X3, MAX) PRINT MAX END	SUB **LARGEST** (A AS SINGLE, B AS SINGLE, C AS SINGLE, LARGE AS SINGLE) LARGE = A IF B > LARGE THEN LARGE = B IF C > LARGE THEN LARGE = C END SUB

Example
Write a program that reads a temperature and a letter "C" or "F" and convert it to the other scale.

The main program

```
REM This program reads a temperature and an alphabet "C" or "F" and
REM then it converts the temperature to the other scale
DIM TEMP AS SINGLE, UNIT AS STRING
DIM CENTIGRADE AS SINGLE, FAHRENHEIT AS SINGLE
INPUT "Give the temperature: "; TEMP
INPUT "Write C if it is Centigrade or F if it is in Fahrenheit:"; UNIT
IF UNIT = "C" THEN
        PRINT TEMP,"deg C is equivalent to ", FAHRENHEIT(TEMP), "deg F"
ENDIF
IF UNIT = "F" THEN
        PRINT TEMP "deg F is equivalent to ", CENTIGRADE(TEMP), "deg C"
ENDIF
END
```

The functions

```
FUNCTION FAHRENHEIT (X AS SINGLE)
        FAHRENHEIT = X*9/5 + 32
END FUNCTION

FUNCTION CENTIGRADE (X AS SINGLE)
        CENTIGRADE = (X – 32)*5/9
END FUNCTION
```

Same program but by using subroutine
*The main program************

```
REM This program reads a temperature and an alphabet "C" or "F" and
REM then it converts the temperature to the other scale
DIM TEMP AS SINGLE, UNIT AS STRING
DIM CENTIGRADE AS SINGLE, FAHRENHEIT AS SINGLE
INPUT "Give the temperature: "; TEMP
INPUT "Write C if it is in Centigrade or F if it is in Fahrenheit:"; UNIT

CALL CONVERT (TEMP, UNIT, VALUE)

IF UNIT = "C" THEN
        PRINT VALUE, "deg F"
ELSE
        PRINT VALUE, "deg C"
END IF
END
```

*The sub-routine program**********

```
SUB CONVERT (TEMP AS SINGLE, UNIT AS STRING, VALUE AS SINGLE)
IF UNIT = "C' THEN
        VALUE = X*9/5 + 32
ELSE
        VALUE = (X – 32)*5/9
ENDIF
END SUB
```

FORMATTING

LOCATE

By using LOCATE statement the display of text can be started from a predefined position on the screen.

LOCATE moves the cursor to a specified position on the screen

LOCATE [row%] [,[column%]

– row% is the row number (maximum 25 rows)
– column% is column number (maximum 80 columns)

Example

```
CLS
LOCATE 2, 10
PRINT "Enter X value"
LOCATE 2, 25
INPUT "", j
LOCATE 3, 10
PRINT "The value you have entered is "; j
```

GRAPHICS

SCREEN statement

SCREEN statement sets the screen mode and other characteristics of the screen. The type of graphical display depends on the screen mode. Here we will use screen mode 9. The characteristics of screen mode 9 are as follows:

SCREEN 9:

640 x 350 graphics
80 x 25 text format
16 colors assigned to 4 attributes (64K adapter memory), or
64 colors assigned to 16 attributes (more than 64K adapter memory)

COLOR statement

COLOR sets the screen display colors. It can set the foreground color (i.e. the color of the characters and lines) and (or) the color of background.

Example

```
SCREEN 9
COLOR 4, 7          ‘ This will set the foreground color in red and background color in white
PRINT "Institute"   ‘ Institute will be written in red color against a white background
COLOR 1,7           ‘ This will set the foreground color in blue and background color in white
PRINT "Institute"   ‘ Institute will be written in blue color against a white background
COLOR 1,0           'This will set the foreground color in blue and background color in black
PRINT "Institute"   ‘ Institute will be written in blue color against a black background
```

QBASIC color values	Name of the color	QBASIC color values	Name of the color
0	Black	8	Gray
1	Blue	9	Light Blue
2	Green	10	Light Green
3	Cyan	11	Light Cyan
4	Red	12	Light Red
5	Magenta	13	Light Magenta
6	Brown	14	Yellow
7	White	15	High-intensity white

LINE statement

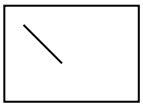
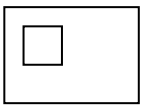
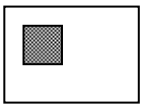
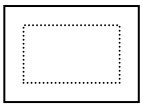
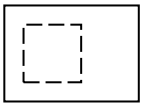
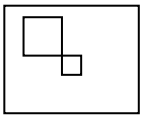
LINE statement draws a line or rectangle on the screen.

Syntax

LINE [(x1!,y1!)] – (x2!,y2!) [,[color%] [,[B | BF] [,style%]]]

STEP	Specifies that coordinates are relative to the current graphics cursor position.
(x1!,y1!)	The screen coordinates of the start of the line and of (x2!,y2!) the end of the line.
color%	A color attribute that sets the color of the line or rectangle. The available color attributes depend on graphics adapter and the screen mode set by the SCREEN statement.
B	Draws a rectangle instead of a line.
BF	Draws a filled box.
style%	A 16-bit value whose bits set whether or not pixels are drawn. Use to draw dashed or dotted lines.

Examples

Sl	Statement	Comment	Display
1	SCREEN 9	Text mode 25 rows and 80 cols. Graphics 640 x 350	
2	LINE (10, 10)-(100, 100)	'Draws a line from (10,10) to (100,100)	
3	LINE (10, 10)-(100, 100),4	'Draws a line from (10,10) to (100,100) with red color.	
4	LINE (10, 10)-(100, 100),4, B	'Draws a box from (10,10) to (100,100) with red color.	
5	LINE (10, 10)-(100, 100),4, BF	'Draws a box from (10,10) to (100,100) filled with red color.	
6	LINE (10, 10)-(100, 100), , BF	'Draws a box from (10,10) to (100,100) filled with default color.	
7	LINE (10, 10)-(100, 100), 4, B, 1	'Draws a box from (10,10) to (100,100) with dotted border.	
8	LINE (10, 10)-(100, 100), 4, B, 3	'Draws a box from (10,10) to (100,100) with dashed border.	
9	LINE – (150, 150), 4, B	'Draws a box from previous point (i.e. 100,100) to (150,150).	

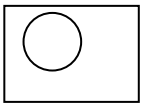
CIRCLE

Draws a circle or ellipse on the screen.

CIRCLE (x!,y!), radius![,[color%] [,[start!] [,[end!] [,aspect!]]]

STEP	Specifies that coordinates are relative to the current graphics cursor position.
(x!,y!)	The coordinates for the center of the circle or ellipse.
radius!	The radius of the circle or ellipse in the units of the current coordinate system, determined by the most recent SCREEN, VIEW, and WINDOW statements.
color%	A color attribute that sets the circle's color. The available color attributes depend on your graphics adapter and the screen mode set by the most recent SCREEN statement.
start!	The starting angle for the arc, in radians.
end!	The ending angle for the arc, in radians.
aspect!	The ratio of the length of the y axis to the length of the x axis, used to draw ellipses.

Examples

Sl	Statement	Comment	Display
1	SCREEN 9	Text mode 25 rows and 80 cols. Graphics 640 x 350	
2	CIRCLE (100, 100), 50	'Draws a circle with center (100,100) and radius of 50 pixels.	
3	CIRCLE (100, 100), 50, 4	'Draws a red circle with center (100,100) and radius of 50 pixels.	
4	CIRCLE (100, 100), 50, 4, 0, 1.57	'Draws a red arc with center (100,100) and radius of 50 pixels starting from 0 radian to π/2 radian	
5	CIRCLE (100, 100), 50, 4, , , 0.5	'Draws a red ellipse with center (100,100) and y axis / x axis ratio of 0.5.	

Q4. 500 mg of a drug is administered orally to a patient. Blood samples were taken from every hour. Concentration of the drug (c) in the plasma is plotted against time (t) to obtain a curve. Write a program to approximate the area under the curve from t =0 to t=24 hours, using the trapezoidal rule.
The area of a trapezium = ½ (a + b)h

IMPORTANCE OF COMPUTER IN PHARMACY

The development of computer is probably the greatest development in the latter part of 20th century. The concept of 'Internet' in the fag end of this century further revolutionized our life-style and with it, obviously, the profession of pharmacy.

Computer, the marvelous gadget, has spread its tentacle to all the nook and corners of pharmacy profession from the desk of academics to the bed of patients, from the test tube of the researchers to the counter of retail shop.

Computer in pharmaceutical research

To invent a new drug-molecule thousands of molecules are synthesized and their biological actions are determined pharmacologically on experimental animals. First of all it is very tedious and time consuming and costly affair as well to synthesize so many compounds and evaluating their gross pharmacological effects. *Computer aided drug design* came to play in this field. With computer several possible drug molecules are drawn on the screen of the computer and their 3-dimensional structures are constructed by the computer programs. Then it compares between all the structures and converge to the structure having minimum energy level. These 3D molecular structures are then fitted to the receptor structures already available. Some computer programs can also show the possible biological effects of those molecules with respect to a standard drug without a single experiment. After selecting a prospective candidate (drug molecule) the molecule is synthesized and the pharmacological experiment has been carried out.

In every step of research computer became an indispensable aid. e.g. analysis of the data, storing it for future retrieval.

The instruments/ equipment like HPLC, GC, NMR, Mass etc. all are operated automatically by computer. To develop a dosage form its pharmacokinetic parameters, statistical analysis, drawing the graphs, feathering of biexponential plasma concentration-time curves, determination of area under the curve etc. require specific softwares.

Computer in industry

In the production department the sterile units, the tableting, capsuling, packaging equipments are now-a-days are fully automated with the help of computer. To assure the quality human error is replaced by the accuracy of computers with the present day equipments.

In the quality control department the HPLC, GC, NMR, Mass, IR-spectrophotometer, UV-spectrophotometer all are assisted by computer to identify a compound quickly with the database already provided by the suppliers

In order to obtain Quality Assurance, GMP, ISO 9000 series certificates the industries require extensive documentation which simply is impossible without computers in different department interconnected by Local Area Network (LAN). Inventory control, personnel management are supported by computer network.

Dissemination of information about a company, its products and other relevant information became very easy by publishing through websites in the internet. e.g. Nearly all the big industries in India have their own websites. E-mail has revolutionized the concept of mailing message, keeping contact with customers.

Marketing of a product has got a boost from the concept of Internet marketing. India has also several such portals e.g. www.rediff.com, www.indiaonline.com etc.

Finally, the marketing and distribution of pharmaceutical products is supported extensively by computers. Advanced countries use bar codes to identify trade packages. Record keeping of wholesaler, retailer and two-way flow of information greatly improves the speed and efficiency of distribution.

Computer in pharmacy education

Mode of teaching and learning are slowly but steadily heading toward a system which will not depend on time or distance. Universities are reorienting themselves to give degree to any students at any time and at any where in the world. This has become possible only through Internet.

Multimedia packages supplied in CD-ROM has become the latest teaching aid. Through audio-visual interaction, easy searching facility through the databases is slowly supplementing the book. Remington, USP Martindale Extra Pharmacopoeia - these books are also published in CD-ROM.

Drug information services

Computers are now an essential tool for pharmacists involved in drug-information retrieval and storage. On-line databases such as MEDLINE , TOXILINE, AIDSLINE any may more databases are available in "www.medscape.com" website. Other better known databases of interest to pharmacists are CAS (*Chemical Abstracts Services*) Online, International Pharmaceutical Abstracts, and SCISEARCH (*Science Citation Index and Current* and *Current Contents*).

With the advent of CD-ROM technology large amount of information is provided in this mode. Micromedex provides DRUDEX, IDENTIDEX, POISINDEX, EMERGINDEX and the current edition of *Martindal's Extra Pharmacopoeia* on a single compact disk, updated quarterly.

Computer application in pharmacology

Pharmacological experiments usually involves animals and most often they are sacrificed. To reduce the number of animal experiments, programs are created that simulates the experiments.

Paired t-test between two sets of data can be carried out in spreadsheet programs (like MS Excel) or can be programmed in BASIC or some other programming languages.

Computer programming in biopharmaceutics

Bipharmaceutics is concerned with the release of drug from the dosage form in-vitro or in-vivo and then absorption and bioavailability and bioequivalence.

1. Data obtained from the **dissolution study** of an oral solid dosage form requires computer for quick and accurate data analysis from the raw data. The raw data includes only the absorption values of the sample solutions taken from the dissolution chamber from time to time.
   (i) The input data are time and corresponding absorption values.
   (ii) The absorption values are converted to concentration by the equation of the standard curve.
   (iii) Calculation is done to convert this concentration into the "amount of drug released" .
   (iv) Graphs may be drawn taking "time" in X-axis and "amount released" in Y-axis.
   (v) Fitting the release profile into some models like "zero order release", "first order release", "Higuchi model", "Hixon-Crowel model" etc.
2. To determine the bioavailability of bioequivalence of two dosage forms of the same drug the "area under the curve(AUC)" of the :"plasma concentration profile" is required. To determine this area under the curve a special computer program is required.
3. For compartmental analysis the "plasma concentration data" is fitted to some equation and the number of hypothetical compartments are thus calculated.
4. To determine the values of A, B, α, β from the equation

$$C = A e^{-\alpha t} + B e^{-\beta t}.$$

feathering technique is followed which require special programming.

All the above problems of biopharmaceutics and phamakokinetics require intensive and complex calculations. Hence, computer programming for those specific problems are required for accurate and speedy calculation. The program may include graphical representation for better visual understanding.

Computer in Hospital Pharmacy

The following list of reports or functions are commonly found in a modern hospital computer system:

1. Patient-record database management
2. Medication order entry.
3. Drugs labels and fill lists.
4. Lists for intravenous solutions and admixtures.
5. DUR / DUE reports.

6. Drug - usage and cost reports.
7. Drug-information literature retrieval.
8. Drug-therapy problem detection.
9. Drug-therapy monitoring.
10. Drug-formulary search and update.
11. Purchasing and inventory control.
12. Billing for third-party services.

Increasingly, hospital pharmacy computer systems are becoming interconnected with systems in other hospital departments.

The future

As the hardware and software technologies advances, in near future individualization of dosage regimen will be possible. The modern sensors will sense the parameters in the body and the drug delivery system will supply the dose accordingly - hence, highly individualization of dosing may be possible.

In addition, emerging drug delivery systems, including implantable pumps and similar devices, are expected to depend heavily on imbedded microprocessors to both sense the patient's need for the drug and control drug administration.

WEBSITES RELATED TO PHARMACY

CHAPTER – 2
Application of computer in pharmacy

Syllabus:
Introduction to various uses of computer in pharmaceutical research and development, industries, authorities, education and hospitals.

