INDEX

7	Develop, implement and execute a C program that reads two matrices A(m*N) original and reversed strings. and B(p*q) and compute the product A and B. Read matrix A in row major order and matrix B in column major order. Print both the input matrices and resultant matrix with suitable headings in matrix format. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.	34-35
8	Develop, implement and execute a C program to search a Name in a list of names using Binary search technique.	36
9	Write and execute a C program that a. Implements string copy operation STRCOPY(str1,str2) that copies a string str1 to another string str2 without using library function. b. Reads a sentence and print frequency of each of the vowels and total count of consonants.	37
10	a. Design and develop a C function RightShift(x,n) that takes two integets x and n as input and returns value of the integer x rotated to the right by n positions. Assume the integers are unsigned. Write a C program that invokes this function with different values for x and n. Tabulate the results with suitable headings. b. Design and develop a C function isprime(num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given range.	38-39
11	Draw the flowchart and write a recursive C function to find the factorial of a number n!, defined by fact(n)=1 if n=0. Otherwise fact(n)=n*fact(n-1). Using this function, write a C program to compute the binomial coefficient ${}^{n}C_{r}$. Tabulate the results for different values of n and r with suitable messages.	40-41
12	Given two university information files "studentname.txt" and "usn.txt" that contains student name and usn respectively. Write a C program to create a new file called "output.txt" and copy the content of files "studentname.txt" and "usn.txt" into output file in the sequence shown below. Display the output file "output.txt" onto the screen. --- Student Name USN --- Name1 USN1 Name2 USN2	42
13	Write a C program to maintain record of "n" student details using an array of structures with four fields (Roll number, Name, Marks and Grade). Each field is of an appropriate data type. Print the marks of the student name as input	43
14	Write a C program using pointers to compute the sum ,mean and standard deviation of all elements stored in an array of n real numbers	44

PART – A: Personal Computer and its Accessories

Laboratory Session-1: Write-up on Functional block diagram of Computer, CPU, Buses, Mother Board, Chip sets, Operating System & types of OS, Basics of Networking & Topology and NIC.

The computer defined

A computer is an electronic device that processes data, converting it into information that is useful to people. Any computer regardless of its type is controlled by programmed instructions, which give the machine a purpose and tell it what to do.
A computer performs basically five major functions as shown in figure 1.1 they are:

1. It accepts data or instructions by the user in the way of input.
2. It stores the data.
3. It can processes the data as required by the user.
4. It gives result in the form of output.
5. It controls all operations inside a computer.

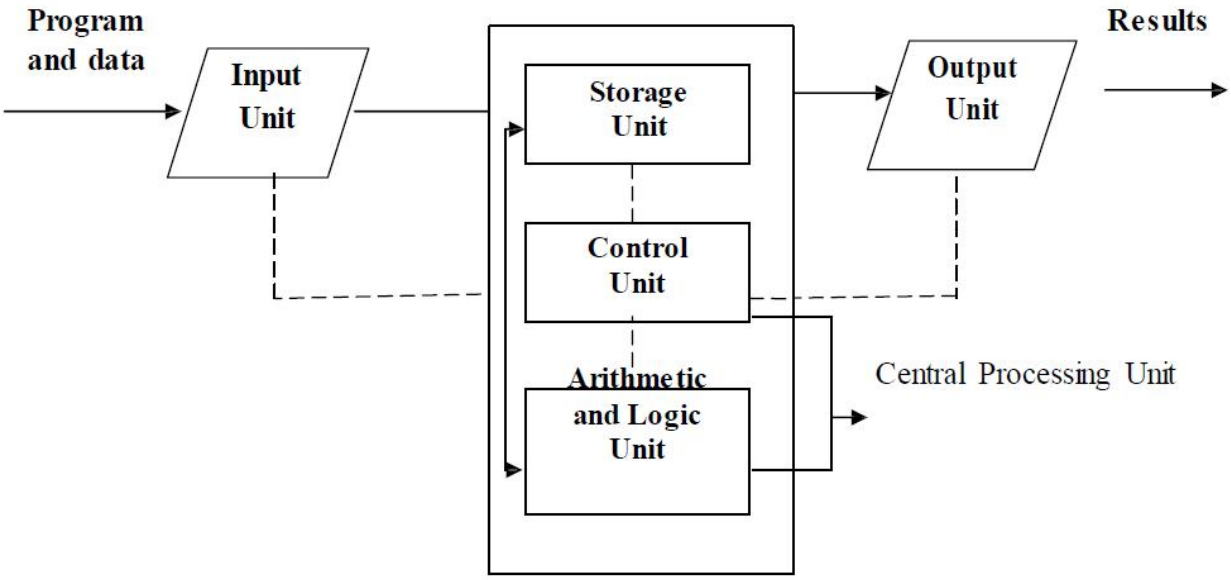


Figure 1.1: Functional block diagram of computer.

1. Input: This is the process of entering data and programs in to the computer system. It takes as inputs raw data and performs some processing giving out processed data. Standard input device is keyboard.

2. Storage: The process of saving data and instructions permanently is known as storage. Data has to be fed into the system before the actual processing starts. It is because the processing speed of Central Processing Unit (CPU) is so fast that the data has to be provided to CPU with the same speed. Therefore the data is first stored in the storage unit for faster access and processing. This storage unit or the primary storage of the computer system is designed to do the above functionality. It provides space for storing data and instructions.

The storage unit performs the following major functions:

• All data and instructions are stored here before and after processing.

• Intermediate results of processing are also stored here.

3. Processing: The task of performing operations like arithmetic and logical operations is called processing. The Central Processing Unit (CPU) takes data and instructions from the storage unit and makes all sorts of calculations based on the instructions given and the type of data provided. It is then sent back to the storage unit.

4. Output: This is the process of producing results from the data for getting useful information. Similarly the output produced by the computer after processing must also be kept somewhere inside the computer before being given to you in human readable form. Again the output is also stored inside the computer for further processing.

5. Control: The manner how instructions are executed and the above operations are performed. Controlling of all operations like input, processing and output are performed by control unit. It takes care of step by step processing of all operations inside the computer.

FUNCTIONAL UNITS

In order to carry out the operations mentioned in the previous section the computer allocates the task between its various functional units. The computer system is divided into three separate units for its operation. They are

1) Arithmetic logical unit

2) Control unit.

3) Central processing unit.

Arithmetic Logical Unit (ALU) Logical Unit

Logical Unit: After you enter data through the input device it is stored in the primary storage unit. The actual processing of the data and instruction are performed by Arithmetic Logical Unit. The major operations performed by the ALU are addition, subtraction, multiplication, division, logic and comparison. Data is transferred to ALU from storage unit when required. After processing the output is returned back to storage unit for further processing or getting stored.

Control Unit (CU)

The next component of computer is the Control Unit, which acts like the supervisor seeing that things are done in proper fashion. Control Unit is responsible for co ordinating various operations using time signal. The control unit determines the sequence in which computer programs and instructions are executed. Things like processing of programs stored in the main memory

interpretation of the instructions and issuing of signals for other units of the computer to execute them. It also acts as a switch board operator when several users access the computer simultaneously. Thereby it coordinates the activities of computer's peripheral equipment as they perform the input and output.

Central Processing Unit (CPU)

The ALU and the CU of a computer system are jointly known as the central processing unit. You may call CPU as the brain of any computer system. It is just like brain that takes all major decisions, makes all sorts of calculations and directs different parts of the computer functions by activating and controlling the operations.

BUSES, MOTHERBOARD AND CHIPSET

BUSES:

In computer architecture, a bus is a communication system that transfers data between components inside a computer or between computers. Buses can be parallel buses, which carry data words in parallel on multiple wires or serial buses which carry data in bit-serial form.

MOTHERBOARD:

The Motherboard of a computer also known as the System Board, the main board or Circuit board is the platform on which the various components that make up the hardware of the computer are connected. The entire computer circuitry finds their base on the System board. It could be said to be the most important part of the computer. A typical ATX PC motherboard with constituent components is shown in figure 1.2.The components of an ATX Motherboard are:

1. Mouse & keyboard
2. USB
3. Parallel port
4. CPU Chip
5. RAM slots
6. Floppy controller
7. IDE controller
8. PCI slot
9. ISA slot
10. CMOS Battery
11. AGP slot
12. CPU slot
13. Power supply plug in

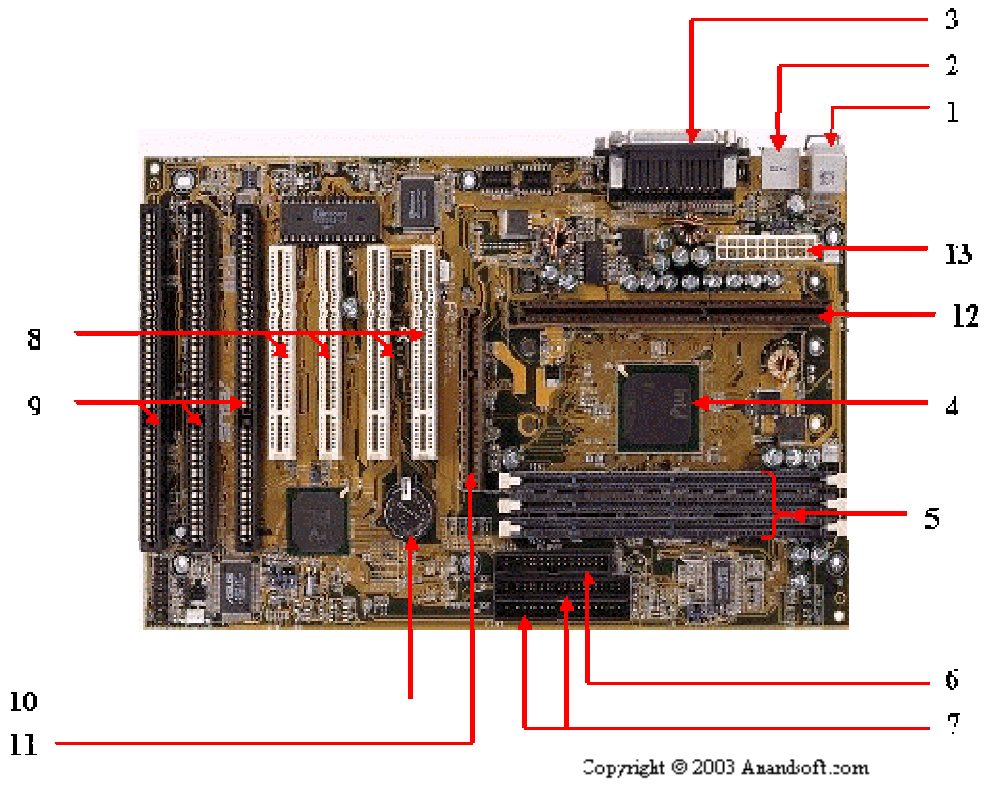
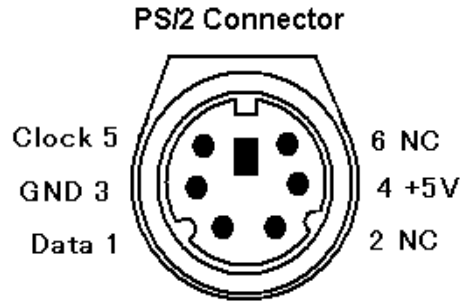


Figure 1.2: Typical ATX Motherboard

1. Mouse & keyboard: Keyboard Connectors are two types basically. All PCs have a Key board port connected directly to the motherboard. The oldest, but still quite common type, is a special DIN, and most PCs until recently retained this style connector. The AT-style keyboard connector is quickly disappearing, being replaced by the smaller mini DIN PS/2-style keyboard connector.



You can use an AT-style keyboard with a PS/2-style socket (or the other way around) by using a converter. Although the AT connector is unique in PCs, the PS/2-style mini-DIN is also used in more modern PCs for the mouse. Fortunately, most PCs that use the mini-DIN for both the keyboard and mouse clearly mark each mini-DIN socket as to its correct use. Some keyboards have a USB connection, but these are fairly rare compared to the PS/2 connection keyboards.

2. **USB (Universal Serial Bus):** USB is the General-purpose connection for PC. You can find USB versions of many different devices, such as mice, keyboards, scanners, cameras, and even printers. A USB connector's distinctive rectangular shape makes it easily recognizable. USB has a number of features that makes it particularly popular on PCs. First, USB devices are hot swappable. You can insert or remove them without restarting your system.

3. Parallel port: Most printers use a special connector called a parallel port. Parallel port carry data on more than one wire, as opposed to the serial port, which uses only one wire.

4. CPU Chip: The *central processing unit,* also called the *microprocessor* performs all the calculations that take place inside a pc. CPUs come in Variety of shapes and sizes. Modern CPUs generate a lot of heat and thus require a cooling fan or heat sink. The cooling device (such as a cooling fan) is removable, although some CPU manufactures sell the CPU with a fan permanently attached.

5. RAM slots: Random-Access Memory (RAM) stores programs and data currently being used by the CPU. RAM is measured in units called bytes. RAM has been packaged in many different ways. The most current package is called a 168-pin DIMM (Dual Inline Memory module).

6. Floppy controller: The floppy drive connects to the computer via a 34-pin *ribbon cable,* which in turn connects to the motherboard. A *floppy controller* is one that is used to control the floppy drive.

7. IDE controller: Industry standards define two common types of hard drives: EIDE and SCSI. Majority of the PCs use EIDE drives. SCSI drives show up in high end PCs such as network servers or graphical workstations. The EIDE drive connects to the hard drive via a 2-inch-wide, 40-pin ribbon cable, which in turn connects to the motherboard. *IDE controller* is responsible for controlling the hard drive.

8. PCI slot: Intel introduced the *Peripheral component interconnect* bus protocol. The PCI bus is used to connect I/O devices (such as NIC or RAID controllers) to the main logic of the computer. PCI bus has replaced the ISA bus.

9. ISA slot: (Industry Standard Architecture) It is the standard architecture of the Expansion bus. Motherboard may contain some slots to connect ISA compatible cards.

10. CMOS Battery: To provide CMOS with the power when the computer is turned off all motherboards comes with a battery. These batteries mount on the motherboard in one of three ways: the obsolete external battery, the most common onboard battery, and built-in battery.

11. AGP slot: If you have a modern motherboard, you will almost certainly notice a single connector that looks like a PCI slot, but is slightly shorter and usually brown. You also probably have a video card inserted into this slot. This is an Advanced Graphics Port (AGP) slot

12. CPU slot: To install the CPU, just slide it straight down into the slot. Special notches in the slot make it impossible to install them incorrectly. So remember if it does not go easily, it is probably not correct.

13. Power supply plug in: The Power supply, as its name implies, provides the necessary electrical power to make the pc operate. The power supply takes standard 110-V AC power and converts into +/-12-Volt, +/-5-Volt, and 3.3-Volt DC power.

CHIP SETS:

The motherboard's busses are regulated by a number of controllers. These are small circuits which have been designed to look after a particular job, like moving data to and from EIDE devices (hard disks, etc.).

A number of controllers are needed on a motherboard, as there are many different types of hardware devices which all need to be able to communicate with each other. Most of these controller functions are grouped together into a couple of large chips, which together comprise the *chip set*.

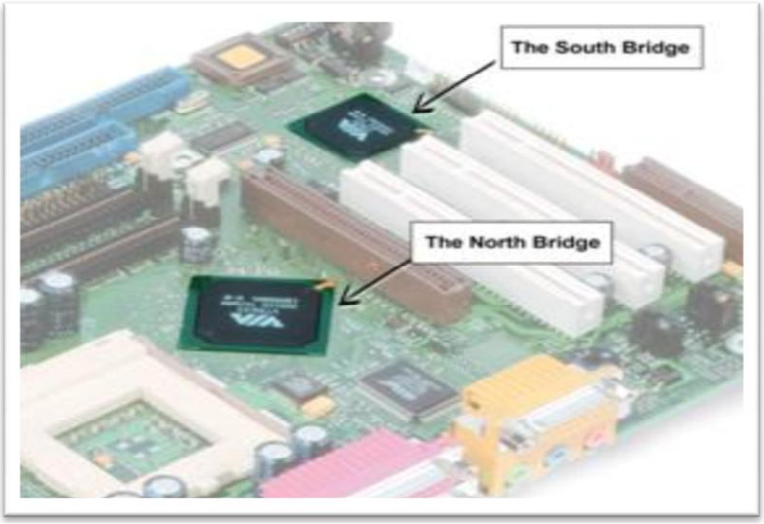


Figure 1.3. The two chips which make up the chipset, and which connect the motherboard's busses.

The most widespread chipset architecture consists of two chips, usually called the *north* and *south bridges*. This division applies to the most popular chipsets from VIA and Intel. The north bridge and south bridge are connected by a powerful bus, which sometimes is called a *link channel*:

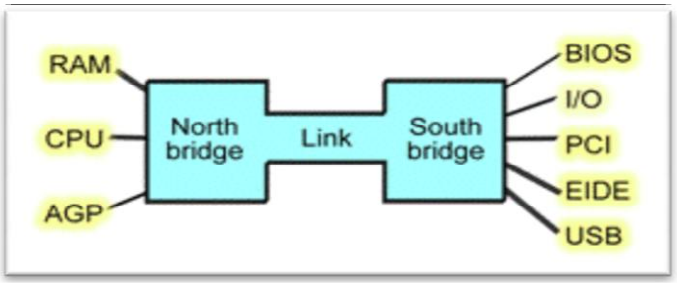


Figure.1.4. The north bridge and south bridge share the work of managing the data traffic on the motherboard.

OPERATING SYSTEM (OS) AND TYPES OF OS

Operating system is a platform between hardware and user which is responsible for the management and coordination of activities and the sharing of the resources of a computer. It hosts the several applications that run on a computer and handles the operations of computer hardware. There are different types of operating systems. These are as follows:

1. **Real-time Operating System:** It is a multitasking operating system that aims at executing real-time applications.

2. **Multi-user and Single-user Operating Systems:** The operating systems of this type allow a multiple users to access a computer system concurrently.

3. **Multi-tasking and Single-tasking Operating Systems:** When a single program is allowed to run at a time, the system is grouped under a single-tasking system, while in case the operating system allows the execution of multiple tasks at one time, it is classified as a multi-tasking operating system.

4. **Distributed Operating System:** An operating system that manages a group of independent computers and makes them appear to be a single computer is known as a distributed operating system.

5. **Embedded System:** The operating systems designed for being used in embedded computer systems are known as embedded operating systems.