The development of computer is undoubtedly the greatest development in the latter part of 20th century. The concept of 'Internet' in the beginning of this century further revolutionized our life-style and with it, obviously, the profession of pharmacy.
Computer, the marvelous gadget, has spread its tentacle to the entire nook and corners of pharmacy profession from the desk of academics to the bed of patients, from the test tube of the researchers to the counter of retail shop.

Computer in pharmaceutical research

1. Computer aided drug design
To invent a new drug-molecule thousands of molecules are synthesized and their biological actions are determined pharmacologically on experimental animals. First of all it is very tedious and time consuming and costly affair as well to synthesize so many compounds and evaluating their gross pharmacological effects. *Computer aided drug design* came to play in this field. With computer several possible drug molecules are drawn on the screen of the computer and their 3-dimensional structures are constructed by the computer programs. Then it compares between all the structures and converge to the structure having minimum energy level. These 3D molecular structures are then fitted to the receptor structures already available. Some computer programs can also show the possible biological effects of those molecules with respect to a standard drug without a single experiment. After selecting a prospective candidate (drug molecule) the molecule is synthesized and the pharmacological experiment has been carried out.

2. Computer with analytical instruments
In every step of research computer became an indispensable aid. e.g. analysis of the data, storing it for future retrieval. The instruments/ equipment like HPLC, FTIR, GC, NMR, Mass etc. all are operated automatically by computer.

3. Others
To develop a dosage form its pharmacokinetic parameters, statistical analysis, drawing the graphs, feathering of biexponential plasma concentration-time curves, determination of area under the curve etc. require specific softwares.

Computer in industry

1. In production department
In the production department the sterile units, the tableting, capsuling, packaging equipments, now-a-days, are fully automated with the help of computer. To assure the quality human error is replaced by the accuracy of computers with the present day equipments. Operational speed increased tremendously.

2. In quality control department
In the quality control department the HPLC, GC, NMR, Mass, IR-spectrophotometer, UV-spectrophotometer all are assisted by computer to identify a compound quickly with the database already provided by the suppliers

3. In inventory management
In order to obtain Quality Assurance, GMP, ISO 9000 series certificates the industries require extensive documentation which simply is impossible without computers in different department interconnected by Local Area Network (LAN). Inventory control, personnel management are supported by computer network.

4. In marketing department
- Dissemination of information about a company, its products and other relevant information became very easy by publishing through websites in the internet. e.g. Nearly all the big industries in India have their own websites. E-mail has revolutionized the concept of mailing message, keeping contact with customers.
- Marketing of a product has got a boost from the concept of Internet marketing. India has also several such portals e.g. www.rediff.com, www.indiaonline.com etc.

5. In distribution of product in market

Finally, the marketing and distribution of pharmaceutical products is supported extensively by computers. Advanced countries use bar codes to identify trade packages. Record keeping of wholesaler, retailer and two-way flow of information greatly improves the speed and efficiency of distribution.

Computer in pharmacy education

1. Distant learning through internet

Mode of teaching and learning are slowly but steadily heading toward a system which will not depend on time or distance. Universities are reorienting themselves to give degree to any students at any time and at any where in the world. This has become possible only through Internet.

2. Modern teaching aid

Multimedia packages supplied in CD-ROM has become the latest teaching aid. Through audio-visual interaction, easy searching facility through the databases is slowly supplementing the book. British Pharmacopoeia, Remington, USP Martindale Extra Pharmacopoeia - these books are also published in CD-ROM.

Drug information services

Computers are now an essential tool for pharmacists involved in drug-information retrieval and storage. On-line databases such as MEDLINE , TOXILINE, AIDSLINE any may more databases are available in "www.medscape.com", "www.nih.gov.us" websites. Other better known databases of interest to pharmacists are CAS (*Chemical Abstracts Services*) Online, International Pharmaceutical Abstracts, and SCISEARCH (*Science Citation Index and Current* and *Current Contents*).

With the advent of CD-ROM technology large amount of information is provided in this mode. Micromedex provides DRUDEX, IDENTIDEX, POISINDEX, EMERGINDEX and the current edition of *Martindal's Extra Pharmacopoeia* on a single compact disk, updated quarterly.

Computer in Hospital Pharmacy

The following list of reports or functions are commonly found in a modern hospital computer system:

1. Patient-record database
   In a modern hospital the name, address, sex, age and the medical history of a patient are documented in a database. All the computers in the hospitals are interconnected by LAN. Therefore, the details of all the patients are easily available from any department. The health status of a patient can be retrieved at any moment from any computer.
2. Medication order entry.
   The patient may be given some prescription or some medicine. This information along with date is entered in the database for future reference.
3. Lists for intravenous solutions and admixtures.
   During treatment of in-house patients different types of intravenous solutions are required to be mixed. But this types of mixtures may produce physical, chemical or therapeutic incompatibilities. Therefore, the information regarding the interaction of various intravenous preparations are stored in some database for quick reference of the pharmacists.
4. Drug-information literature retrieval.
   Information about a drug like the mode of action, indication, contraindication, adverse effect, caution for pregnant women and lactating mother etc. are kept in the database for ready reference of the pharmacist and the physicians. Each hospital has its own drug-formulary. Here the relevant information can be found regarding a drug.
5. Drug-therapy problem detection.
   When a problem arises due to some treatment, that problem along with the medication is noted down in some forms for future research. A problem can be searched whether it appeared previously and how the problem was solved.
6. Purchasing and inventory control.
   The hospital store is controlled by a special purchase and inventory control software. This software helps the store-keeper to make decision of which drug or material and the quantity to purchase.

The future

As the hardware and software technologies advances, in near future individualization of dosage regimen will be possible. The modern sensors will sense the parameters in the body and the drug delivery system will supply the dose accordingly - hence, highly individualization of dosing may be possible.

In addition, emerging drug delivery systems, including implantable pumps and similar devices are expected to depend heavily on embedded microprocessors to both sense the patient's need for the drug and control drug administration.

CHAPTER – 3
Operating Systems

Syllabus:
Introduction to different types of file manipulation and storage, maintenance functions by using DOS and WINDOWS–
File manipulations: Directories / folder / files searching, creating, copying, moving, deleting, renaming.
Maintenance: Checking, scanning and formatting a floppy disk.

DISK OPERATING SYSTEM (DOS)

DOS is the abbreviation of **D**isk **O**perating **S**ystem. It is a program that keeps link between the hardware and any application software packages.

Internal and External Commands
There are two types of DOS commands viz. *Internal* and *external*.

INTERNAL COMMANDS
A diskette containing DOS program is inserted in the Floppy Disk Drive (FDD) and then the switch is put on. When DOS is booted the internal commands are loaded into the memory (RAM). They are executed from a file named **command.com**.
They are always available for use but they are not displayed by DIR command.

Example:

DATE - Displays current date and set a new date.
TIME - Displays current time and set a new time
VER - Displays the version of DOS the computer is presently using
COPY - Copies files
DEL or ERASE - Deletes files
DIR - Lists a directory of file names on the disks
REN - Rename the files
TYPE - Displays a renamed file on the screen

EXTERNAL COMMANDS
These are program files and are not loaded while booting of DOS. These program files can be displayed, renamed, copied and deleted. They can be seen in the DOS diskette.
Example:

CHDIR or CD - Change directory
CHKDSK - Check disk-check the available disk space in a disk.
CLS - Clear screen
COMP - Compares files
DISKCOMP - Compares disks
DISKCOPY - Make verbatim copies of the source disk
DOSSHELL - Displays the management menus
FORMAT - Organize new disks for use with DOS

DATE

DATE command is an internal DOS command
Type the following:

```
A:\>DATE ↵
Current date is the 02-26-2003
Enter new date (mm-dd-yy):-
```

Type 02-26-03 , press

```
A:\>DATE
Current Date is Tue 07-26-2003
Enter new date (mm-dd-yy): 02-26-03↵
A:\>
```

The current date is now recorded. You can check it by again typing the DATE command at A:\> prompt.

TIME

This is an internal command. It displays and sets the system date.

```
A:\> TIME↵
Current time is 2:39:48. 94 P
Enter new time : 4:40:00. 0 P↵
A:\>
```

The time expression is given by *hours : minutes : seconds . hundredths of a second,.* ***a*** for am and ***p*** for pm.
Type the current hour and minute in the form of 4:40 and press Enter. The time will be recorded by DOS.

VER

It is an internal command and displays the version of DOS the computer is presently using.

```
C:\>VER
MS-DOS Version 6.22.
C:\>
```

MEM

MEM is an external command. MEM is used to display system memory information. The MEM command is entered alone or with a parameter.

MEM Displays total system memory values
MEM/P Displays programs presently loaded in memory with used and unused memory

Type the following command:

```
C:\>MEM
655360 bytes total conventional memory
655360 bytes available to MS-DOS
605344 largest executable program size

3145728 bytes total contiguous extended memory
      0 bytes available contiguous extended memory
3080192 bytes available XMS memory
        MS-DOS resident in High Memory Area.
C:\>
```

CHANGING DRIVE

```
A:\> C: ↵
C:\> D: ↵
D:\> A: ↵
A:\>
```

Type the drive letter put a colon (:) ,
Press Enter (↵) switch.
The new drive will be displayed as prompt

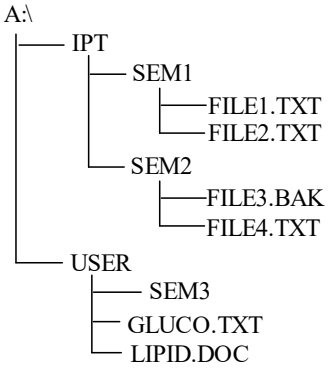
DIRECTORY COMMANDS

A directory is a list of filenames that is automatically displayed on the screen when you type DIR and press Enter.

Command	*Description*	*Example*
DIR	Lists a directory of the disk located in the active disk drive.	A:\>DIR ↵ Volume in drive A has no label Volume serial number is 0F66-14F7 Directory of A:\ COMMAND COM 37557 12.13.90 4:30 AUTOEXEC BAT 244 12.06.90 10:03 A:\>
DIR A:	Lists a directory of the specified disk A:	C:\>DIR A: ↵
DIR/P or DIR C:/P	Pauses the directory listing when the screen is full. Press any key to see the next screen load of filenames.	C:\> DIR /P ↵ C:\> DIR A:/P ↵

DIR/W or DIR A:/W	Displays a wide directory, omits file size, date and time information.	C:\>DIR/W ↵ Or C:\>DIR A:/W ↵
DIR filename . ext	Displays the specified filename; Used to verify the presence of a file on the active disc	C:\>DIR A:\abc.txt
DIR filename.* Or DIR filename.???	Displays s directory of all filenames having the same extensions.	C:\> DIR abc.*
DIR *.ext	Displays a directory of all filenames having the same extension	C:\>DIR *.txt
DIR/S	Displays all files in all subdirectories.	C:\> DIR/S

STRUCTURE OF DIRECTORIES, SUBDIRECTORIES AND FILES



Root Directory	A:\
Subdirectory	IPT USER
Subdirectory under subdirectory	SEM1, SEM2 SEM3
Files	FILE1.TXT FILE2.TXT FILE3.BAK FILE4.TXT GLUCO.TXT LIPID.DOC

Command	*Abbreviation*	*Purpose*
MKDIR	MD	Make a new directory
CHDIR	CD	Change to another directory
RMDIR	RD	Remove (or delete) a directory from the disk.

MKDIR command

MD Path\Name of the directory

MD directory_name or MKDIR directory_name	Used to create a new sub directory that is subordinate to the currently active directory.	C:\>MD A:\SEM3\CEUTICS ↵ A new directory CEUTICS will be created under A:\SEM3

CHDIR command

CD Path\Name of the directory

CD user or CHDIR user	Used to change from one directory to another but under the same parent directory	A:\>CD IPT A:\IPT\>CD SEM1 A:\IPT\SEM1\>
CD..	Double dots indicates the parent directory	A:\IPT\SEM1>CD.. ↵ A:\IPT>
CD\	Back slash changes the control to root directory	A:\IPT\SEM1>CD\ ↵ A:\>

RMDIR (or RD) command

The RMDIR or RD command is used to remove a subdirectory. First you must delete all files within a subdirectory before it can be deleted.

To remove an empty subdirectory, go to the directory level immediately above the one to be removed and then type RD.

Example → Next page

e.g. We want to delete SEM3 directory. Type the following commands.
A:\USER\SEM3>RD SEM3 ↵ *This is a wrong command.*
Bad command or file name
A:\USER\RD SEM3 ↵ *This is the correct command.*

ERASE or DELETE

The ERASE or DELETE command removes files from a diskette and reallocates the disk space that the file previously occupied.
The general form is :

DEL *file name*
ERASE *file name*

Code	Result & Comment
A:\IPT\SEM1>DEL FILE1.TXT ↵	A:\IPT\SEM1> FILE1.TXT Delete (Y/N) ?Y ↵ A:\IPT\SEM1> Deletes file1.txt
A:\IPT\SEM1>DEL *.TXT ↵	A:\IPT\SEM1>FILE1.TXT Delete(Y/N)?Y ↵ A:\IPT\SEM1>FILE2.TXT Delete(Y/N)?Y ↵ A:\IPT\SEM1 Deletes all the files in directory SEM1 having extension name .TXT
A:\IPT\SEM2>DEL FILE? .*	A:\IPT\SEM2>FILE3.BAK Delete (Y/N) ? Y ↵ A:\IPT\SEM2>FILE4.TXT Delete (Y/N) ? Y ↵ A:\IPT\SEM2> *Deletes all the files starting with name FILE and with any extension name.*

RENAME a file or Directory

It is an internal command. It is used to change the name of one or more files.
The general form of RENAME command is,
REN path\oldname newname
RENAME path\oldname newname

Code	*Result & Comments*
A:\IPT>REN FILE1.TXT ABC.RAR ↵	Renames the file FILE1.TXT to ABC.RAR
A:\>REN IPT\SEM2\FILE3.BAK ABC1.TXT ↵	Renames the file FILE3.BAK to ABC1.TXT

TYPE

It is an internal command. This command displays the contents of standard text (ASCII) files on your screen.
The general form of type command is

TYPE filename.ext ↵

The named file is displayed on the screen. If it is longer than the screen, it scrolls upward until the last time of the file displayed. Then the DOS prompt (A:\>) is redisplayed.

A:\>TYPE IPT\FILE.TXT MORE

This MORE command freezes each screen full of information. Press any key to display next screen.

CHKDSK (Check disk)
It is an external command. It checks the status of a selected disk. It displays the following information.

Information	As displayed on the screen
	A:\>CHKDSK ↵
Volume serial number	Volume serial number is OF66-14F7
The total disk space	362496 bytes total disk space
Total disk space occupied and the number of files	71680 bytes in 2 hidden files 1024 bytes in 1 directories 193536 bytes in 16 user files
Total space available on the disk.	96256 bytes available on disk
Total space in the bad sector of the disk.	1024 bytes in bad sector
	1024 bytes in each allocation unit

	354	total allocation units
	94	available allocation units on disk
Total RAM memory available.	655360	total bytes memory
Total RAM memory remaining free.	605392	bytes free
A:\>		

Use

1. The CHKDSK command can be used to determine the space available on a disk.
2. It tells the user whether or not enough memory (RAM) exists to run certain application programs.
3. A file can remain scattered over a disk in different fragments. [CHKDSK filename] this command checks the designated file for continuity in the way it is arranged on the disk.

CLS

It clears the display screen and redisplays the DOS prompt at the upper left-hand corner of the screen.

A:\> CLS ↵

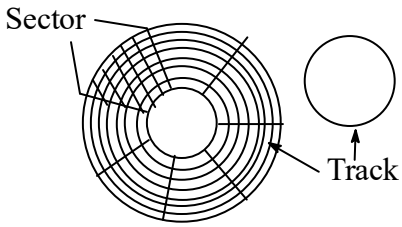
Use: This command is used to remove unwanted clutter from the display screen.

FORMAT

It is an external command

This command prepares a new disk for use by organizing the disk into magnetic tracks and sectors.

In a 3. 5" 1.44 MB diskette has 18 sectors per track and 80 tracks



FORMAT	Will format the disk in default drive	A:\>FORMAT Will format drive A:
FORMAT B:	Format the disk in drive B:	A:\>FORMAT B:
FORMAT B:/U	Specifies unconditional format, that destroys all data on the target disk to **prevent** subsequent unformating with UNFORMAT command	A:\> FORMAT B:/U
FORMAT B:/Q	Quick format takes less 10 seconds., this command removes the file allocation table and root directory; it does not scan the disk for bad areas(sectors).	A:\> FORMAT B:/Q

WINDOWS

Name of the company:	Microsoft Incorporation
Versions published till date:	Windows 3.0, Windows 3.1, Windows 3.11 Windows 95, Windows 98 Windows 2000, Windows XP Windows NT (Server), Windows 2003 Server

INTRODUCTION

What is Windows?

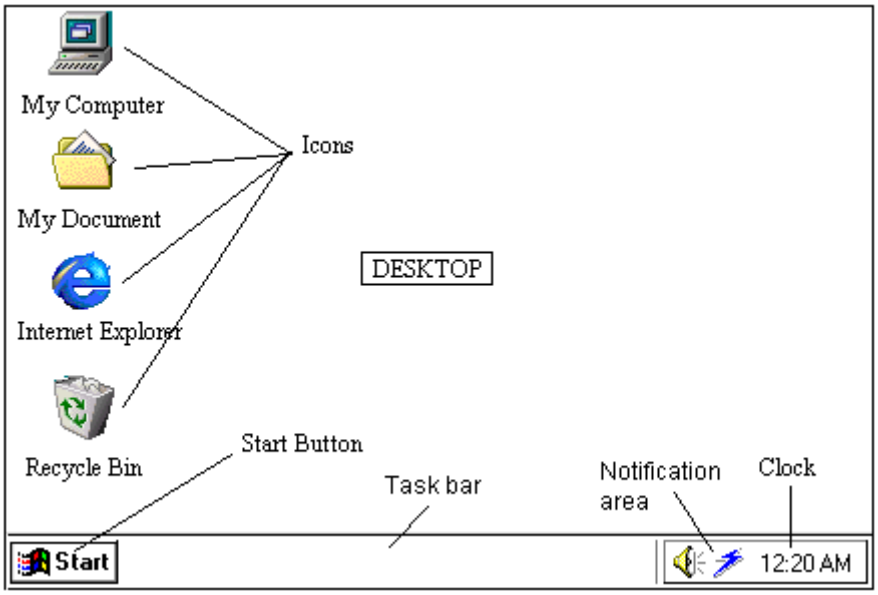
Windows is an Operating System program that controls all parts of a computer (PC). It uses graphical user interface (GUI). This kind of interface displays graphical objects called icons, which represents the items we use.
The icons are buttons those when "pushed" activate the item underlying it.
The Windows operating system gets its name from its use of rectangular boxes called windows those are used to display information and other programs. Multiple windows can be opened at the same time, making it easy to move from one task to another.

THE DESKTOP

The Windows screen is called a desktop. It displays icons those represents various tools and features.
The following tasks can be performed on the desktop.

(i) Add or remove items from the desktop.
(ii) Rearrange the items.
(iii) Open items.

PARTS OF WINDOWS DESKTOP



The minimum parts of a desktop are as follows:

(i) Icons: e.g. My Computer, Recycle Bin etc.
(ii) Task bar – The bottom of the screen is the task bar. It displays the Start button on the left hand corner and clock on the right hand corner. To the left of the clock is the notification area. It displays other indicators, like printers, volume control of an audio device or an anti-virus software running in the memory.

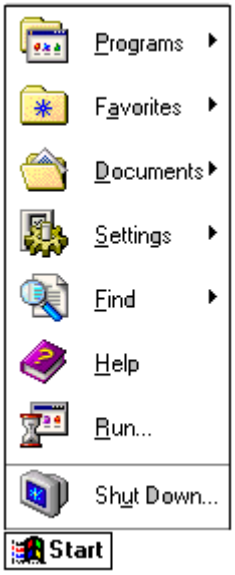
THE START MENU

When the mouse pointer is placed on the start button it is called pointing. Clicking the start button displays the start menu.

The basic start menu consists of a list of seven commands from which we can select.

The icons to the left of each command are graphic representations of the command.

Command	Used to
Programs	Start Programs
Document	Opens files and related programs
Settings	Change or view the computer system settings.
Find	Locate files
Help	Obtain direct access to the help features.
Run	Start a program using DOS command-line type functionality.
Shut Down	Shut down, restart and log-off the computer.



▸ Indicates a cascading menu will be displayed.

Pointing a menu will display the submenu from which another menu can be selected.

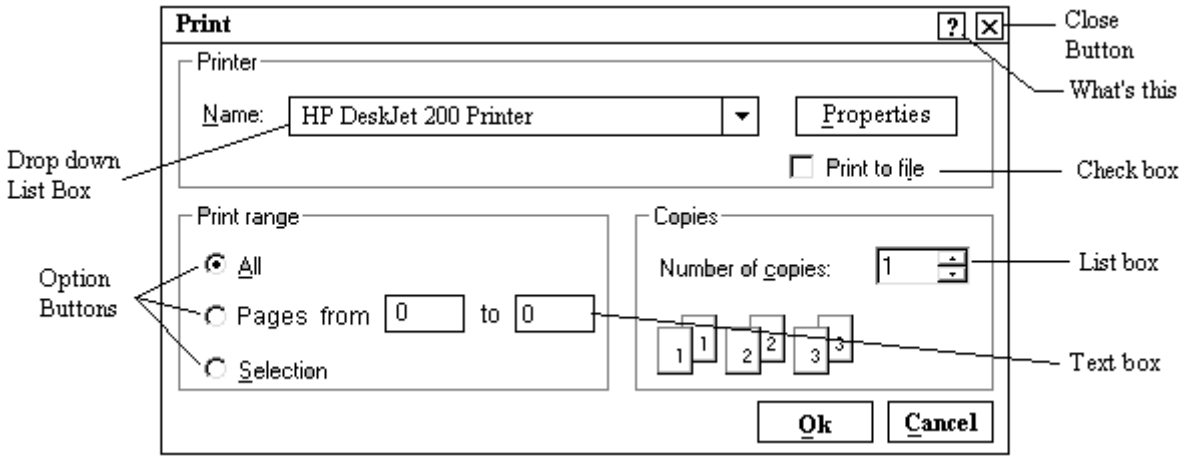
To cancel any selection click anywhere on the desktop but outside the menu box.

To choose a command, point to the command and click the left button of the mouse.

USING A DIALOG BOX

A dialog box is a common feature that is displayed whenever the program requires additional information to complete a command.

e.g. Print dialog box, Save dialog box, Open dialog box etc.



Common features of dialog box:

A title bar at the top of the box that displays a name identifying the contents of the dialog box.

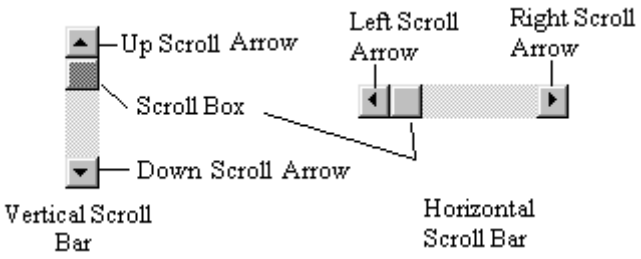
Inside the dialog box a combination of following controls may be there:

Controls	**Meaning**	**Controls**	**Meaning**
Close button	Used to close a dialog box.	What's this	Displays help on dialog box options
Text Box	An area where information can be typed into.	Option box	An option displays a black dot. Only one option can be selected from a list of option buttons.
List box	A box displaying a list of information from which we can select.	Drop down list box	A text box that displays the currently selected items and a button displays a drop down list items from which we can select or we can type the information also.

Check Box	An option preceded with a square. The selected option displays ✔. We can select more than one check box option.	Sliding control	Dragging the lever in the control increases or decreases the related setting, such as volume.
Command button	Instructs Windows to carry out the instructions on the button. E.g Ok, Cancel, Close.		

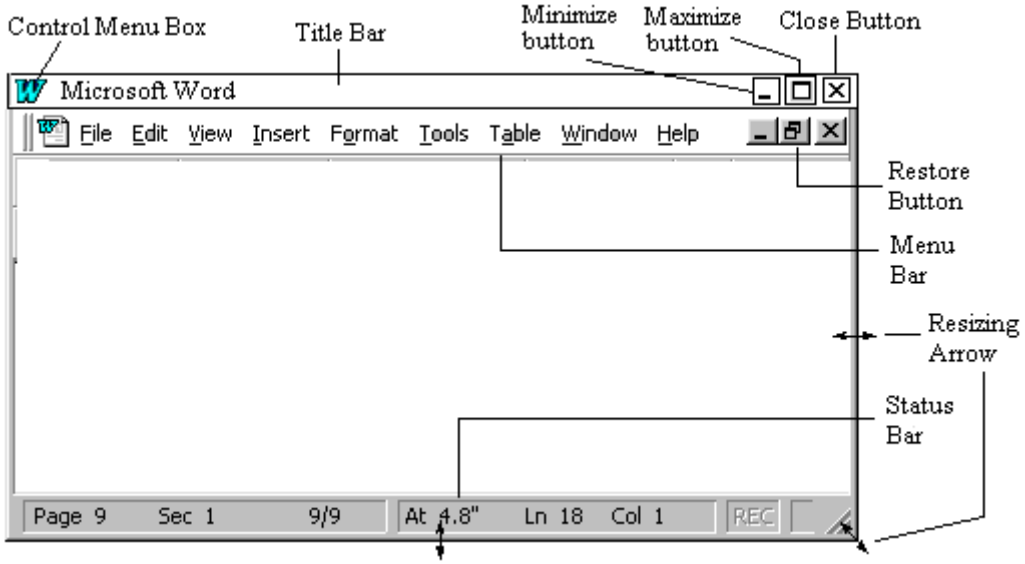
THE SCROLL BAR

Whenever there is more information than can be displayed in a window or a list box, a scroll bar is displayed.
Clicking the small arrows makes the information in the direction of the arrows.
Dragging the scroll bar moves the information in the direction of dragging.



WINDOW

A window is a rectangular section of the screen that is dedicated to a specific activity or application.
The window border outlines the window.



The basic parts of a window are described below:

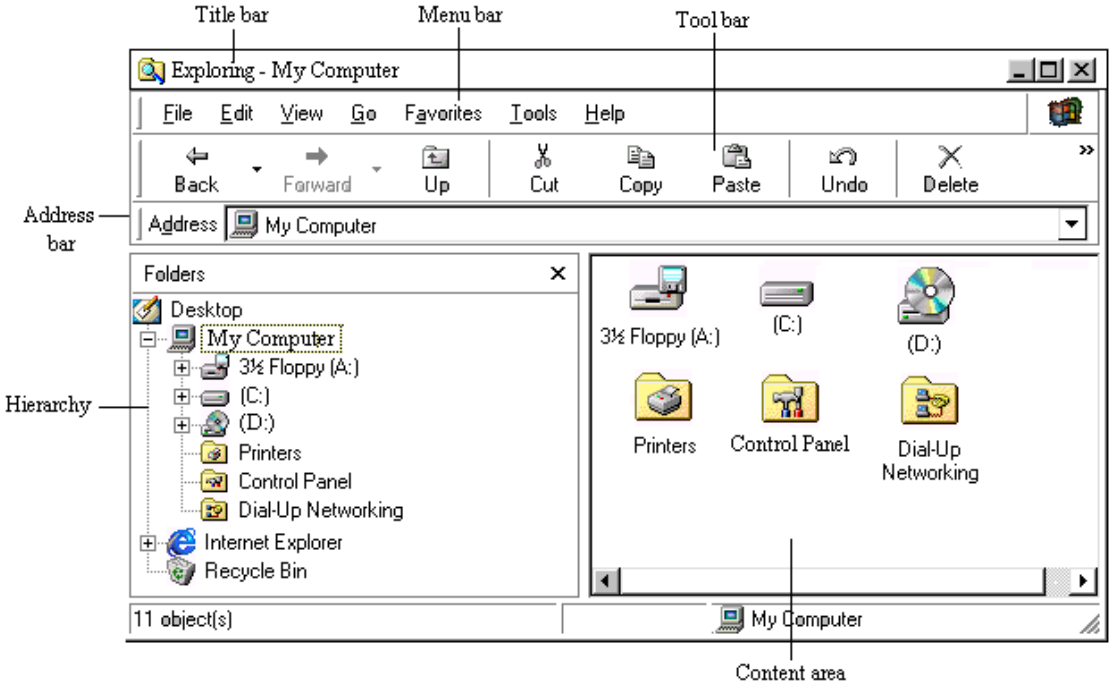
Feature	Meaning	Feature	Meaning
Title Bar	A bar located at the top of the window that displays the application name and button.	Control Menu Box	An icon located at the left hand of the title bar that when clicked displays control menu.
Minimize Button	Used to reduce a window to its smallest size.	Maximize / Restore Button	Used to enlarge a window to its maximum size. This button changes to restore when maximized. The restore button allows the window to return to its previous size.
Close Button	Used to exit the application running in the window and close the window.	Menu Bar	A bar located below the title bar containing a list of menus.
Status Bar	Present at the bottom of a window, it displays the current status of the application.		

N.B. A window always includes a menu bar and does not display command button.
A dialog box never includes a menu bar and may display command buttons.