NETWORK, TOPOLOGY OF NETWORKS AND NIC

A computer network is a group of computer systems and other computing hardware devices that are linked together through communication channels to facilitate communication and resource-sharing among a wide range of users.

Networks are used to:

Facilitate communication via email, video conferencing, instant messaging, etc. Enable multiple users to share a single hardware device like a printer or scanner Enable file sharing across the network

Allow for the sharing of software or operating programs on remote systems

Make information easier to access and maintain among network users.

There are many types of networks, including:

Local Area Networks (LAN)

Wide Area Networks (WAN)

Metropolitan Area Networks (MAN)

Internetwork

NETWORK TOPOLOGIES

In communication networks, a topology is a usually schematic description of the arrangement of a network, including its nodes and connecting lines. There are two ways of defining network geometry: the physical topology and the logical (or signal) topology.

Physical topologies:

The physical topologies of a network are the actual geometric layout of workstations. There are several common physical topologies, as described below and as shown in the illustration.

Bus topology:

In the bus network topology, every workstation is connected to a main cable called the bus. Therefore, in effect, each workstation is directly connected to every other workstation in the network.

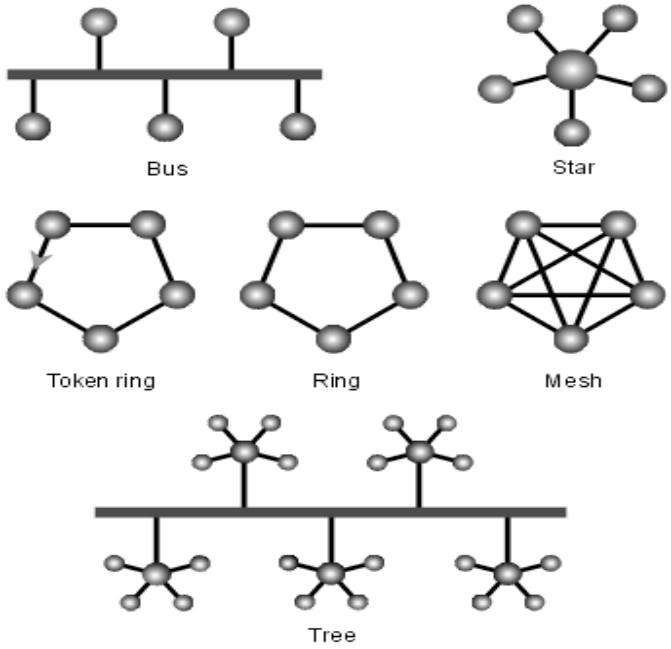
Star topology:

In the star network topology, there is a central computer or server to which all the workstations are directly connected. Every workstation is indirectly connected to every other through the central computer.

Token Ring topology:

In the ring network topology, the workstations are connected in a closed loop configuration. Adjacent pairs of workstations are directly connected. Other pairs of workstations are indirectly connected, the data passing through one or more intermediate nodes.

If a Token Ring protocol is used in a star or ring topology, the signal travels in only one direction, carried by a so-called token from node to node.



Mesh topology:

The mesh network topology employs either of two schemes, called full mesh and partial mesh. In the full mesh topology, each workstation is connected directly to each of the others. In the partial mesh topology, some workstations are connected to all the others, and some are connected only to those other nodes with which they exchange the most data.

Tree topology:

The tree network topology uses two or more star networks connected together. The central computers of the star networks are connected to a main bus. Thus, a tree network is a bus network of star networks.

Logical Topology

Logical (or signal) topology refers to the nature of the paths the signals follow from node to node. In many instances, the logical topology is the same as the physical topology. But this is not always the case. For example, some networks are physically laid out in a star configuration, but they operate logically as bus or ring networks.

NIC (Network Interface Card)

To connect to a network, a computer uses a network interface card (NIC). A NIC controls the wired and wireless connections of a computer to exchange information with other computers and the Internet.

A computer uses a network interface card (NIC) to become part of a network. The NIC contains the electronic circuitry required to communicate using a wired connection (e.g., Ethernet) or a wireless connection (e.g., Wi-Fi). A network interface card is also known as a network interface controller, network adapter, or Local Area Network (LAN) adapter.

Laboratory Session 2: Write-up on RAM, SDRAM, FLASH memory, Hard disks, Optical media, CD-ROM/R/RW, DVDs, Flash drives, Keyboard, Mouse, Printers and Plotters.

A memory is just like a human brain. It is used to store data and instructions. Computer memory is the storage space in computer where data is to be processed and instructions required for processing are stored. The memory is divided into large number of small parts. Each part is called cell. Each location or cell has a unique address, which varies from zero to memory size minus one.

For example, if computer has 64k words, then this memory unit has 64 * 1024=65536 memory location. The address of these locations varies from 0 to 65535.

Memory is primarily of three types: Cache Memory

Primary Memory/Main Memory

Secondary Memory

Cache Memory

Cache memory is a very high speed semiconductor memory, which can speed up CPU. It acts as a buffer between the CPU and main memory. It is used to hold those parts of data and program which are most frequently used by CPU. The parts of data and programs are transferred from disk to cache memory by operating system, from where CPU can access them.

ADVANTAGE

Cache memory is faster than main memory.

It consumes less access time as compared to main memory.

It stores the program that can be executed within a short period of time. It stores data for

temporary use.

DISADVANTAGE:

Cache memory has limited capacity. It is very expensive.

Primary Memory (Main Memory)

Primary memory holds only those data and instructions on which computer is currently working. It has limited capacity and data gets lost when power is switched off. It is generally made up of semiconductor device. These memories are not as fast as registers. The data and instructions required to be processed earlier reside in main memory. It is divided into two subcategories RAM and ROM.

Characteristic of Main Memory

These are semiconductor memories.

It is known as main memory.

Usually volatile memory.

Data is lost in case power is switched off.

It is working memory of the computer.

Faster than secondary memories.

A computer cannot run without primary memory.

Secondary Memory

This type of memory is also known as external memory or non-volatile. It is slower than main memory. These are used for storing Data/Information permanently. CPU directly does not access these memories; instead they are accessed via input-output routines. Contents of secondary memories are first transferred to main memory and then CPU can access it. For example, disk, CD-ROM, DVD, etc.

Characteristics of Secondary Memory

These are magnetic and optical memories.

It is known as backup memory.

It is non-volatile memory.

Data is permanently stored even if power is switched off.

It is used for storage of the data in the computer.

Computer may run without secondary memory.

Slower than primary memory.

Random Access Memory (RAM)

A RAM constitutes the internal memory of the CPU for storing data, program and program result. It is read/write memory. Since access time in RAM is independent of the address to the word that is, each storage location inside the memory is as easy to reach as other location & takes the same amount of time. We can reach into the memory at random & extremely fast but can also be quite expensive.

RAM is volatile, i.e., data stored in it is lost when we switch off the computer or if there is a power failure. Hence, a backup uninterruptible power system (UPS) is often used with computers. RAM is small, both in terms of its physical size and in the amount of data it can hold.

RAM is of two types: Static RAM (SRAM)
Dynamic RAM (DRAM)

Static RAM (SRAM)

The word **static** indicates that the memory retains its contents as long as power remains applied. However, data is lost when the power gets down due to volatile nature. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not have to be refreshed on a regular basis.Because of the extra space in the matrix, SRAM uses more chips than DRAM for the same amount of storage space, thus making the manufacturing costs higher.

Characteristics of the Static RAM:

It has long data lifetime

There is no need to refresh Faster

Used as cache memory

Large size

Expensive

High power consumption

Dynamic RAM (DRAM)

DRAM, unlike SRAM, must be continually **refreshed** in order for it to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory because it is cheap and small. All DRAMs are made up of memory cells. These cells are composed of one capacitor and one transistor.

Characteristics of the Dynamic RAM:

It has short data lifetime

Need to refresh continuously

Slower as compared to SRAM

Used as RAM

Lesser in size

Less expensive

Less power consumption

Synchronous DRAM (SDRAM):

A type of DRAM that can run at much higher clock speeds than conventional memory. SDRAM actually synchronizes itself with the CPU's bus and is capable of running at 133 MHz, about three times faster than conventional RAM, and about twice as fast DRAM. SDRAM is replacing DRAM in many newer computers.

Characteristics of SDRAM:

It starts working as soon as it receives an instruction.

It synchronizes its operation with the computer clock and the rest of the system.
The point of synchronization is pipe lining.
SDRAM inherits all the features of DRAM (which inherits all the features of RAM).

Read Only Memory (ROM)

The memory from which we can only read but cannot write on it, this type of memory is non-volatile. The information is stored permanently in such memories during manufacture.A ROM stores such instructions as are required to start computer when electricity is first turned on, this operation is referred to as bootstrap. ROM chip are not only used in the computer but also in other electronic items like washing machine and microwave oven.
Following are the various types of ROM:

MROM (Masked ROM): The very first ROMs were hard-wired devices that contained a pre-programmed set of data or instructions. These kinds of ROMs are known as masked ROMs. It is inexpensive ROM.

PROM (Programmable Read only Memory): PROM is read-only memory that can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM programmer. Inside the PROM chip, there are small fuses, which are burnt open during programming. It can be programmed only once and is not erasable.

EPROM (Erasable and Programmable Read Only Memory): The EPROM can be erased by exposing it to ultra-violet light for duration of up to 40 minutes. Usually, an EPROM eraser achieves this function. During programming, an electrical charge is trapped in an insulated gate region. The charge is retained for more than ten years because the charge has no leakage path. For erasing this charge, ultra-violet light is passed through a quartz crystal window (lid). This exposure to ultra-violet light dissipates the charge. During normal use the quartz lid is sealed with a sticker.