EXPLORING THE WINDOWS

The computer can be explored either from My Computer icon or using Explorer program.
Windows Explorer program is an application with which one can organize the files and folders in a disk.
To run Windows Explorer go to Start → Program → Windows Explorer.

The exploring window is divided into two areas.
The left area is Folders that shows a graphical representation of the organization of the major parts of the computer.
The graphic representation of the organization of the disk displayed in the Folders area is called a hierarchy or tree.
The right area called Contents area, displays the contents of the selected item in the Folders area. In this case My Computer is selected.



Hierarchy of Folders

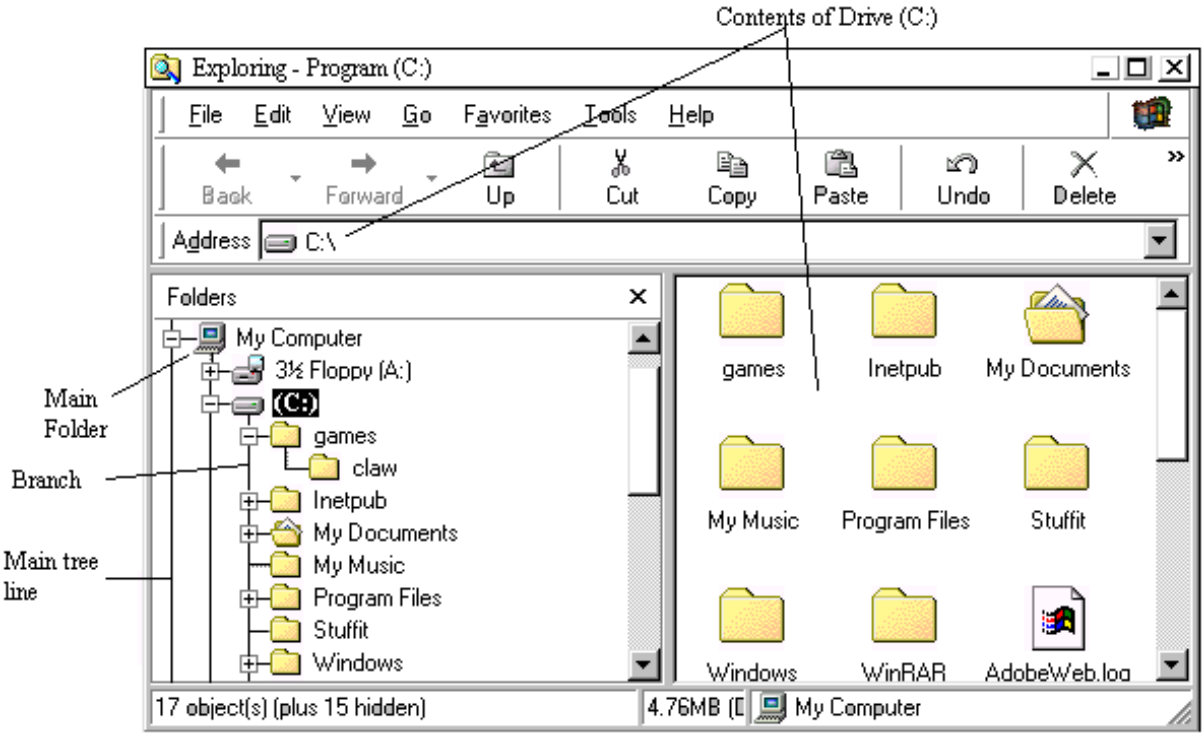
The organization of folders, subfolders, and files on the disks is displayed as a hierarchy or tree.

- The top-level folder of a disk is the main folder. On the hard disk, the main folder is represented by the My Computer icon. (see 2nd Fig.)
- The left most vertical line is the main tree line. All folders are branches from the main folder. Subfolders are branches under a folder.
- Files can be stored in main folder, a folder or a subfolder.
- Selecting different item in the Folders area displays the contents of the selected item in the Contents area. To see the contents of the hard disk click on the (C:). Your screen should be similar to the following figure.
- The folders in Folders area can also be displayed by clicking on the [+] sign displayed to the left of the drive icon to expand the hierarchy. The [+] sign will change to [–] sign, which shows that the drive icon is fully opened or expanded. Similarly the folders containing the subfolders can also be expanded by clicking on the [+] sign.
- Another way to expand the drive icon or folder is to double click on the icon.

COPYING FILES

The Windows applications include features that allow us to remove (Cut), duplicate (Copy) and insert (Paste) information from one location to another location.

- The commands for these tasks are found in the Edit Menu and on the Tool Bar.
- The information must be selected (**** very important step) before it is copied or cut. The location that contains the information is called the source. After copying or cutting the information it is stored in a temporary storage area in memory called the Clipboard.
- Finally the location where the information is to be inserted is selected. This location is called destination.
- After the destination is selected the information from the Clipboard is inserted by using Paste command.



Copy and Paste a file:

Method-I

1. The file to be copied is selected by clicking it in the Content area of the Explorer Window. The file will become blue (i.e. selected).
2. Click Edit Menu → Click Copy
3. Select the destination folder where to Paste the file. The selected folder will become blue.
4. Click Edit Menu → Click Paste

Method-II (Short cut method)

1. The file to be copied is selected by clicking it in the Content area of the Explorer Window. The file will become blue (i.e. selected).
2. Click Copy in the toolbar
3. Select the destination folder where to Paste the file. The selected folder will become blue.
4. Click Paste button on the tool bar.

Method-III (Using Right Button of Mouse)

1. Click right Mouse Button on the source **file or folder** → Click Copy.(in the popup menu)
2. Click right Mouse Button on the destination **folder** → Click Paste.(in the popup menu).

Method-IV (Using Keyboard)

1. The file to be copied is selected by clicking it in the Content area of the Explorer Window. The file will become blue (i.e. selected).
2. Press Ctrl + C (Press both keys simultaneously)
3. Select the destination folder where to Paste the file. The selected folder will become blue.
4. Press Ctrl + V (Press both keys simultaneously).

Method-V (Dragging and Dropping by Mouse)

1. Press Ctrl key and the source file (or folder) to be copied is clicked and dragged (*press the left button of the mouse and move the mouse insertion point where ever desired*) onto the **destination folder** (*destination folder must be visible in the Folders area*).
2. Left button of the mouse is released.

Cut and Paste a file:
In the **copy** method the source file will remain in the source folder but in **cut** method the source file will be removed and will be placed in the destination folder.
Method-I
Same as Method-I of Copy method only in step2 click Cut in the Edit Menu.
Method-II
Same as Method-II of Copy method only in step2 click Paste in the toolbar.
Method-III
Same as Method-III of Copy method only in step2 click Paste in the popup menu.
Method-IV
Same as Method-IV of Copy method only in step2 press Ctrl + X on the key board.
Method-V
Same as Method-V of Copy method only in step2 press Shift while dragging.

Creating a folder / subfolder
1. Select the Folder in the Folders area , in which the new folder will be created. (**** Important step)
2. Click File Menu → Click New → Click Folder. A New Folder icon will be displayed in the content area and it will remain in selected condition.
3. Type a name for the folder and press Enter.

Rules for File and Folder names in Windows
1. Names may be up to 255 characters.
2. They can contain the letters from A to Z, the numbers from 0 to 9, and any of the following special characters: underscore (_), caret (^), dollar sign ($), filde (~), exclamation point (!), number sign (#), percent sign (%), ampersand (&), hyphen (-), braces ({}) , parentheses ( ), “at” sign (@), apostrophe (‘), and the grave accent (`).
3. Spaces are allowed in the names
4. Following characters are not allowed in the names: \ / : * < > |

e.g The name of a folder may be “Photography Yearly Sales.xls”

Deleting Files or Folders
When a file or folder is no longer needed the user may decide to delete it.
Method-I
1. Select the file / folder to be deleted in the Content Area of the Explorer Window.
2. Click File Menu → Click Delete. Confirm File Delete message box will appear. Click Ok button.

Method-I
1. Select the file / folder to be deleted in the Content Area of the Explorer Window.
2. Click Delete button on the toolbar.

N.B.
When a file/folder is deleted from a Floppy disc it is permanently deleted.
When a file/folder is deleted from a Hard Disk the file/folder is placed in Recycle Bin. The file / folder will be retained in Recycle Bin unless it is permanently deleted from it. If the file / folder is present in the Recycle Bin then it can be restored back in the source folder.

Extending a selection
You can select several files or folders at a time to copy or cut. This is called extending a selection.
Method:
1. Select a file/folder.
2. Press Ctrl key and go on clicking any other files/folders. Those files or folders will be selected.

Renaming file and folder
Click on a file (or folder) to select → Click File Menu → Click Rename → Type the new name → Press Enter.

Checking a disk
Select the disk in the folder area → Click File Menu → Click Properties → Click Tools Tab → Click Check Now Button → Click Start Button.

Scanning a disk
Select the disk in the folder area → Click File Menu → Click Properties → Click Tools Tab → Click Check Now Button → Click Thorough Radio Button → Click Start Button.

Formatting a disk
Select the disk in the folder area → Click Right Mouse Button → Click Format in the popup menu → Click Start.

CHAPTER – 4
Programming with QBASIC

Syllabus:
Introduction to programming: Problem analysis, algorithm, flow chart, coding, compilation, execution, debugging and testing; programme documentation.
Constants, types of variables, array variables, arithmetic operations, precedence rule, parentheses rule, logical operations, few important library functions.
Design of programs: Initialization, input, validation, processing, print, closing a procedure.
Sequential structure of programming with example
Selective structure of programming: IF statement, block IF, nested IF statement syntax and example.
Repetitive structure of programming: FOR … NEXT loop statement syntax and example.
Function and Subroutine: Introduction with a simple example.
Graphics: LINE, CIRCLE, Changing the color of background and text.

INTRODUCTION TO PROGRAMMING

Principle behind programming with any programming language is nothing but computer-aided problem solving. It consists of the following processes:

1. Problem analysis.
2. Algorithm development.
3. Program coding.
4. Program compilation and execution
5. Program debugging and testing.
6. Program documentation.

1. Problem analysis

Whenever a problem is obtained it is first analyzed systematically to feed into a computer program. The following points are emphasized:

(i) Some information/data should be input, so input data is identified. Input information may be some data.
(ii) After program execution some information / data should come out as output, so output data is also identified. Output information may be data or graph or both.
(iii) The input data should be manipulated by some procedure, say by some calculation etc. to obtain the output information.

Example

Given three numbers x1, x2, and x3; find the largest of these numbers.

Problem analysis

Here the only information required as output in the largest of the three numbers x1, x2, and x3. The input information are the numbers x1, x2 and x3. In computer programming those three numbers are given three variable names e.g. X1, X2 and X3 to represent the memory cells containing these data items.

Initially, X1 is compared with X2. If X1 is larger than X2, it is compared with X3. If X1 is larger than X3, X1 is the largest number; otherwise X3 is the largest number. However, if X1 is not larger than X2, X2 is compared with X3. If X2 is larger than X3, X2 is the largest number; otherwise X3 is the largest number.

2. Algorithm development and flow chart

Once the problem input and output data are identified and precise statement of the problem in terms of the input and output data is available, the problem is expressed by abbreviated statements in English (or other natural languages) that specify the steps to be performed in solving the problem are called **program design language (PDL) statements** or **pseudocode**.

The finite set of PDL statements, which describe the logic for solving a specific problem, is called an **algorithm** and the process of defining PDL statements is called the **algorithm development**.
An algorithm has the following features:

1. it has a finite number of inputs
2. it terminates after a finite number of steps
3. the action taken in each step are precise and unambiguous
4. all operations specific in each step can be done exactly and in a finite amount of time and
5. it has one or more outputs derived from the inputs by applying the algorithm.

Algorithm of the previous problem

1. Input X1, X2, X3.
2. Compare X1 with X2.
3. If X1 is larger than X2, compare X1 with X3.

If X1 is larger than X3, report X1 as the largest; otherwise report X3 as the largest.

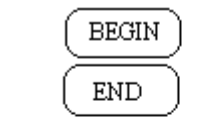
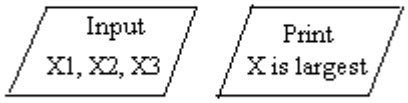
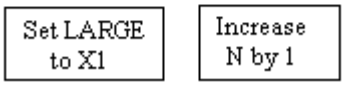
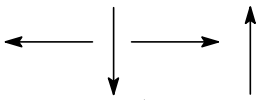
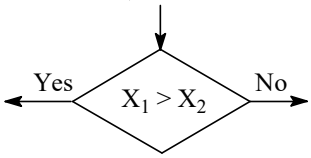
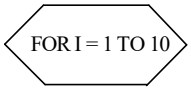
4. If X2 is not larger than X2, compare X2 with X3.

If X2 is larger than X3, report X2 as the largest; otherwise report X3 as the largest.

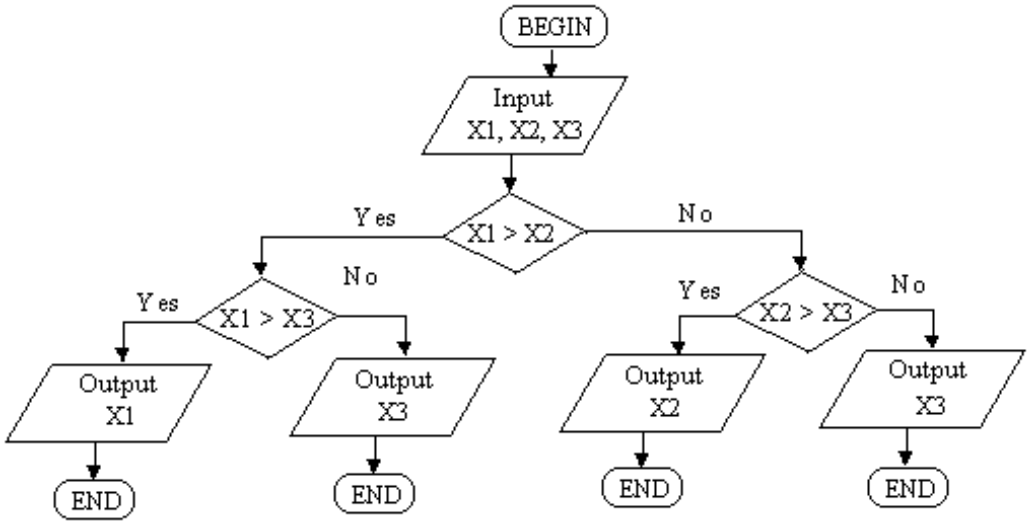
5. End.

Flowcharts: A diagram of the statements of an algorithm showing the relationships among these s**tatements is called** flowchart**.**

Symbols used in flowchart

Name of the symbol	**Symbol**	**Use**
1. Terminal Symbol		Used to indicate the beginning, or end, of a n algorithm.
2. Input / Output Symbol		A parallelogram is used to indicate the input and output, of information to or from the computer.
3. Computation or Process Symbol		A rectangle is used to indicate the assignment of values or computation.
4. Flowline Symbol		An arrow is used to indicate the order in which the steps of the algorithms are to be carried out.
5. Decision Symbol		A diamond is used to indicate a decision point. The question inside the diamond can have answers 'yes' or 'no'
6. Loop Structure Symbol		A hexagon indicates the beginning of a loop structure.

Flowchart for the above problem



3. Program coding
Program coding is the process of translating the solution of the problem, as worked out in algorithm, or flowchart, into the exact instructions that will make up the program.
The algorithm must be written using the vocabulary of the a programming language (ForTran, BASIC, C etc.) and must conform to the syntax of that programming language.

```
REM
REM     QBASIC Code for Problem 1
REM
DIM  X1 AS SINGLE, X2 AS SINGLE, X3 AS SINGLE
INPUT X1, X2, X3
IF (X1 > X2) THEN
        IF (X1 > X3) THEN
        PRINT X1
        ELSE
        PRINT X3
        ENDIF
ELSE
        IF (X2 > X3) THEN
                PRINT X2
        ELSE
                PRINT X3
        ENDIF
ENDIF
STOP
END
```

4. Program processing
Statements are entered in the computer through a key board, the program is compiled by the QBASIC compiler and then the program is executed.
5. Program debugging and testing
Very often, no matter how simple, or complex, the programs do not execute successful for the first time; the programs contain at least a few errors or bugs. The programs are therefore, **debugged**, i.e. errors or bugs are eliminated, till they execute successfully.
6. Program documentation
Before putting the program into use, the documentation for the program is provided and this is the last step in the programming process. Program documentation provides a written description of each step of the program.

BASIC stands for **B**eginner's **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode
QBASIC is also a version of BASIC language, called Quick BASIC. It was developed by Bill Gates and Paul Allen way back in 1975.

Advantages of QBASIC

1. It is a very easy to use language because many of its commands are pure English like LET, PRINT, etc. .
2. It has a simple structure for its programs: its lines may be numbered as 10, 20, 30, etc. and are executed in order.
3. It has structural similarity with many other higher level languages.
4. Simple graphics can be drawn with this program.
5. Easy helps are given readily.

PROGRAMMING IN QBASIC

How to start QBASIC program

First of all search the program QBASIC.EXE in the hard disk or floppy disk. Copy it in the floppy. Go to
Start → Click Program → Click MSDOS.
A DOS screen will be opened within the Windows. Go to A:\>QBASIC ↵
QBASIC screen will open. Go on typing the codes in the QBASIC window. Press F5 to run the program.

CONSTANTS

A **constant** is a quantity whose value does not change during program execution. It may be of **numeric, character** or **logical** type.

1. Numeric constants

Any string of digits, preceded by a single algebraic sign, is called a numeric constant. The general rules for forming numeric type constants are given below:

(a) The decimal digits 0,1,2,3,….,9 are used.
(b) The *minus* sign must precede a negative constant; a plus sigh is optional and an unsigned constant is considered positive.
(c) No commas are to be included in a string of digits.
(d) The number of digits of a constant is limited by either a maximum number of digits or a maximum magnitude.
(e) A space within a numeric constant is not allowed.

Numeric constants are subdivided into (i) *Integer* and (ii) *Real* constants.

(i) **Integer constant** (also called *fixed point co*nstant) is a constant, which does not include a decimal point.

Thus

25 0 -7 +15274

are valid integer constants, whereas the following are not for the reasons indicated beside:

Incorrect	*Reasons*	*Correct*
18.0	Contains a decimal point	18
-284.	Contains a decimal point	-284
10,235	Contains a comma.	10235
3-	Minus sign cannot be after the number.	– 3
– – 7	Two minus signs cannot precede a number	– 7

(ii) **Real constants** (also called *floating point constant*) is a constant with a **decimal** point and may have a fractional part.

Thus

18.3 – 163.0 42. + .0125 + 0.256

are valid real constants, whereas the following are not valid real constants,

Incorrect	*Reasons*	*Correct*
1,465.3	Contains a comma.	1465.3
– 56	Contains no decimal point	– 56.

Scientific notation

Very small and very large numbers are expressed conveniently by scientific notation.

Standard decimal form	Scientific notation	BASIC notation
0.0000567	5.67×10^{-5}	5.67E–5
0.0000000679	6.79×10^{-8}	6.79E–8
20000000000	2.0×10^{10}	2.0E+10
– 0.000000076	-7.6×10^{-8}	–7.6E-8

So the general form of a scientific notation is

(co-efficient) x $10^{(\text{integer})}$ = (co-efficient) E (integer)

The part appearing before E is called **mantissa** and the part following the E is called **exponent**.

2. Character constant

Any sequence of acceptable characters in the BASIC character set, enclosed within double quote (“ “), is called a **character constant** or **string constant**.

The number of characters in a character constant is the **length** of the constant.

If an apostrophe sign is to be one of the characters of the constants, it must be entered as a single quote.

Examples:

Character / String constant	Length of the string	Remarks
“ UNIX–PC ”	7	‘–‘ is a character
“ I.P.T.”	6	‘ .’ dots are also characters
“Don’t”	5	‘ single quote is a character
“More Pages”	10	‘ ‘ space is also a character

VARIABLES

A **variable** is a name used to identify the data stored in a memory location whose content may change during program execution. The rules for naming a variable are given below:

1. A variable name can contain letters A to Z (i.e. alphabet characters) and digits 0 to 9 (i.e. numerical characters) but no special characters such as +, $, *, etc.
2. The first character of a variable **must** be a letter (A to Z).
3. Maximum length of a variable name may be 40 character long.
4. No space is allowed in the variable name.

The following are valid variable names

A2X3 ITEM PAY_DAY123

The following are not valid variable names

Variable Name	Reasons
2AB	The first character is not a letter
BETA–3	(–) is a special character
X18.2	(.) is a special character
ABCDEFGHIJKLMNOPQRSTUVWZYZ12	More than 40 characters

- Certain words called reserve words such as PRINT, INPUT, CLS etc. are part of the BASIC language and hence are not valid variable names.

TYPE SPECIFICATION STATEMENTS

A variable can store a certain type of data after specifying the type of that variable. Following are the type specification statement of variables:

Type specification	Basic Keywords	Remarks	Data type suffix
Integer	INTEGER	A 16-bit signed integer variable.	%
Long-integer	LONG	A 32-bit signed integer variable.	&
Single-precision	SINGLE	A single-precision 32-bit floating-point variable.	!
Double-precision	DOUBLE	A double-precision 64-bit floating-point variable.	#
String	STRING * n%	A fixed-length string variable n% bytes long.	
String	STRING	A variable-length string variable.	$

Type specification statements may be declared **explicitly** in the following syntax

DIM variablename AS variabletype

e.g.

```
DIM NUM AS INTEGER
DIM NUM1 AS INTEGER, NUM2 AS LONG, NUM3 AS SINGLE, NUM4 AS DOUBLE
DIM FNAME AS STRING*15   –FNAME is a variable that contains maximum length of 15 characters
DIM LNAME AS STRING      – LNAME contains any length of characters.
```

A variable can also be declared implicitly by within the program by *Variablename suffix* For example

NUM% NUM2& NUM4#
NUM1% NUM3! FNAME$
LNAME$

Limits of QBASIC environment

	Maximum limit	Minimum limit
Variable name length	40 characters	1 character
String length	32,767 characters	0 characters
Integers	32,767	– 32,767
Long Integers	2,147,483,647	-2,147,483,648
Single-precision numbers		
Positive number	3.402823E+38	2.802597E-45
Negative number	-2.802597E-45	-3.402823E+38
Double-precision numbers		
Positive number	1.79769313486231D+308	4.940656458412465D-324
Negative number	-4.940656458412465D-324	-1.79769313486231D+308

Undeclared variables

If a variable is not explicitly declared by *DIM variablename AS variabletype* then QBASIC automatically assigns the variable as **Single precision.**

ARRAY VARIABLES

An array is a collection of values, or data, that are related in some way. The data may be stored in consecutive memory locations, each of which can be accessed directly. Such a *collection of values*, or data, is called an ***array*** and the *data item* its ***element***.

e.g. The test scores of 100 students are S_1, S_2, S_3, ….., S_{100}. Where S is the single group name for all the student's score and the subscript (1,2,3, .. .100) identifies the test scores of a specific student. Thus S_1 is the scores of the first student, S_{10} is the score of the 10^{th} student and so on.

In BASIC language the above data set is represented as follows: S(1), S(2), S(3),…., S(100).

The number of subscripts in a pair of parentheses may be more than one.

e.g. S(23,34,2)

The name and the range of subscript of an array may be declared in a DIM statement of the form:

DIM variable (m_1 TO n_1, m_2 TO n_2, .. , m_k TO n_k) AS type

- Each pair of integer constants, usually called **parameters.**
  m_1 is the lower bound of the array, minimum value is 0.
  n_1 is the upper bound of the array.
- AS type Declares the data type of the array or variable (INTEGER, LONG, SINGLE, DOUBLE, STRING)

e.g. DIM PUPIL (0 TO 10) AS INTEGER, MARKS (10 TO 50) AS LONG

ARITHMETIC OPERATIONS

The numeric data can be manipulated using *arithmetic operations*. A combination of variables and constants together with operation symbols, we use the phrase *expression*.

For numeric data there are six arithmetic operations:

Operation	Arithmetic Operator	BASIC operator	Examples
Addition	+	+ (plus sign)	P + Q
Subtraction	–	– (minus sign)	P – Q
Multiplication	x	* (asterisk sign)	P * Q
Division	÷	/ (slash sign)	P / Q
Exponentiation	P^Q	^ (caret sign)	P ^ Q
Negation	– P	– P	– P

Rules

1. Single mode operation

When two constants or variables of the same type are combined, using one of the four arithmetic operations (+, –, *, /) the result will be same *type* as the constants or the variables.

e.g. 8 + 2 = 10 8 – 2 = 6 8 * 2 = 16 8 / 2 = 4

e.g. 8. + 2. = 10. 8.0 – 2. = 6.0 8. * 2. = 16. 8. / 2. = 4.

If both the constants are integers and after division a fraction appears then the result will be the integer portion of the fraction

e.g. 5 / 2 = 2 (not 2.5) 1 / 2 = 0 (not 0.5)

If both the constants are real (i.e. either *single* or *double*) and after division a fraction appears then the result will be the real.

e.g. 5./2. = 2.5 1. / 2. = 0.5

2. Mixed mode operation

When one integer and another real constants or variables are involved in an operation then the integer is automatically converted to its real equivalent, and the result is a real type.

e.g. 5 / 2.0 = 2.5

3. Exponentiation

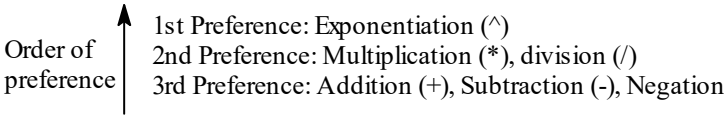
(a) If the *exponent* is an integer value the QBASIC performs the operation by repeated multiplication.
e.g. – 4 ^ 2 = (–4) * (–4) = 16

(b) If the *exponent* is a real (either single or double) then QBASIC performs the operation by logarithmic method. E.g.
– 4.0 ^ 2.0 = exp (2.0 log (–4.0)) = Undefined
Because logarithm of negative value is *undefined*.

N.B.
$(-4.0)^{2.0} = x$ (let) Taking logarithm on both sides yields: $2.0 \log (-4.0) = \log x$
Therefore, $x = 10^{2.0 \log (-4.0)}$. log (–4.0) is undefined.

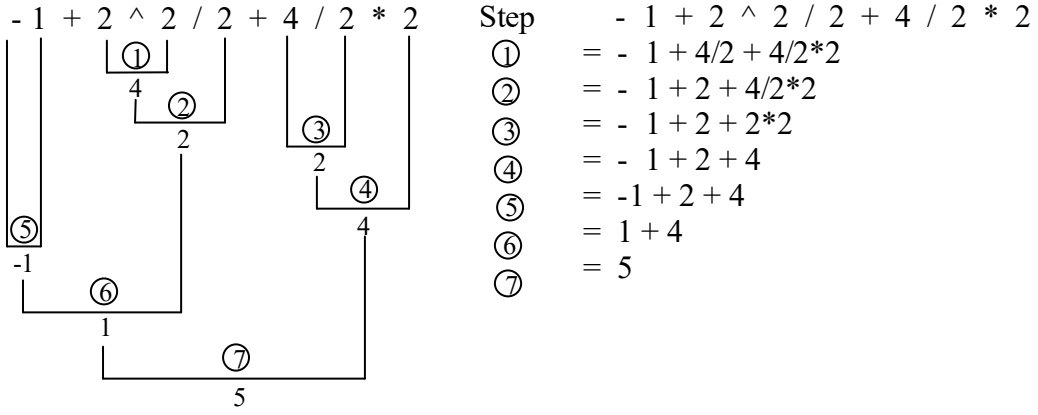
PRECEDENCE RULE

Precedence rule states that the operations will be performed in the following descending order of precedence:

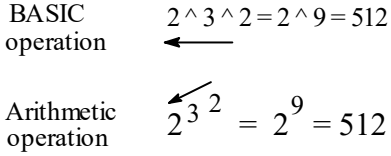


- When the order of precedence of operators is same, such as in multiplication and division, the operations will be performed in order from **left to right**.

e.g.

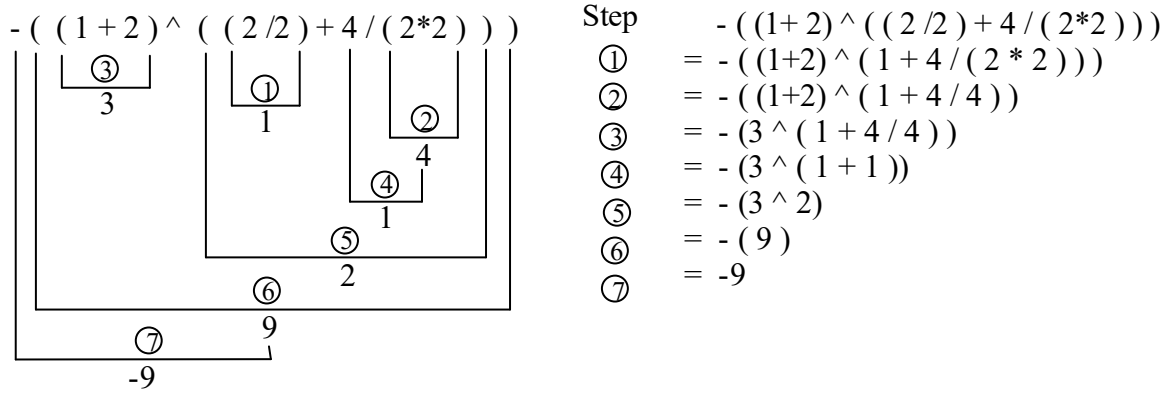


- If exponentiation operator appears multiple times in succession, the operation is performed from **right to left**.