EEPROM (Electrically Erasable and Programmable Read Only Memory): The EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (milli second). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of re-programming is flexible but slow.

Advantages of ROM:

Non-volatile in nature
These cannot be accidentally changed
Cheaper than RAMs

Easy to test

More Reliable than RAMs

These are static and do not require refreshing

Its contents are always known and can be verified

Flash Memory

Flash memory is a form of EEPROM that allows multiple memory locations to be erased or written in one programming operation. In other terms, it is a form of rewritable memory chip that, unlike a RAM chip, power supply is not required to hold the contents. It is commonly used in memory cards, USB flash drives, MP3 players, digital cameras and mobile phones.

Regular EEPROM erases content byte by byte; most flash memory erases data in whole blocks, making it suitable for use with applications where large amounts of data require frequent updates. Inside the flash chip, data is stored in cells protected by floating gates. Tunneling electrons change the gate's electronic charge in "a flash" (hence the name), clearing the cell of its contents so it can be rewritten.

Hard Disk

A hard disk or also known as hard disk drive (HDD), is one of the vital parts of a computer and any other device that requires some kind of data storage. They can be found either built into the device in question or they can be used as an external storage part.

Characteristics of HDD

Data transfer rate-. This is the measured amount of data that the hard disk can transfer per second. A 7200 rpm hard disk has a data transfer rate of about 70 mb per second. This includes both reading and writing from a local disk.

Seek time- Ranging from 3 ms (high performance servers) to 15 ms (mobile drives). This is the time required for the read/write head of the disk to go to the correct point to start the reading or writing process.

Power Consumption- The speed and capacity of a hard disk are the main factors that affect its power consumption rate. Slower hard disks consume less power compared to higher speed hard disks. Audible Noise -is something that manufacturers want to decrease to a minimum level. We might have not heard the audible noise of the hard disk of PC or laptop but definitely have noticed the noise of an external hard disk drive before.

Optical media

Media, in the computer world, refers to various types of data storage. For example, hard drives, CDs, DVDs, and USB drives are all different types of media. Optical media refers to discs that are read by a laser. This includes CD-ROMs, DVD-ROMs, and all the variations of the two formats -- CD- R, CD-RW, DVD-R, DVD+R, Blu-ray, and many others.

Optical media typically does not have as fast of a seek time as hard drives (the time it takes to access information on different parts of the disk), but it has many other advantages. Because optical discs are not based on magnetic charges like hard drives are, the discs are less likely to lose their data and have a longer shelf life -- around seven times longer than magnetic media. The discs are also more durable than hard drives and are much cheaper to produce, making them great for backups and for transferring small amounts of data between different computers.

CD-ROM/R/RW, DVDs

CD-ROM- Stands for "Compact Disc Read-Only Memory." A CD-ROM is a CD that can be read by a computer with an optical drive. The "ROM" part of the term means the data on the disc is "read-only," or cannot be altered or erased. Because of this feature and their large capacity, CD-ROMs are a great media format for retail software. The first CD-ROMs could hold about 600 MB of data, but now they can hold up to 700 MB.

CD-R- Stands for "Compact Disc Recordable." CD-R discs are blank CDs that can record data written by a CD burner. The word "recordable" is used because CD-Rs are often used to record audio, which can be played back by most CD players. However, many other kinds of data can also be written to a CD-R, so the discs are also referred to as "writable CDs."

CD-RW- Stands for "Compact Disc Re-Writable." A CD-RW is a blank CD that can be written to by a CD burner. Unlike a CD-R (CD-Recordable), a CD-RW can be written to multiple times. The data burned on a CD-RW cannot be changed, but it can be erased. Therefore, a CD-RW should be erased every time if we want to make changes to the files or add any new data. Because CD-RWs can be erased, they don't store data reliably for as long as CD-Rs do, therefore CD-Rs are used for long-term backups.

DVDs

Stands for “Digital Versatile Disc”. A DVD is a high-capacity optical disc that looks like a CD, but can store much more information. While a CD can store 650 to 700 MB of data, a single-layer, single-sided DVD can store 4.7 GB of data. This enables massive computer applications and full-length movies to be stored on a single DVD.

The advanced DVD formats are even more amazing. There is a two-layer standard that doubles the single- sided capacity to 8.5 GB. These disks can also be double-sided, ramping up the maximum storage on a single disc to 17 GB. To read DVDs in a computer it should have DVD-ROM drive.

Flash drives

Flash drives have many names —pen drives, and USB keychain drives these are small data storage device that uses flash memory and has a built-in USB connection. Flash drives are typically no more than two to three inches in length and less than an inch in width.

Early flash drives could store only a few megabytes of data, but modern flash drives can store several gigabytes of information. Since they are small in size but have large storage capacities, flash drives have replaced most previous portable data storage mediums such as floppy disks because they have a built- in USB connection, flash drives also don't require a special disk drive to be used. Instead, they can be used on any computer with a USB port, which nearly all modern computers have.

Keyboard

Most common and very popular input device is keyboard. The keyboard helps in inputting the data to the computer. The layout of the keyboard is like that of traditional typewriter, although there are some additional keys provided for performing some additional functions.Keyboards are of two sizes 84 keys or

101/102 keys, but now 104 keys or 108 keys keyboard is also available for Windows and Internet.

The keys are following

Sr. No.	Keys	Description
1	Typing Keys	These keys include the letter keys (A-Z) and digits keys (0-9) which generally give same layout as that of typewriters.
2	Numeric Keypad	It is used to enter numeric data or cursor movement. Generally, it consists of a set of 17 keys that are laid out in the same configuration used by most adding machine and calculators
3	Function Keys	The twelve functions keys are present on the keyboard. These are arranged in a row along the top of the keyboard. Each function key has unique meaning and is used for some specific purpose.
4	Control keys	These keys provide cursor and screen control. It includes four directional arrow key. Control keys also include Home, End, Insert, Delete, Page Up, Page Down, Control(Ctrl), Alternate(Alt), Escape(Esc).
5	Special Purpose Keys	Keyboard also contains some special purpose keys such as Enter, Shift, Caps Lock, Num Lock, Space bar, Tab, and Print Screen

Mouse

Mouse is most popular Pointing device. It is a very famous cursor-control device. It is a small palm size box with a round ball at its base which senses the movement of mouse and sends corresponding signals to CPU on pressing the buttons.Generally, it has two buttons called left and right button and scroll bar is present at the mid. Mouse can be used to control the position of cursor on screen, but it cannot be used to enter text into the computer.

ADVANTAGES

Easy to use

Not very expensive

Moves the cursor faster than the arrow keys of keyboard.

Printers

Printer is the most important output device, which is used to print information on paper.There are two types of printers:

Impact Printers
Non-Impact Printers

Impact Printers- The printers that print the characters by striking against the ribbon and onto the paper, are called impact printers.

Characteristics of Impact Printers are the following:

Very low consumable costs

Impact printers are very noisy

Useful for bulk printing due to low cost

There is physical contact with the paper to produce an image

These printers are of two types: Character printers
Line printers

Character Printers:

Character Printers are printers, which print one character at a time. These are of further two types:

Dot Matrix Printer (DMP)

Daisy Wheel.

Dot Matrix Printer

In the market, one of the most popular printer is Dot Matrix Printer because of their ease of printing features and economical price. Each character printed is in form of pattern of Dot's and head consists of a Matrix of Pins of size(5*7, 7*9, 9*7 or 9*9) which comes out to form a character that is why it is called Dot Matrix Printer.

Advantages

Inexpensive

widely Used

Other language characters can be printed

Disadvantages

Slow Speed

Poor Quality

Daisy Wheel

Head is lying on a wheel and Pins corresponding to characters are like petals of Daisy (flower name) that is why it is called Daisy Wheel Printer. These printers are generally used for word-processing in offices which require a few letters to be send here and there with very nice quality representation.

Advantages

More reliable than DMP's

Better quality

The fonts of character can be easily changed.

Disadvantages

Slower than DMP's

Noisy

More expensive than DMP's.

Line Printers

Line printers are printers, which print one line at a time. These are of further two types:

Drum Printer

Chain Printer

Drum Printer

This printer is like a drum in shape so it called drum printer. The surface of drum is divided into number of tracks. Total tracks are equal to size of paper, i.e., for a paper width of 132 characters, Drum will have 132 tracks. A character set is embossed on track. The different character sets available in market are 48 character set, 64 and 96 characters set. One rotation of drum prints one line. Drum Printers are fast in speed and prints between 300 to 2000 lines per minute.

Advantages

Very high speed

Disadvantage

Very expensive

Characters fonts cannot be changed.

Chain Printer

In this printer, chain of character sets is used so it called Chain Printers. A standard character set may have 48, 64, 96 characters.

Advantages

Character fonts can easily be changed.

Different languages can be used with the same printer.

Disadvantages

Noisy

Do not have the ability to print any shape of characters.

Non-impact Printers

The printers that print the characters without striking against the ribbon and onto the paper are called Non- impact Printers. These printers print a complete page at a time, also called as Page Printers.

These printers are of two types: Laser Printers

Inkjet Printers

Characteristics of Non-impact Printers:

Faster than impact printers.
They are not noisy.
High quality.
Support many fonts and different character size.