PARENTHESES RULE

Parentheses rule states that operation will be performed in the innermost set of parentheses and then in the next outer set, etc., until all the operations inside the parentheses have been performed.



LOGICAL OPERATIONS OR BOOLEAN OPERATIONS

Boolean operators perform bit manipulations, Boolean operations, or tests on multiple relations. They return a true (nonzero) or false (zero) value to be used in making a decision.

Boolean operators	**Description**	**Syntax**	**Examples**

NOT	Used to perform logical negation on an expression.	*result* = **Not** *expression*	`Dim A, B, C, D, MyCheck`
			`A = 10: B = 8: C=5`
			`MyCheck = Not (A > B) ' Returns False.`

AND	Used to perform a logical conjunction on two expressions.	*result* = *expression1* **And** *expression2*	`Dim A, B, C, D, MyCheck`
			`A = 10: B = 8: C=5`
			MyCheck = A > B **And** B > C ' Returns True

OR	Used to perform a logical disjunction on two expressions.	*result* = *expression1* **Or** *expression2*	`Dim A, B, C, D, MyCheck`
			`A = 10: B = 8 : C=5`
			MyCheck = A > B **Or** B > C ' Returns True.

Other logical operators are XOR (exclusive or), EQV (equivalence) and IMP (impliactions).

LIBRARY FUNCTIONS

There are a number of computational processes that require multiple steps to obtain the desired result and are used again and again by different programmers. Examples include computing the square root of a given number, determining the absolute value of an expression, finding the largest value from a set of numbers and so on.

Instead of programming them again and again QBASIC provides a number of **built-in** functions. This buit-in functions are called **intrinsic** functions.

Function	**Description**	**Syntax**	**Example**
ABS	Returns the absolute value of a number.	ABS(numeric-expression)	PRINT ABS(45.5 – 100.0) 'Output is: 54.5
CINT	rounds a numeric expression to an integer	CINT(numeric-expression)	PRINT CINT(12.49), CINT(12.51) 'Output is: 12 13
CLNG	rounds a numeric expression to a long (4-byte) integer	CLNG(numeric-expression)	PRINT CLNG(338457.8) 'Output is: 338458
CSNG	converts a numeric expression to a single-precision value	CSNG(numeric-expression)	CSNG(975.3421515) 'Output is: 975.3422
CDBL	converts a numeric expression to a double-precision value	CDBL(numeric-expression)	CDBL(1 / 3) 'Output is: .3333333333333333
FIX	truncates a floating-point expression to its integer portion	FIX(numeric-expression)	PRINT FIX(12.49), FIX(12.54) 'Output is: 12 12
INT	returns the largest integer less than or equal to a numeric expression	INT(numeric-expression)	PRINT INT(12.54), INT(-99.4) 'Output is: 12 -100
ATN	returns the arctangent of a specified numeric expression	ATN(numeric-expression)	CONST PI=3.141592654 PRINT ATN(TAN(PI/4.0)), PI/4.0 'Output is: .7853981635 .7853981635 PRINT (COS(180 * (PI / 180))) 'Output is: -1 N.B. To convert from degrees to radians, multiply degrees by (PI / 180).
SIN	return the sine of a specified angle in radian	SIN(angle)	
COS	return the cosine of a specified angle in radian	COS(angle)	
TAN	return the tangent of a specified angle in radian	TAN(angle)	

Function	Description	Syntax	Example
EXP	returns e raised to a specified power, where e is the base of natural	*EXP(numeric-expression)* N.B. For EXP, the numeric expression is a number less than or equal to 88.02969.	PRINT EXP(0), EXP(1) 'Output is: 1 2.718282
LOG	returns the natural logarithm of a numeric expression	*LOG(numeric-expression)* N.B. For LOG, any positive numeric expression.	PRINT LOG(1), LOG(EXP(1)) 'Output is: 0 1
MOD	Divides one number by another and returns the remainder.	*numeric-expression1 MOD numeric-expression2* **numeric-expression1, numeric-expression2** – Any numeric expressions. Real numbers are rounded to integers.	PRINT 19 MOD 6.7 'QBasic rounds 6.7 to 7, then divides. 'Output is: 5
SQR	Returns the square root of a numeric expression.	*SQR(numeric-expression)* **numeric-**expression – A value greater than or equal to zero.	PRINT SQR(25), SQR(2) 'Output is: 5 1.414214

STRING MANIPULATIONS

Function	Description	Syntax	Example
INSTR	Returns the position of the first occurrence of a string in another string	*INSTR([start%,]stringexpression1$,stringexpression2$)* N.B. **start%** – Sets the character position where the search begins. If start% is omitted, INSTR starts at position 1. **stringexpression1$ –** The string to search **stringexpression2$** –The string to look for.	a$ = "Microsoft QBasic" PRINT INSTR(1, a$, "QBasic") 'Output is 11
LEFT$ RIGHT$	Return a specified number of leftmost or rightmost characters in a string.	LEFT$(stringexpression$,n%) RIGHT$(stringexpression$,n%) N.B. **stringexpression$** – Any string expression. **n%** – The number of characters to return, beginning with the leftmost or rightmost string character.	a$ = "Microsoft QBasic" PRINT LEFT$(a$, 5) 'Output is: Micro PRINT RIGHT$(a$, 5) 'Output is: Basic
MID$	The MID$ function returns part of a string (a substring). The MID$ statement replaces part of a string variable with another string	*MID$ (stringexpression$, start% [,length%])* *MID$ (stringvariable$, start% [,length%])=stringexpression$* N.B. **stringexpression$** – The string from which the MID$ function returns substring, or the replacement string used by the MID$ statement. It can be any string expression. **start%** – The position of the first character in the substring being returned or replaced. **length%** – The number of characters in the substring. If the length is omitted, MID$ returns or replaces all characters to the right of the start position. **stringvariable$** – The string variable being modified by the MID$ statement.	a$ = "Where is Paris?" PRINT MID$(a$, 10, 5) 'Output is: Paris Text$ = "Paris, France" PRINT Text$ 'Output is: Paris, France MID$(Text$, 8) = "Texas " PRINT Text$ 'Output is: Paris, Texas
LEN	Returns the number of characters in a string or the number of bytes required to store a variable.	*LEN(stringexpression$)* *LEN(variable)* N.B. **stringexpression$** – Any string expression. **Variable** – Any nonstring variable.	a$ = "Microsoft QBasic" PRINT LEN(a$) 'Ouput is 16

Exercise
Write the QBASIC equivalent of the following expressions:

Arithmetic Expression	QBASIC Expression	Arithmetic Expression	QBASIC Expression
a + b / c – d	A + B/C – D	$x = \sqrt{y}$	X = SQR (Y)
$\frac{a + b}{c + d}$	(A + B) / (C + D)	x = | y – a |	X = ABS (Y)
$a^2 - b^2$.	A ^ 2 – B ^ 2	$x = e^{y + a}$	X = EXP(Y + A)
$\frac{ab}{c^2 - d}$	A*B / (C^2 – D)	$\log_{10} (2a - 3b)^2$.	LOG((2*A – 3*B)^2)/LOG (10)

Do the following exercises
Convert the following arithmetic expressions to QBASIC expressions

(a) $a^z - 2.0$ (b) $\frac{ab}{d} - c^2$ (c) $(a^n)^m + a^n a^m$. (d) $a + \frac{1}{1 + \frac{1}{1 + a}}$

(e) $\sqrt{3a^2 + 6b^2}$ (f) $\sqrt{x^2 / (y + z)}$ (g) $\log_e (x + 3y)^3$. (h) $\cos(2x - y) + |x2 + y^2| + e^{xy}$.

(i) $\sqrt{|\sin (a - |b|)|}$ (j) $\left| \sqrt{a - b^2} - \frac{c^2}{\sin (x + y)} \right|$

Convert the following QBASIC expressions to arithmetic expressions
(a) SQR (3*A^2 + * B ^ 2)
(b) SQR (X ^ 2 / (Y + Z))
(c) COS (2*X–Y) + ABS (X^2 + Y^2) + EXP (X * Y)
(d) LOG ((2 * A – 3 * B) ^ 2
(e) A + B / ABS (M – N)
(f) EXP (ABS (Z)) – Y ^ 3 / ABS (X)
(g) COS (LOG (3 * X + Y))
(h) SQR (ABS (SIN (A – ABS (B)))))
(i) ABS (SQR (A – B ^ 2) – C ^ 2 / SIN (X + Y))

Exercise
Calculate the values of each of the following expressions as per precedence rules:

(a) 9 – 6 + 3	(b) 3 ^ 2 + 4 / 5	(c) 2.0 / 4	(d) 3 + 2 ^ 3	(e) (3 + 2) ^ 3
(f) – 4.0 ^ 4	(g) – (4.0 ^ 4)	(h) 12.0 / 3.0 * 2.0	(i) 12 / (3.0 * 2.0)	(j) (2 ^ 3) ^ 2
(k) 2 ^ 3 ^ 2	(l) ((3 + 2) ^ 3) / 5	(m) (3 + 2 ^ 3) / 5	(n) (3. + 2 ^ 3) / 5	

The values are calculated as follows:
(a) 9 – 6 + 3 = 3 + 3 = 6
(b) 3 ^ 2 + 4 / 5 = 9 + 4/5 = 9 + 0 = 9
(c) 2.0 / 4 = .5
(d) 3 + 2 ^ 3 = 3 + 8 = 11
(e) (3 + 2) ^ 3 = 5 ^ 3 = 125
(f) – 4.0 ^ 4 = – (4.0)^4 = – (4.0) (4.0) (4.0) (4.0) = – (256.0) = -256.0
(g) – (4.0 ^ 4) = – (4.0 * 4.0 * 4.0 *4.0) = – (256.0) = – 256.0
Rest of the problems do yourself.

DESIGN OF PROGRAMS

Although computer program differ greatly in purpose and processing, they can all be organized into the following functional modules:

1. **Initialization** establishes initial values for some variables, prints headings, messages etc.
2. **Input** performs input of data required by the program.
3. **Validation** performs validation of input data to detect errors, omissions etc.
4. **Processing** performs computation, or data manipulation.
5. **Output** performs output of data to be provided by the program like printing the output.
6. **Closing procedure** performs procedure to end the execution of the program.

STRUCTURE OF PROGRAM

All computer program can be coded by using only three logical structures (patterns) given below, or a combination of these structures:

1. **Sequential structure**: It consists of one action followed by another. In other words performs operation A and then operation B and so on.
2. **Selective structure**: It consists for a test for a condition followed by two or more alternative paths for the program to follow The program selects one of the paths depending on the test of the condition.
3. **Repetitive or Iterative structure**: Here an operation, or a set of operations, is repeated as long as some condition satisfied.

SEQUENTIAL STRUCTURE

1. **Assignment statement**

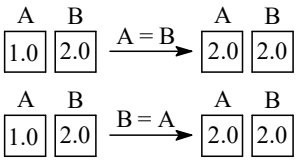
The assignment statement is used to assign values to variables and has the form of *variable = expression.*

Where expression may be a

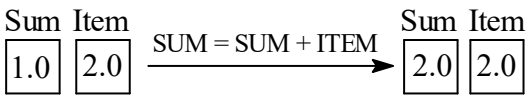
(a) constant (b) a variable to which a value has been assigned

(c) a formula which the computer can evaluate.

e.g.



e.g.



Examples of Rules of assignments

1. In the left-hand side of the '=' symbol a variable should be there.
2. No value, constant, or expression can be on the left-hand side of the '=' symbol.
3. In the right-hand side any value, constant or valid expression can be placed.
4. The type of variable and the expression or constants in both the sides should be same.

e.g X = 2 Y = (N + 4) / 3 K = 1. / 3. + 1./3 IPT = "Institute"

Hence, the following are not valid QBASIC assignment statements for the reasons indicated:

15 = N	Variable N is on the right instead on the left of the = sign.
X + 4.3 = 3.14	Numeric expression should not appear to the left-hand side of = sign.
STRING = 4 & 7	4 & 7 is not a legal expression.
A = B = 1	B = 1 is not a valid expression
DIM A AS INTEGER A = "abcd"	When A is declared as integer in the first line a string constant "abcd" cannot be assigned to A.

2. **Input and output statements**

The computer can manipulate the data only it is available to its memory. Data can be input in the memory either within the program or by the user during program execution.

Data can be input in the memory by the following statements and the result is shown beside:

Sl. No.	Syntax	Example	Displayed result during program execution
1	INPUT variablename	INPUT ipt	_
2	INPUT "prompt"; variablename	INPUT "First Data:"; X	First Data: ?_
3	INPUT "prompt", variablename	INPUT "First	First Data: _

		Data:", X	
Sl. No.	Syntax	Example	Displayed result during program execution
4	INPUT variable1, variable2	INPUT X, Y	?12 , 20 Press enter
5	INPUT "prompt", variable1, variable2	INPUT "X=",X, Y	X= 12, 20
6	INPUT "prompt"; variable1, variable2	INPUT "X=";X, Y	X= ? 12, 20

Data can be output from the memory by the following statements and the result is shown beside:
PRINT writes data to the screen or to a file.
LPRINT prints data on the printer LPT1.
The general syntax is
PRINT [expressionlist] [{; | ,}]
LPRINT [expressionlist] [{; | ,}]
_ expressionlist A list of one or more numeric or string expressions to print.
_ {; | ,} Determines where the next output begins:; means print immediately after the last value., means print at the start of the next print zone. Print zones are 14 characters wide.

Sl. No.	Syntax	Example	Displayed result during program execution
1	PRINT variable1, variable2	PRINT A, X, Z	25 10 12
2	PRINT variable1; variable2	PRINT A; X; Z	25 10 12
3	PRINT string constant	PRINT "Institute"	Institute
4	PRINT string1 + string2	PRINT "ab" + "cd"	Abcd
5	PRINT string, variable	PRINT "X=",X	X= 20
6	PRINT string; variable	PRINT "X=";X	X= 20

STOP
The STOP statement is used to halt the program execution. But the program execution will not terminate here and the control will not be given to the user. Press F5 to continue.

END
This statement terminates a program and give the control to the user.
END must be the last statement of a program.

Example: Write the algorithm, flow chart and the program code.

1. Write a program that accepts input temperature in degree Farenheit, converts it to degree Centigrade, and outputs the temperature in degree Centigrade.
2. Write a program that reads values for the three sides of a triangle, calculates its perimeter and its area, and outputs these values.
3. Write a program that inputs the radius of a circle, computes and outputs (a) the area of the circle, (b) the area of the largest square contained within the circle, and (c) the ratio of (a) to that of (b). Use the constant, PI, the value 3.1416.
4. Write a program to find the surface area (SUR), volume (VOL) of a box with dimensions a,b,c where SUR = 2(ab +bc + ca) and VOL =abc.
5. Assuming $a_1b_2 - a_2b_1 \neq 0$, the solution of the linear equations $a_1x + b_1y = c_1$ and $a_2x + b_2y = c_2$ is given by

   $$x = \frac{b_2c_1 - b_1c_2}{a_1b_2 - a_2b_1} \qquad y = \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1}$$

   Write a program that reads values of a1, b1, c1, a2, b2, c2 and calculates and prints the solution pair x and y.
6. Suppose that a car starts from rest and has constant acceleration for t seconds. The final velocity v and the distance traveled d by the car is given by the formulae:
   v = ft and d = ½ ft^2.
   Write a program that reads f and t, and prints t, d, and v.
7. Write a program to convert a given measurement in feet to an equivalent one in (i) yards, (ii) inches, (iii) centimeters, (iv) meters. (1yard = 3 ft, 1 foot = 12 inches, 1 inch = 2.54 centimeter, 1 meter = 100 centimeter)

2. SELECTIVE STRUCTURE

The selective structure consists of a test for a condition followed by alternative paths which the program can follow. Selection among alternative paths is programmed with the IF statements.

Relational expression

If A and B represents two *numeric or string constants* they can be combined by the following six *relational operators*:

Relational operator	Application in QBASIC	Meaning
<	A < B	A is less than B
< =	A < = B	A is less than or equal to B
=	A = B	A is equal to B
< >	A < > B	A is not equal to B
>	A > B	A is greater than B
> =	A > = B	A is greater than or equal to B

- The relational operator can be used to compare numerical values. If a relational operator is present in between two numerical expression then the numerical expression will be executed first.
  e.g. Assuming the two *single* values X = 20.0, Y = 4.0 and the integer variable M = 5 then
  X + 5 > M * Y → 20.0 + 5 > 5*4.0 → 25.0 > 20.0 → Answer is "True"
- The relational operators can be used to compare two string expressions.
  e.g. Let us assume three string variables X = "a", Y = "b", Z = "A"
  X > Y → True because ASCII code of "a" is 97 and "b" is 98
  X > Z → True because ASCII code of "a" is 97and "A" is 65

Logical expression

Logical operators used in QBASIC

Symbol	Expression	Meaning	Logical value
NOT	NOT A	Negation	**True** if and only if A is **False**.
AND	A AND B	Conjunction	**True** if and only if, A and B are both **True**.
OR	A OR B	Disjunction	**True** if and only if, at least one is **True**.

The following two truth tables summarizes the logical values for these expressions:

Truth Table - I

A	NOT A
T	F
F	T

Truth Table -II

A	B	A AND B	A OR B
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

Precedence of relational and logical operations

Precedence 1: Parentheses rules
Precedence 2: Numeric expressions
Precedence 3: Relational operations
Precedence 4: Logical operations

If more than one logical operations are in one expression then the execution will start from the left hand side.

e.g. Say A =3, B =4 and C =12 and the expression will be executed as follows:

A > B AND (A < C OR B < > C) OR NOT (A = B)
⇒ A > B AND (True OR B < > C) OR NOT (A = B)
⇒ A > B AND (True OR True) OR NOT (A = B)
⇒ A > B AND True OR NOT (A = B)
⇒ A > B AND True OR NOT (False)
⇒ A > B AND True OR True
⇒ False AND True OR True
⇒ False OR True
⇒ True

GOTO statement

The GOTO statement is used to jump from one line to a another line.

Syntax: GOTO *line*

_ *line* The label or number of the line to execute next.

Precaution: Indiscriminate use of GOTO statement may make a program difficult to debug and understand therefore, GOTO should be used with caution.

e.g.

```
10  LARGER = V2
    SMALLER = V3
    GOTO 10
```

IF Statements

Executes a statement or statement block depending on specified conditions.

Syntax I (Logical IF Statement)

```
IF condition THEN statements [ELSE statements]
```

Syntax II (Block IF statement)

```
IF condition1 THEN
   [statementblock-1]
[ELSEIF condition2 THEN
   [statementblock-2]]...
[ELSE
   [statementblock-n]]
END IF
```

condition1	Any expression that can be evaluated as
condition2	true (nonzero) or false (zero).
statementblock-1	One or more statements or one or more lines.
statementblock-2	
statementblock-n	
statements	One or more statements, separated by colons.

Example of Logical IF :

```
IF  (num  <  100 ) PRINT  num
IF (num  < 100 ) PRINT  num  ELSE  PRINT "No number"
```

Example of block IF:

```
DIM  v1 AS INTEGER, v2 AS INTEGER, larger AS INTEGER
INPUT  v1,v2
IF (v1 > v2) THEN
        larger = v1
ELSE
        larger = v2
END IF
PRINT larger
END
```

Example of nested block IF:

```
DIM v1 AS INTEGER, v2 AS INTEGER
INPUT v1, v2
IF  (v1  >  v2) THEN
        PRINT "Semester 1"
ELSE
        IF (v2 > v1) THEN
                PRINT "Semester 2"
        ELSE
                PRINT "Semester 3"
        END IF
END IF
END
```

Nest

IF Block

Exercises

Problem1. Write a program to find the largest of the three numbers.

Solution:

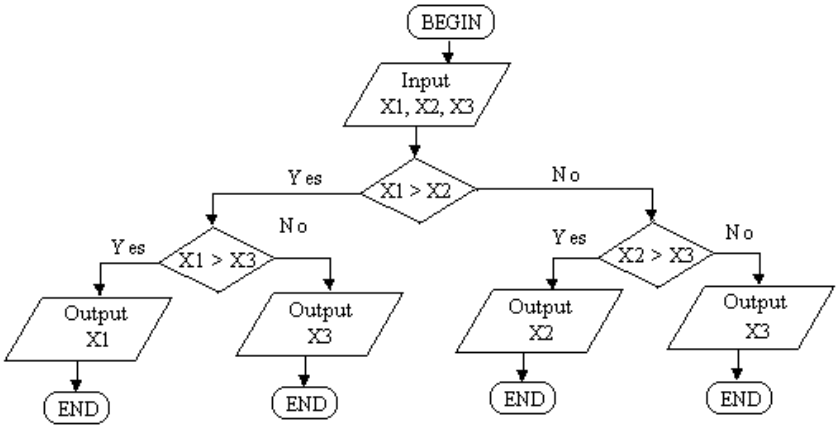


Fig. Flow chart

Program coding:

```
REM Program to find the largest of 3 numbers
DIM x1 AS SINGLE, x2 AS SINGLE, x3 AS SINGLE, largest AS SINGLE
INPUT x1,  x2,  x3
IF (x1 > x2) THEN
        IF (x1 > x3) THEN
                largest = x1
        ELSE
                largest = x3
        END IF
ELSE
        IF (x2 > x3) THEN
                largest = x2
        ELSE
                largest = x3
        END IF
END IF
PRINT largest
END
```

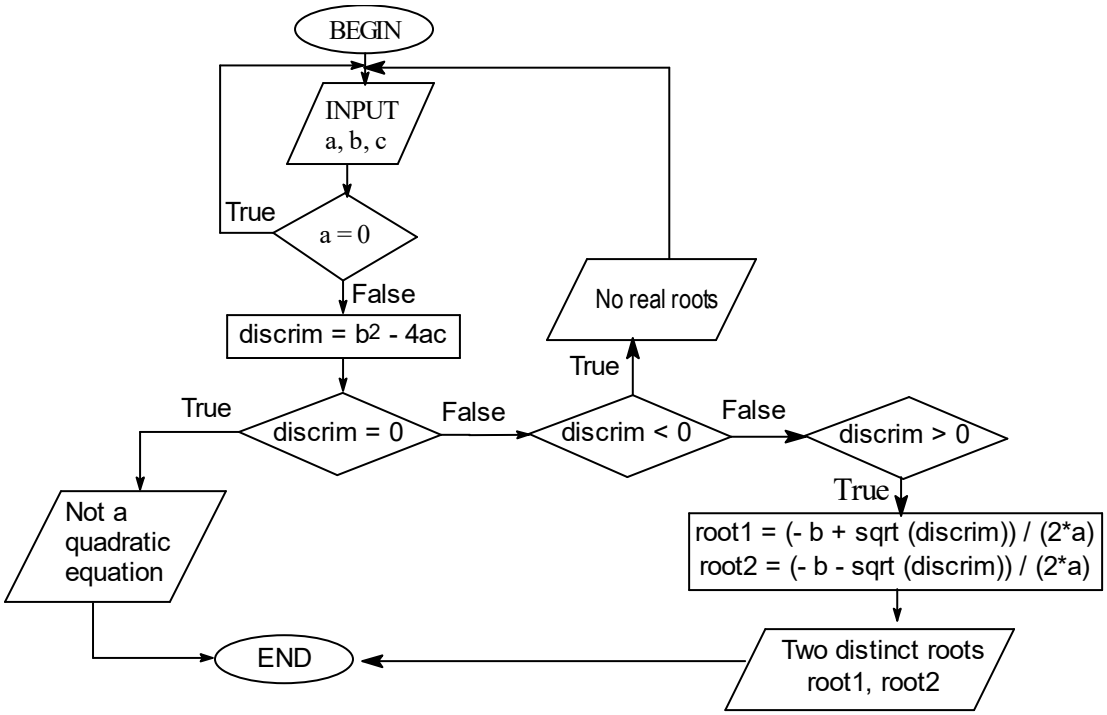
Problem 2: Write a program that reads the coefficients ***a****,* ***b*** *and* ***c*** *(****a*** *≠ 0) of the quadratic equation* $ax^2 + bx + c = 0$ *and computes its real roots. The roots of the quadratic equation are given by*

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

N.B. When the discriminant $(b^2 - 4ac) > 0 \Rightarrow$ two real roots are present
When the discriminant $(b^2 - 4ac) = 0 \Rightarrow$ two identical real roots are present and are equal to $(-b/2a)$
When the discriminant $(b^2 - 4ac) < 0 \Rightarrow$ no real roots are present

Algorithm

1. Begin : Declare a, b, c, discrim, root, root1, root2 as SINGLE
2. Read a, b and c.
3. If a = 0 then give the message "Not a quadratic equation" and GOTO statement 2 to read data again.
4. Calculate discriminant from the formula discrim = b^2 – 4*a*c
5. If discrim < 0 then give the message " No real roots" and END the program here.
6. If discrim = 0 then give the message "Two identical roots" and give the value by using root = – b /(2*a) and end the program here.
7. If discrim > 0 then give the message "Two distinct roots"
   root1 = (– b + SQR(discrim)) / (2*a) root2 = (– b – SQR(discrim)) / (2*a)
   end the program here.



Program coding

```
        DIM  a AS SINGLE, b AS SINGLE, c AS SINGLE
        DIM  discrim AS SINGLE, root AS SINGLE, root1 AS SINGLE, root2 AS SINGLE

10      INPUT a, b, c

        REM Checking for A <> 0
        IF  ( a = 0 ) THEN
                PRINT "Not a quadratic equation and  A cannot be zero. Input data again."
                GOTO 10          ' Line 10 is the input line.
        END IF
REM Calculate the discriminant
        discrim = b^2 – 4*a*c
REM  Discriminant is less than zero
        IF discrim < 0 THEN
                PRINT "No real root. The program ends here"
                GOTO 99          ' Line 99 is the last line of this program.
        END IF
REM Discriminant is equal to zero
        IF discrim = 0 THEN
                root = – b / (2 * a)
                PRINT "Two identical roots", root
                GOTO 99          ' Line 99 is the last line of this program.
        END IF
REM Discriminant is greater than zero
IF discrim > 0 THEN
        root1= (– b + SQR (discrim) ) / (2 * a)
        root2= (– b – SQR (discrim) ) / (2 * a)
        PRINT "Two distinct roots :", "Root 1="; root1, "Root2="; root2
END IF
99      END
```

EXERCISES

1. If I = 2, J = 3, K = 6 then what values do the following expressions have?
(a) I > J AND I * J <= K
(b) I * J >= K AND I > J
(c) NOT I > J AND I * J >= K
(d) I > J OR I * J >= K
(e) I > J AND (I <= K OR I * J <= K)

2. If I = 1 and J = –1 then what values the following logical expression have?
(a) ((I > 0) AND (J < 0) OR (NOT (I > 0) AND NOT (J < 0)
(b) ((I > 0) AND NOT (J < 0) OR (NOT (I > 0) AND (J < 0)

3. Write a logical IF statement that prints YES if FOOD is between –1 and 1 (i.e. $-1 \leq FOOD \leq 1$)

4. What will be the final value of NERD at the end of the each program fragment?

(a)
```
NERD = 5 : JOCK = 10
IF (3*NERD < JOCK ) THEN NERD = NERD + 2
NERD = NERD + 3
```

(b)
```
NERD = 5 : JOCK = 10
IF (2* JOCK <= 3* NERD) THEN GOTO 10
NERD = NERD +1
GOTO 20
10  NERD = JOCK
20  NERF = NERD + JOCK
```

8. Write a program to convert to Fahrenheit if the temperature input is in Centigrade and convert to Centigrade if the input temperature is in Fahrenheit scale.

9. Write a program to calculate the parts of alcohols (one of higher and other of lower strength) required to prepare a alcohol of desired strength. (Hints. Use aligation method.)

10. Write a program that calculates the Reynold's number from diameter (DIA), velocity (VEL), density of liquid (DENSITY) and viscosity of the liquid (VISCOSITY) from the formula

REYNUM = (DIA*VEL*DENSITY)/VISCOSITY

If REYNUM ≤ 2000 give output
"The liquid is flowing in a stream line flow. Reynold's number is:" REYNUM
If REYNUM ≥ 4000 give output
"The liquid is flowing in a turbulent flow. Reynold's number is:" REYNUM
If 2000 ≤ REYNUM ≤ 4000 give output
"The liquid is flowing in transition state. Reynold's number is:" REYNUM

3. REPETITIVE STRUCTURE

In a *repetitive* structure a set of statements are executed many times, but that set of statements will appear only once in the program.

Any repetitive structure contains an **entry point**, a **repetition** or **loop body**, and an **exit point**, as illustrate in the figure:

The number of repetitions in a repetitive structure can be either **condition-controlled** or **counter-controlled**.

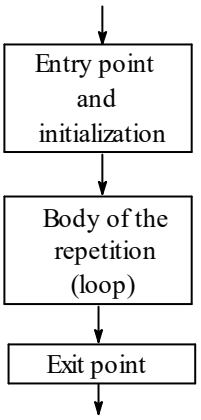


Fig. Repetitive structure

IF Loop

In the IF loop the number of repetition can be condition controlled.