Laser Printers

These are non-impact page printers. They use laser lights to produce the dots needed to form the characters to be printed on a page.

Advantages

Very high speed.

Very high quality output.

Gives good graphics quality.

Supports many fonts and different character sizes.

Disadvantage

Expensive.

Cannot be used to produce multiple copies of a document in a single printing.

Inkjet Printers

Inkjet printers are non-impact character printers based on a relatively new technology. They print characters by spraying small drops of ink onto paper. Inkjet printers produce high quality output with presentable features. They make less noise because no hammering is done and these have many styles of printing modes available. Color printing is also possible. Some models of Inkjet printers can produce multiple copies of printing also.

Advantages

High quality printing

More reliable

Disadvantages

Expensive as cost per page is high

Slow as compared to laser printer

Plotters

A plotter is a large printer that generates high-quality documents by moving ink pens over the surface of a page. Plotters are particularly useful to engineers and architects, as they produce high-quality blueprints, maps, and floor plans.

Note: Raster vs Vector Devices- Dot-matrix, inkjet and laser printers are called Raster imaging devices,because they construct the image from a matrix (raster) of dots. Plotters are Vector imaging devices – it constructs the image from a series of line segments produced by dragging a pen over a sheet of paper under computer control. A Vector in computer graphics is a line that is defined by its start and end point.

Plotters can draw complex line art, including text, but do so very slowly because of the mechanical movement of the pens. A printer is aimed primarily at printing text. This makes it fairly easy to control, simply sending the text to the printer is usually enough to generate a page of output. This is not the case of the line art on a plotter, where a number of printer control languages were created to send the more detailed information like "draw a line from here to here". The most popular of these is likely HPGL.

Types of plotters: Drum plotter
Flatbed plotter

Drum plotter:

A type of pen plotter that wraps the paper around a drum with a pin feed attachment. The drum turns to produce one direction of the plot, and the pens move to provide the other. The plotter was the first output device to print graphics and large engineering drawings.

Flatbed plotter:

A graphics plotter that contains a flat surface that the paper is placed on. The size of this surface (bed) determines the maximum size of the drawing.

PART – B: Problem Solving in C

1. Design and develop a flowchart or an algorithm that takes three coefficients (a, b, and c) of a Quadratic equation (ax2+bx+c=0) as input and compute all possible roots. Implement a C program for the developed flowchart/algorithm and execute the same to output the possible roots for a given set of coefficients with appropriate messages.

Variables

a,b,c : Used to store co-efficients of quadratic equation.

d/Δ : Used to store Discriminate.

Realpart : Used to store the real part of the roots.

Imaginarypart: Used to store the imaginary part of the roots.

root1, root2 : Used to store the roots of equation.

Algorithm

Step 1: Accept the co-effiecients a,b,c of the quadratic equation as inputs from the user.

Step 2: Check for the product of co-efficients, if the product is equal to zero then the co-efficients do not form the quadratic equation.

Step 3: Now compute the discriminant Δ=b*b-4*a*c.

Step 4: If the discriminant Δ=0 then the roots are equal i.e root1=root2=-b/(2*a).

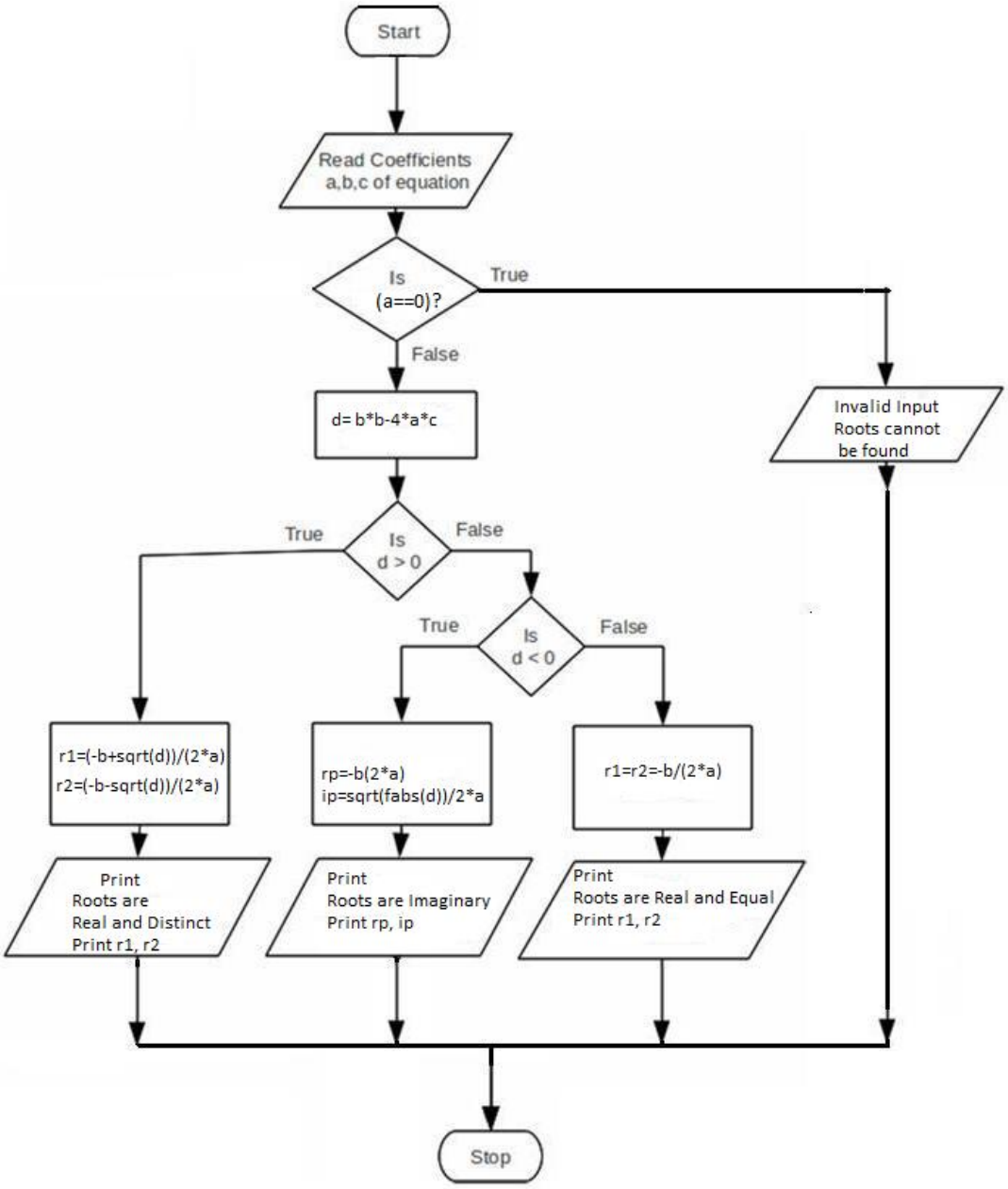
Step 5: If the discriminant Δ>0 then the roots are real ie root1=(-b + sqrt(d)) / (2*a), root2= (-b - sqrt(d)) / (2*a).

Step 6: If the discriminant Δ<0 then the roots are imaginary where Realpart= -b/(2*a) and Imaginarypart= sqrt(fabs(d))/(2*a).

Therefore root1= Realpart + i * Imaginarypart. root2= Realpart – i * Imaginarypart.

Step 7: End.

Flowchart


Start
Read Coefficients a,b,c of equation
Is (a==0)?
True
False
Invalid Input Roots cannot be found
d= b*b-4*a*c
Is d > 0
True
False
Is d < 0
True
False
r1=(-b+sqrt(d))/(2*a)
r2=(-b-sqrt(d))/(2*a)
rp=-b(2*a)
ip=sqrt(fabs(d))/2*a
r1=r2=-b/(2*a)
Print Roots are Real and Distinct Print r1, r2
Print Roots are Imaginary Print rp, ip
Print Roots are Real and Equal Print r1, r2
Stop

```
/*Program to execute quadratic equation and roots*/
#include<stdio.h>
#include<math.h>
void main()
{
 float a,b,c,d,x1,x2,rp,ip;
 printf("Enter the coefficients of a,b and c of quadratic equation\n");
 scanf("%f %f %f",&a,&b,&c);
        if(a==0)
        printf("It is linear equation,roots cannot be found");
        else
        {
                d=b*b-4*a*c;
                if(d==0)
                {
                        printf("Roots are real and equal\n");
                        x1=x2= -b/(2*a);
                        printf("x1=%f and x2=%f\n",x1,x2);
                }
                else if(d>0)
                {
                        printf("Roots are real and distinct\n");
```

```
                x1=(-b+sqrt(d))/2*a;

                x2=(-b-sqrt(d))/2*a;

                printf("x1=%f and x2=%f",x1,x2);

        }

        else

        {

                printf("Roots are imaginary\n");

                rp=-b/2*a;

                ip=(sqrt(fabs(d))/2*a);

                printf("x1=%f+i*%f \n x2=%f-i*%f",rp,ip,rp,ip);

        }

        }

}
```

2. Design and develop an algorithm to find the *reverse* of an integer number NUM and check whether it is PALINDROME or NOT. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: Num: 2014, Reverse: 4102, Not a Palindrome

Variables

num : Used to store the number entered by the user.

temp : Used to temporarily store the accepted number.

rev : Used to store the result after repeated calculation.

rem : Used to store single digits of the number.

Algorithm

Step 1:Start

Step 2: Accept a integer number and store in a variable **num**.

Step 3: Copy temporarily the accepted number in a variable **temp**.

Step 4: Repeat Step 5 until **num != 0**.

Step 5: Compute **rem**=num%10.

Compute rev=rev*10+rem.

Compute **num**=num / 10.

Step 6: If **temp = revnum** then

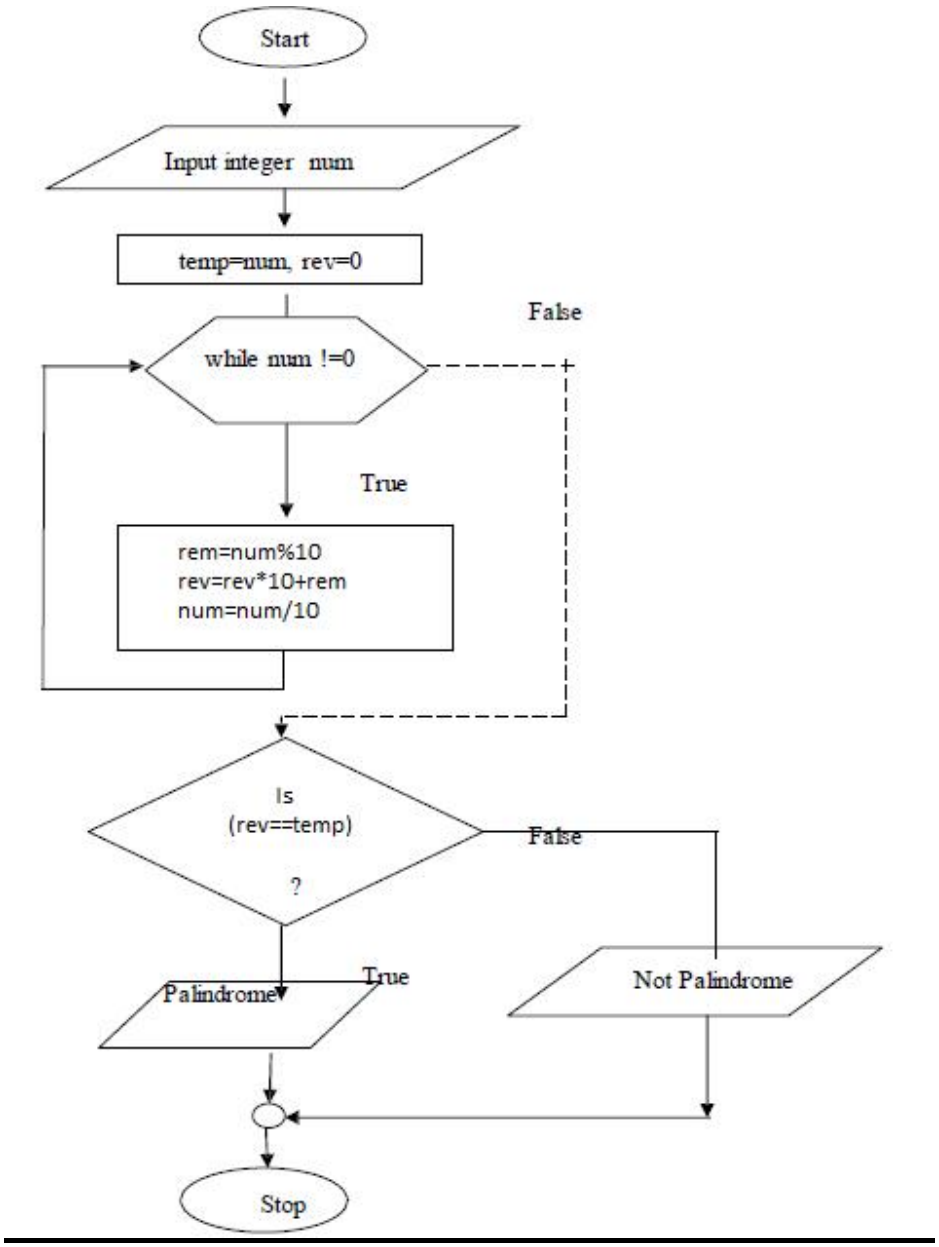
print the accepted number is palindrome and goto

Step 7. else

print the number is not palindrome and goto Step 7

Step 8: End.

FLOWCHART:


Start
Input integer num
temp=num, rev=0
while num !=0
False
True
rem=num%10
rev=rev*10+rem
num=num/10
Is
(rev==temp)
?
False
True
Palindrome
Not Palindrome
Stop

```
/*Program to find given number is palindrome or not*/
#include<stdio.h>
void main()
{
        int num,rev=0,rem,temp;
        printf("Enter a number");
        scanf("%d",&num);
        temp=num;
        while(num!=0)
                {
                 rem=num%10;
                 rev=rev*10+rem;
                 num=num/10;
                }
        printf("Num %d\n",temp);
        printf("Reverse %d\n",rev);
        if(rev==temp)
        printf("Number is palindrome\n");
        else
        printf("Number is not palindrome\n");
}
```

3a. Design and develop a flowchart to find the square root of a given number *N*. Implement a C program for the same and execute for all possible inputs with appropriate messages. Note: Don't use library function *sqrt(n)*.

Variables

N : Used to store the number entered by the user.

sqroot :Used to store square root of number.

temp : Used to temporarily store number.

Algorithm

Step 1: Start

Step 2: Accept a number and store in a variable **N**. Step 3:Compute sqroot=N/2

Step 4: Initialize temp=0

Step 4: Repeat Step 5 until sqroot **!=** temp.

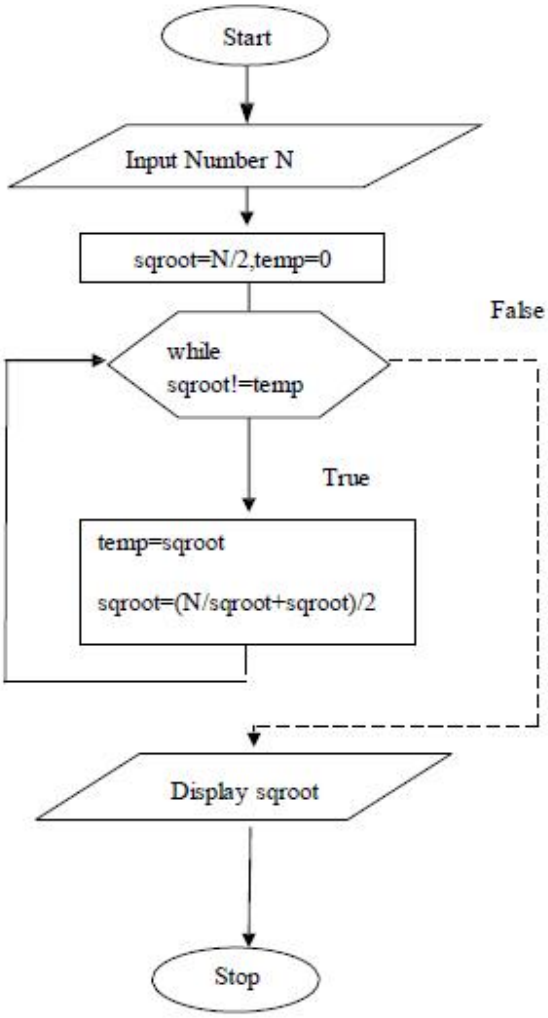
Step 5: Set temp=sqroot

Compute sqroot=(N/sqroot+sqroot)/2

Step 6: print the number N and its square root stored in sqroot.

Step 7: End.

Flowchart


Start
Input Number N
sqroot=N/2,temp=0
while
sqroot!=temp
False
True
temp=sqroot
sqroot=(N/sqroot+sqroot)/2
Display sqroot
Stop

```
/*Program to find square root of a number*/

#include<stdio.h>

void main()

{

        int num;

        float sqrt;

   float temp;

        printf("Enter a number");

        scanf("%d",&num);

        temp=num;


        sqrt=num/2;

        while(sqrt!=temp)

                {

                 temp=sqrt;

                 sqrt=(num/sqrt+sqrt)/2;

                }

        printf("Square root is %f",sqrt);

}
```

3b. Design and develop a C program to read a year as an input and find whether it is leap year or not. Also consider end of the centuries.

Variables

year : Used to store year entered by user.