Program fragment

```
100     INPUT  A, B
        SUM = A + B
        PRINT A, B, SUM
        GOTO 100
        STOP
        END
```

In the above program fragment there is no termination point. So this program will iterate (repeat) for an infinite time. To stop this loop a condition may be provided.

For example if the A = 9999 then the program will stop. This A = 9999 is a termination condition.

The same program fragment can be written in the following manner:

```
100     INPUT  A, B
        IF ( A = 9999) THEN GOTO 99
        SUM = A + B
        PRINT A, B, SUM
        GOTO 100
99      STOP
        END
```

Counter controlled by IF loop

To control the number of times a loop is executed, a **counter** may be used whose value is changed by 1 every time then loop is executed. When the counter attains a predetermined value, the loop execution is terminated.

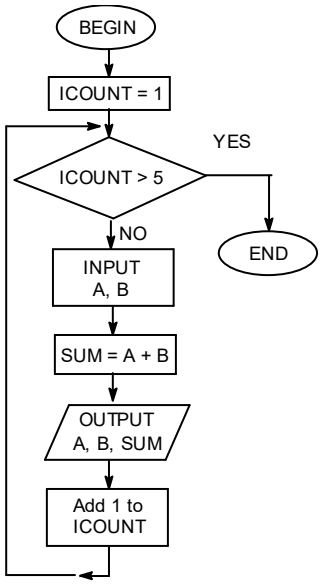
Example:

```
        DIM ICOUNT AS INTEGER
        DIM A AS INTEGER, B AS INTEGER
        DIM SUM AS INTEGER

100     IF ( ICOUNT  > 5 ) THEN GOTO 99
        INPUT  A, B
        SUM = A + B
        PRINT A, B, SUM

        ICOUNT  = ICOUNT + 1
        GOTO 100

99      END
```



Counter controlled by FOR … NEXT loop

QBASIC can handle counter-controlled repetitive structures, or program loops, by a single statement called the FOR .. NEXT loop. It repeats a block of statements a specified number of times.

Syntax

```
FOR counter = start TO end [STEP increment]
   [statementblock]
NEXT [counter [,counter]...]
```

_ counter — A numeric variable used as the loop counter.
_ start and end — The initial and final values of the counter.
_ increment — The amount the counter is changed each time through the loop.

Example:

```
FOR i = 1 TO 15
        PRINT i
NEXT i
FOR i = 2 to – 6 STEP – 3
        PRINT i
NEXT i
```

In the first loop the loop will be executed 15 times.
In the second loop the loop will be executed thrice
1st loop: i = 2
2nd loop: i = 2 + (–3) = – 1
3rd loop: I = – 1 + (– 3) = – 4

- Where no *increment* value is given the increment is taken as 1. [(i.e. FOR i = 1 TO 15 STEP 1]
- When the value of '**i**' will exceed the **end** value the loop will stop.
- **Start**, **end**, **increment** values may be integer or real value or variables.

Nested FOR … NEXT loop

Several FOR .. NEXT loop may be used within the same program. They may follow one another, or they may be nested, i.e. one loop may remain within another.

The inner most loop will be executed first, then the next outer loop, and so on.

Example

```
DIM PRIN AS INTEGER, RATE AS INTEGER, YEAR AS INTEGER
DIM INTEREST AS SINGLE
FOR PRIN = 1000 TO 10000 STEP 1000
        FOR RATE = 10 TO 15
                FOR YEAR = 1 TO 3
                        INTEREST = PRIN * (RATE / 100.) * YEAR
                        PRINT PRIN, RATE, YEAR, INTEREST
                NEXT YEAR
        NEXT RATE
NEXT PRIN
END
```

Inner loop | Middle loop | Outer most loop

EXERCISES

11. Write a program that reads an integer N and prints the sum of the following:

(a) $1 + 2 + 3 + \ldots.. + N$
(b) $1^2 + 2^2 + 3^2 + \ldots + N^2$.
(c) $1^3 + 2^3 + 3^3 + \ldots + N^3$.
(d) $2 + 4 + 6 + 8 + \ldots + 2N$
(e) $1 + 3 + 5 + 7 + \ldots + (2N+1)$

Solution of (a)

```
DIM N AS INTEGER, SUM AS INTEGER
INPUT N
SUM = 0
FOR I = 1 TO N
        SUM = SUM + I
NEXT I
PRINT SUM
END
```

12. Write a program that reads an integer N and computes factorial of N (i.e. N!)
[Hints: PROD = 1*2*3*….* N]
13. To fit a straight line Y = mX + C through a set of n points (x_1,y_1) (x_2, y_2), (x_3, y_3),…,(x_n, y_n), the following formulae are used to determine the values of m and c :

$$m = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} y_i - \left(\sum_{i=1}^{n} x_i\right)^2} \qquad c = \frac{1}{n}\left[\sum_{i=1}^{n} y_i - m\sum_{i=1}^{n} x_i\right]$$

Write a program that first reads the number of points n, and then reads n pairs of values (x_i,y_i), and compares m and c.

FUNCTIONS AND SUBROUTINES
The functions and subroutines subprograms with which a program can be broken up into parts.
Function Subprogram
The function subprogram is the method by which a programmer can define function whose value cannot be specified by a single expression. The function will be called from the main program. Only one value will be returned by the function.
Syntax

```
FUNCTION name [(parameterlist)]
        [statementblock
        name = expression
        [statementblock]
END FUNCTION
```

_ *name* The name of the function and the data type it returns, specified by a data-type suffix (%, &, !, #, or $).
_ *parameterlist* One or more variables that specify parameters to be passed to the function when it is called

Example

Main program **PROG25.BAS**	Function **PROG25.BAS:MAX**
DIM X1 AS SINGLE, X2 AS SINGLE DIM X3 AS SINGLE INPUT X1, X2, X3 PRINT **MAX**(X1, X2, X3) END	FUNCTION **MAX** (A AS SINGLE, B AS SINGLE, C AS SINGLE) DIM LARGE AS SINGLE LARGE = A IF B > LARGE THEN LARGE = B IF C > LARGE THEN LARGE = C MAX = LARGE END FUNCTION

Subroutine
When more than one value is required to be returned to the main program then subroutine statement is writte.
Syntax

```
SUB name[(parameterlist)]
  [statementblock]
END SUB
```

_ name The name of the SUB procedure, up to 40 characters long, with no data type suffix.
_ parameterlist One or more variables that specify parameters to be passed to the SUB procedure when it is called:

Main program **PROG25.BAS**	Function **PROG25.BAS:LARGEST**
DIM X1 AS SINGLE, X2 AS SINGLE DIM X3 AS SINGLE, MAX AS SINGLE INPUT X1, X2, X3 CALL **LARGEST** (X1, X2, X3, MAX) PRINT MAX END	SUB **LARGEST** (A AS SINGLE, B AS SINGLE, C AS SINGLE, LARGE AS SINGLE) LARGE = A IF B > LARGE THEN LARGE = B IF C > LARGE THEN LARGE = C END SUB

Example
Write a program that reads a temperature and a letter "C" or "F" and convert it to the other scale.

The main program

```
REM This program reads a temperature and an alphabet "C" or "F" and
REM then it converts the temperature to the other scale
DIM TEMP AS SINGLE, UNIT AS STRING
DIM CENTIGRADE AS SINGLE, FAHRENHEIT AS SINGLE
INPUT "Give the temperature: "; TEMP
INPUT "Write C if it is Centigrade or F if it is in Fahrenheit:"; UNIT
IF UNIT = "C" THEN
        PRINT TEMP,"deg C is equivalent to ", FAHRENHEIT(TEMP), "deg F"
ENDIF
IF UNIT = "F" THEN
        PRINT TEMP "deg F is equivalent to ", CENTIGRADE(TEMP), "deg C"
ENDIF
END
```

The functions

```
FUNCTION FAHRENHEIT (X AS SINGLE)
        FAHRENHEIT = X*9/5 + 32
END FUNCTION

FUNCTION CENTIGRADE (X AS SINGLE)
        CENTIGRADE = (X – 32)*5/9
END FUNCTION
```

Same program but by using subroutine

*The main program************

```
REM This program reads a temperature and an alphabet "C" or "F" and
REM then it converts the temperature to the other scale
DIM TEMP AS SINGLE, UNIT AS STRING
DIM CENTIGRADE AS SINGLE, FAHRENHEIT AS SINGLE
INPUT "Give the temperature: "; TEMP
INPUT "Write C if it is in Centigrade or F if it is in Fahrenheit:"; UNIT

CALL CONVERT (TEMP, UNIT, VALUE)

IF UNIT = "C" THEN
        PRINT VALUE, "deg F"
ELSE
        PRINT VALUE, "deg C"
END IF
END
```

*The sub-routine program**********

```
SUB CONVERT (TEMP AS SINGLE, UNIT AS STRING, VALUE AS SINGLE)
IF UNIT = "C' THEN
        VALUE = X*9/5 + 32
ELSE
        VALUE = (X – 32)*5/9
ENDIF
END SUB
```

FORMATTING

LOCATE

By using LOCATE statement the display of text can be started from a predefined position on the screen.

LOCATE moves the cursor to a specified position on the screen

LOCATE [row%] [,[column%]

– row% is the row number (maximum 25 rows)
– column% is column number (maximum 80 columns)

Example

```
CLS
LOCATE 2, 10
PRINT "Enter X value"
LOCATE 2, 25
INPUT "", j
LOCATE 3, 10
PRINT "The value you have entered is "; j
```

GRAPHICS

SCREEN statement

SCREEN statement sets the screen mode and other characteristics of the screen. The type of graphical display depends on the screen mode. Here we will use screen mode 9. The characteristics of screen mode 9 are as follows:

SCREEN 9:

640 x 350 graphics
80 x 25 text format
16 colors assigned to 4 attributes (64K adapter memory), or
64 colors assigned to 16 attributes (more than 64K adapter memory)

COLOR statement

COLOR sets the screen display colors. It can set the foreground color (i.e. the color of the characters and lines) and (or) the color of background.

Example

```
SCREEN 9
COLOR 4, 7          ‘ This will set the foreground color in red and background color in white
PRINT "Institute"   ‘ Institute will be written in red color against a white background
COLOR 1,7           ‘ This will set the foreground color in blue and background color in white
PRINT "Institute"   ‘ Institute will be written in blue color against a white background
COLOR 1,0           'This will set the foreground color in blue and background color in black
PRINT "Institute"   ‘ Institute will be written in blue color against a black background
```

QBASIC color values	Name of the color	QBASIC color values	Name of the color
0	Black	8	Gray
1	Blue	9	Light Blue
2	Green	10	Light Green
3	Cyan	11	Light Cyan
4	Red	12	Light Red
5	Magenta	13	Light Magenta
6	Brown	14	Yellow
7	White	15	High-intensity white

LINE statement

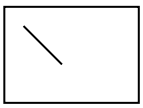
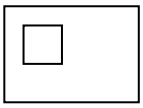
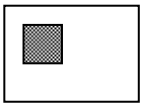
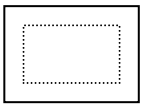
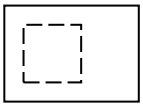
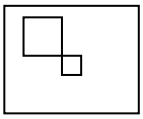
LINE statement draws a line or rectangle on the screen.

Syntax

LINE [(x1!,y1!)] – (x2!,y2!) [,[color%] [,[B | BF] [,style%]]]

STEP	Specifies that coordinates are relative to the current graphics cursor position.
(x1!,y1!)	The screen coordinates of the start of the line and of (x2!,y2!) the end of the line.
color%	A color attribute that sets the color of the line or rectangle. The available color attributes depend on graphics adapter and the screen mode set by the SCREEN statement.
B	Draws a rectangle instead of a line.
BF	Draws a filled box.
style%	A 16-bit value whose bits set whether or not pixels are drawn. Use to draw dashed or dotted lines.

Examples

Sl	Statement	Comment	Display
1	SCREEN 9	Text mode 25 rows and 80 cols. Graphics 640 x 350	
2	LINE (10, 10)-(100, 100)	'Draws a line from (10,10) to (100,100)	
3	LINE (10, 10)-(100, 100),4	'Draws a line from (10,10) to (100,100) with red color.	
4	LINE (10, 10)-(100, 100),4, B	'Draws a box from (10,10) to (100,100) with red color.	
5	LINE (10, 10)-(100, 100),4, BF	'Draws a box from (10,10) to (100,100) filled with red color.	
6	LINE (10, 10)-(100, 100), , BF	'Draws a box from (10,10) to (100,100) filled with default color.	
7	LINE (10, 10)-(100, 100), 4, B, 1	'Draws a box from (10,10) to (100,100) with dotted border.	
8	LINE (10, 10)-(100, 100), 4, B, 3	'Draws a box from (10,10) to (100,100) with dashed border.	
9	LINE – (150, 150), 4, B	'Draws a box from previous point (i.e. 100,100) to (150,150).	

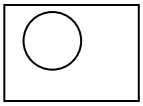
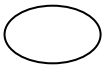
CIRCLE

Draws a circle or ellipse on the screen.

CIRCLE (x!,y!), radius![,[color%] [,[start!] [,[end!] [,aspect!]]]

STEP	Specifies that coordinates are relative to the current graphics cursor position.
(x!,y!)	The coordinates for the center of the circle or ellipse.
radius!	The radius of the circle or ellipse in the units of the current coordinate system, determined by the most recent SCREEN, VIEW, and WINDOW statements.
color%	A color attribute that sets the circle's color. The available color attributes depend on your graphics adapter and the screen mode set by the most recent SCREEN statement.
start!	The starting angle for the arc, in radians.
end!	The ending angle for the arc, in radians.
aspect!	The ratio of the length of the y axis to the length of the x axis, used to draw ellipses.

Examples

Sl	Statement	Comment	Display
1	SCREEN 9	Text mode 25 rows and 80 cols. Graphics 640 x 350	
2	CIRCLE (100, 100), 50	'Draws a circle with center (100,100) and radius of 50 pixels.	
3	CIRCLE (100, 100), 50, 4	'Draws a red circle with center (100,100) and radius of 50 pixels.	
4	CIRCLE (100, 100), 50, 4, 0, 1.57	'Draws a red arc with center (100,100) and radius of 50 pixels starting from 0 radian to π/2 radian	
5	CIRCLE (100, 100), 50, 4, , , 0.5	'Draws a red ellipse with center (100,100) and y axis / x axis ratio of 0.5.	

14. 500 mg of a drug is administered orally to a patient. Blood samples were taken from every hour. Concentration of the drug (c) in the plasma is plotted against time (t) to obtain a curve. Write a program to approximate the area under the curve from t =0 to t=24 hours, using the trapezoidal rule.
The area of a trapezium = ½ (a + b)h