Algorithm

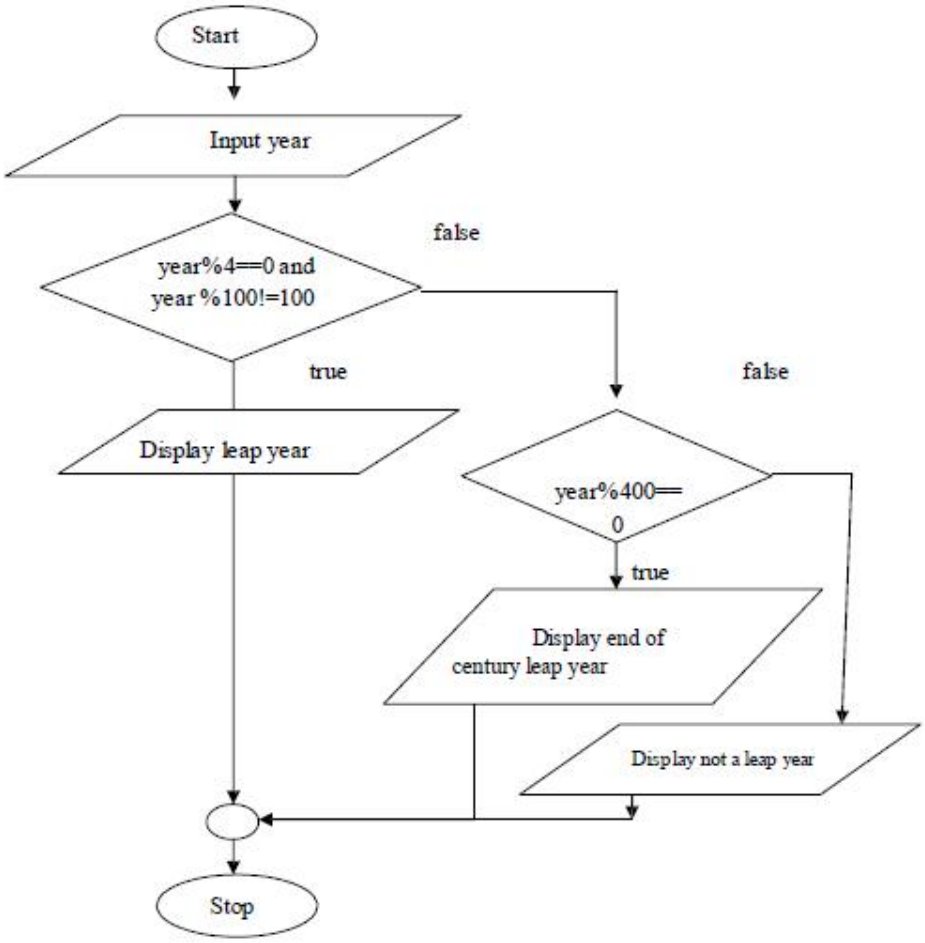
Step 1 : Start

Step 2 : Accept a year from user and store it into variable year.

Step 3 : If year%4 is 0 and year %100 !=0 then display entered year is a leap year. Step 4: If year%400 is 0 then display entered year is end of century leap year.

Step 5: Otherwise display year is not a leap year. Step 6: Stop.

Flowchart



```
/*Program to check leap year or not*/
#include<stdio.h>
void main()
{
        int y;
        printf("Enter a year");
        scanf("%d",&y);
        if((y%4==0)&&(y%100!=0))
        printf("The given year is a leap year");
        else if(y%400==0)
        printf("The given year is a end of century leap year");
        else
        printf("Not a leap year");
}
```

4. Design and develop an algorithm for evaluating the polynomial f(x) = a4x+ a3x+ a2x + a1x + a0, for a given value of x and its coefficients using Horner's method. Implement a C program for the developed algorithm and execute for different sets of values of coefficients and x.

Variables

i : Used as counter in a loop.

coeff[5] : Used to store the co-efficients of the polynomial.

x : Used to store the value of x factor of polynomial.

res : Used to store the result.

Algorithm

Step 1: Start

Step 2: Repeat Setp 3 for i=n to 0.

Step 3: Accept the co-efficients of 4^{th} degree polynomial into an array coeff[i].

Step 4: Accept the value of **x**.

Step 5: Initialize a variable **res=0.**

Step 6: Repeate Step 7 for i=n to 0.

Step 7: **res=res*x+coeff[i].**

Step 8: Print the evaluated result **res**.

Step 9: End.

```
/*Program to evaluate polynomila using Horner's method*/
#include<stdio.h>
void main()
{
        int n,i;
        float coeff[20],x,res=0.0;
        printf("Enter the degree of polynomial");
        scanf("%d",&n);
        printf("\n Enter the n+1 coefficient of polynomial");
        for(i=n;i>=0;i--)
        {
         scanf("%f",&coeff[i]);
        }
        printf("Enter the value of x:");
        scanf("%f",&x);
        for(i=n;i>=0;i--)
        {
         res=res*x+coeff[i];
        }
        printf("\n Evaluated result is= %f",res);
}
```

5. Write C program to compute sin(x)using the Taylor Series approximation given by sin(x)=x-(x/3!)+(x-5)!-(x/7)!+... Compare the result with the built in library function and print both the results.

Variables

i,j : Used as counters in loop.

X : Used to read in degree whose sine value needs to be calculated.

term : Used as part of evaluating the expressions.

val : Used as part of evaluating the expressions.

neg :Used to negate values.

rad :Used to store radians of x

Algorithm for main

Step 1: Read degree of x.

Step 2: Convert degree of x to radians and store it in rad.

Step 3:Set term=1.0,val=rad, neg=1 and i=3.

Step 4:Repeat Step 5till i<=13

Step 5: Compute neg=neg*-1; term=(pow(rad,i)/fact(i))*neg; val=val+term;

Increment i and goto Step 4. Print sin(x) value computed by program.

Step 6: Print sin(x) computed by built in function.

Algorithm for fact(n)

Step 1: Initialize prod=1

Step 2: Repeat step 3 for i=1 to n

Step 3: compute prod=prod*i Step 4: return prod

```
/*Program to implement Taylor's Series*/
#include<stdio.h>
#include<math.h>
int fact(int n);
void main()
{
        int i,neg;
        float x,term,val,rad;
        printf("\n Enter the value of x in degree:");
        scanf("%f",&x);
        rad=(3.142857*x)/180;
        val=rad;
        neg=1;
        for(i=3;i<=13;i=i+2)
        {
                neg=neg*-1;
                term=(pow(rad,i)/fact(i))*neg;
                val=val+term;
                //printf("term %d %f \n",i,term);
        }
                printf("\n sin(%f)computed by prog is %f",x,val);
                printf("\n sin(%f)computed by built in function is %f",x,sin(rad));
```

```
}
int fact(int n)
{
        int fact1=1,i;
        for(i=1;i<=n;i++)
                fact1=fact1*i;
                return fact1;
}
```

6. Develop an algorithm, implement and execute a C program that reads N integer numbers and arrange them in ascending order using Bubble Sort technique.

Variables

A[100] : Used to store array elements.

n : Used to specify the number of elements to Accept.

i,j : Both act as counter variables in loops.

temp : Used to hold temporary values during swapping.

Algorithm

Step 1 : Accept the number n as the size of the array a[].

Step 2 : Accept the elements of array a[].

Step 3 : Print the accepted array elements.

Step 4 : Repeat Step 5 for i=0 to n.

Step 5 : Repeat step 6 for j=0 to n-i-1.

Step 6: if a[j] > a[j+1] then perform step else increment j and compute temp=a[j], a[j]=a[j+1] and a[j+1]=temp.

Step 7 : Print the sorted array a[].

```
/*Program to print given array in ascending order using BST*/
#include<stdio.h>
void main()
{
	int a[10],n,i,j,temp;
	printf("Enter the no of elements");
	scanf("%d",&n);
	printf("Enter the elements of array:\n");
	for(i=0;i<n;i++)
	{
		scanf("%d",&a[i]);
	}
		printf("\n Elements of array before sorting are");
		for(i=0;i<n;i++)
		{
		  printf("\n\t %d",a[i]);
		}
	for(i=0;i<n-1;i++)
	{
		for(j=0;j<n-i-1;j++)
		{
			if(a[j]>a[j+1])
```

```
                    {
                     temp=a[j];
                     a[j]=a[j+1];
                     a[j+1]=temp;
                    }
                }
        }
                printf("\n Elements of array after sorting are");
                for(i=0;i<n;i++)
                {
                 printf("\n\t %d\n",a[i]);
                }
}
```

7. Develop, implement and execute a C program that reads two matrices A (m x n) and B (p x q) and Compute the product A and B. Read matrix A in row major order and matrix B in column major order. Print both the input matrices and resultant matrix with suitable headings and in matrix format. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

Variables

A[ ][ ] : Used to store elements of matrix1.
B[ ][ ] : Used to store elements of matrix 2.
C[ ][ ] : Used to store the result of product of matrices mat1 and mat2.
m : Used to specify the rows of first matrix.
n : Used to specify the columns of first matrix.
p : Used to specify the rows of second matrix.
q : Used to specify the columns of second matrix.
i,j,k : Used as counters in loops.

Algorithm

Step 1 : Accept the number of rows(m,n) and columns(p, q) of A and B.
Step 2 : if (n!=p) then
print the two matrics cannot be multiplied
else
goto Step 3.
Step 3 : Repeat Step 4 for i=0 to m and j=0 to n.
Step 4 : Accept the elements of A[ ][ ].
Step 5 : Repeat Step 6 for i=0 to p and j=0 to q.
Step 6 : Accept the elements of B[ ][ ].
Step 7 : Repeat Step 8 for i=0 to m and j=0 to q.
Step 8 : Assign C[i][j]=0, now repeat Step 9 for k=0 to n.
Step 9 : C[i][j]=C[i][j]+(A[i][k]*B[k][j]) goto Step 10.
Step 10: goto Step 7.
Step 11: Repeat Step 12 for i=0 to m and j=0 to n.
Step 12: Print the elements of matrix A[ ][ ].

Step 13 Repeat Step 14 for i=0 to p and j=0 to q.

Step 14: Print the elements of matrix B[ ][ ].

Step 15: Repeat step 16 for i=0 to m and j=0 to q.

Step 16: Print the elements of C [ ][ ].

Step 17: End.

/*Program to compute the product of two matrices*/

```
#include<stdio.h>
void main()
{
        int a[10][10],b[10][10],c[10][10],m,n,k,p,q,i,j;
        printf("\n Enter the number of rows and number of columns for matrix A");
        scanf("%d %d",&m,&n);
        printf("\n Enter the number of rows and columns for matrix B");
        scanf("%d %d",&p,&q);
        printf("\n\t Enter the elements of matrix A");
        for(i=0;i<m;i++)
        {
                for(j=0;j<n;j++)
                {
                        scanf("%d",&a[i][j]);
                }
        }

        printf("\n\t Enter the elements of matrix B");
        for(i=0;i<q;i++)
        {
                for(j=0;j<p;j++)
                {
                        scanf("%d",&b[j][i]);
                }
        }
        printf("\n\t Matrix A \n");
```

```
for(i=0;i<m;i++)
{
        for(j=0;j<n;j++)
        {
                printf("\t %d",a[i][j]);
        }
        printf("\n");
}
printf("\n\t Matrix B \n");
for(i=0;i<p;i++)
{
        for(j=0;j<q;j++)
        {
                printf("\t%d",b[i][j]);
        }
        printf("\n");
}
if(n==p)
{
        for(i=0;i<m;i++)
        {
                for(j=0;j<q;j++)
                {
                 c[i][j]=0;
                 for(k=0;k<n;k++)
                 {
                        c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
                 }
                }
        }
printf("\n\t Elements of matrix C are\n");
for(i=0;i<m;i++)
```

```
        {
                for(j=0;j<q;j++)
                {
                        printf("\t %d",c[i][j]);
                }
                printf("\n");
        }
        }
else
        printf("\n Matrix multiplication is not possible");
}
```

8. Develop, implement and execute a C program to search a name in a list of names using binary searching techniques.

Variables

name[ ][ ] : Used to store elements of sorted array.

key[] : Used to store elements of sorted array.

strcmp : used to compare two strings.

Algorithm

Step 1:Print the sorted array name[][],key[]

Step 2:Accept number key to be searched.

Step 3:Assign low=0 and high=n-1.

Step 4:Repeat step5 until (low<=high).

Step 5:Compute mid=(low+high)/2.

```
if(strcmp(key,name[mid]) == 0)
goto step6
else if (strcmp(key,name[mid]) > 0)
compute low=mid+1;
            high=high.
 else
 compute low=low;
             high=mid-1.
```

Step 6:print key element is found at position mid+1.

```
 else
 print key element is not found.
```

Step 7:End.

```
/*PROGRAM USING BINARY SEARCH*/
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
int n,i,low,high,mid;
char key[10],names[10][10];
printf("Enter how many names\n");
scanf("%d",&n);
printf("Enter names in alphabetical order\n");
for(i=0;i<n;i++)
scanf("%s",names[i]);
printf("Enter the name to be searched\n");
scanf("%s",key);
low=0;
high=n-1;
while(low<=high)
{
mid=(low+high)/2;
if(strcmp(key,names[mid])==0)
{
printf("name is found at position %d\n",mid+1);
exit(0);
}
else if(strcmp(key,names[mid])>0)
low=mid+1;
else
high=mid-1;
}
printf("Name is not found\n");
}
```

9. Write and execute a C program that

i. Implements string copy operation STRCOPY(str1,str2) that copies a string str1 to another string str2 without using library function.

Variables

str1, str2 : Used to store strings

i,j :counters

Algorithm

Step 1 : Start

Step 2 : Read the string from the user.

Step 3 :set i=0 and j=0

Step 4 : Repeat step 4 until end of character str1 i.e. (str1[i]!='\0')

Step 5 : Compute str2[j]=str1[i] Increment i and j

Step 6 : Print string in str2

Step 7 : stop

```
/*Program to copy one string to another*/
#include<stdio.h>
void main()
{
        char str1[100],str2[100];
        int i=0,j=0;
        printf("Enter the string");
        gets(str1);
        printf("The entered string is %s\n",str1);
        while(str1[i]!='\0')
        {
                str2[j]=str1[i];
        i++;
        j++;
        }
        str2[i]='\0';
        printf("The copied string is %s\n",str2);
}
```

ii. Reads a sentence and prints frequency of each of the vowels and total count of consonants.

Variables

str1 : Used to store the entered sentence

c :Used to temporarily store a character

i :counter,counta, counte,counti,counto,countu,countc: counters.

Algorithm

Step 1 : Start

Step 2 :Read the string from the user.

Step 3 : Repeat step 4 until end of string

Step 4 : Read each character from the string Match the character against switch case Increment the respective counter.

Step 5 :Print frequency of vowels

Step 6 :Print number of consonants

Step 7 : Stop

```
/*Program to calculate frequency of vowels and consonants*/
#include<stdio.h>
void main()
{
        char str1[100];
        int i=0,ca=0,ce=0,ci=0,co=0,cu=0,cc=0;
        printf("Enter the string");
        gets(str1);
        printf("The entered string is %s\n",str1);
        while(str1[i]!='\0')
        {
                switch(str1[i])
                {
                        case'a':
                        case'A':ca++;
                                break;
                        case'e':
                        case'E':ce++;
```

```
                        break;
                case'i':
                case'I':ci++;
                        break;
                case'o':
                case'O':co++;
                        break;
                case'u':
                case'U':cu++;
                        break;
                default:if(str1[i]!=' ')
                        cc++;
            }
                i++;
        }
        printf("\n Frequency of vowels\n");
        printf("\n a= %d",ca);
        printf("\n e= %d",ce);
        printf("\n i= %d",ci);
        printf("\n o= %d",co);
        printf("\n u= %d",cu);
        printf("\n Number of consonants= %d",cc);
}
```

10. a Design and develop a C function RightShift(x,n) that takes two integets x and n as input and returns value of the integer x rotated to the right by n positions. Assume the integers are unsigned. Write a C program that invokes this function with different values for x and n. Tabulate the results with suitable headings.

Variables

numx : Used to store the number.

rotn : Specifies the number of places the number is going to get rotated.

result : Used to store the result i.e the number after performing right rotation.

i : Used as counter in a loop.

Algorithm for main

Step 1: Start.

Step 2: Accept the number **num** for performing right rotation.

Step 3: Accept the number of rotations the number num should be shifted thru.

Step 4: Call function rightrot(num, rotations) and store the returned value into result

Step 5: Print the value stored in result that will be the right rotated number. Step 6: Stop.

Algorithm for rightrot(num,rotations)

Step 1: initialize i=0

Step2 : repeat step 3 for i=0 to rotations

Step 3: if(num%2= 0) num=num>>1 else num=num>>1

num =num+32768

Step 4: return num

```
#include<stdio.h>
unsigned int right(unsigned int num,unsigned int rot);
void main()
{
unsigned int num[10],result,n,i;
int rot[10];
printf("\n\t enter the number of computation");
scanf("%u",&n);

printf("\n\t enter the number ");
for(i=0;i<n;i++)
scanf("%u",&num[i]);

printf("\n\t enter the number of rotation");
for(i=0;i<n;i++)
scanf("%d",&rot[i]);
printf("\n\t n    rot    result");
for(i=0;i<n;i++)
{
result=right(num[i],rot[i]);
printf("\n\t %d\t%d=\t%u\n",num[i],rot[i],result);
}
}
unsigned int right(unsigned int num,unsigned int rot)
{
int i;
for(i=1;i<=rot;i++)
{
if(num%2==0)
{
num=num>>1;
}
else
{
num=num>>1;
num=num+32768;
}
}
return (num);
}
```

b. Design and develop a C function isprime(num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given range.

Variables

m,n : Used to store the range.

result : Used to store the return value from the function isPrime(x).

flag : Acts as status value in function isPrime(x), the value set for flag is returned to the main() function.

Algorithm for main() function

Step 1: Start.

Step 2: Read a number .

Step 3: Call function **isPrime(n)** return value to be stored in **result**.

Step 4: if result=1 then

print the number is prime and goto Step 9 else

print the number is not prime and goto Step 9.

Step 5: End of main().

Algorithm for isPrime(n) function

Step 1: Receive n as argument from main function.

Step 2: Set flag=1

Step 3: if n= 1 set flag=0 otherwise go to step 4

Step 4: Repeat Step 5 for i = 2 to count <n,

if (n % count == 0) then Set flag=0 and break.

Step 5: Return flag value to the main() function.

Step 6: End of isPrime(x) function.

```
#include<stdio.h>
int isprime(int num);
void main()
{
   int i,m,n;
   printf("Enter the range of numbers to print prime numbers");
   scanf("%d %d",&m,&n);
   for (i=m;i<n;i++)
   {
      if(isprime(i))
      {
         printf("%d\t",i);
      }
   }
}
int isprime(int num)
{
   int flag=1,i;
   if(num==1)
      flag=0;
   else
   {
      for(i=2;i<num;i++)
      {
         if(num%i==0)
         {
               flag=0;
               break;
         }
      }
   }return flag;
}
```

11. Draw the flowchart and write a recursive C function to find the factorial of a number, n!, defined by fact(n)=1, if n=0. Otherwise fact(n)=n*fact(n-1). Using this function, write a C program to compute the binomial coefficient $^{n}C_{r}$. Tabulate the results for different values of n and r with suitable messages.

Variables

n :Array to hold different values for n.

r :Array to hold different values for r.

result: Used to store nCr value.

i :Counter.

num :Used to store number of computations.

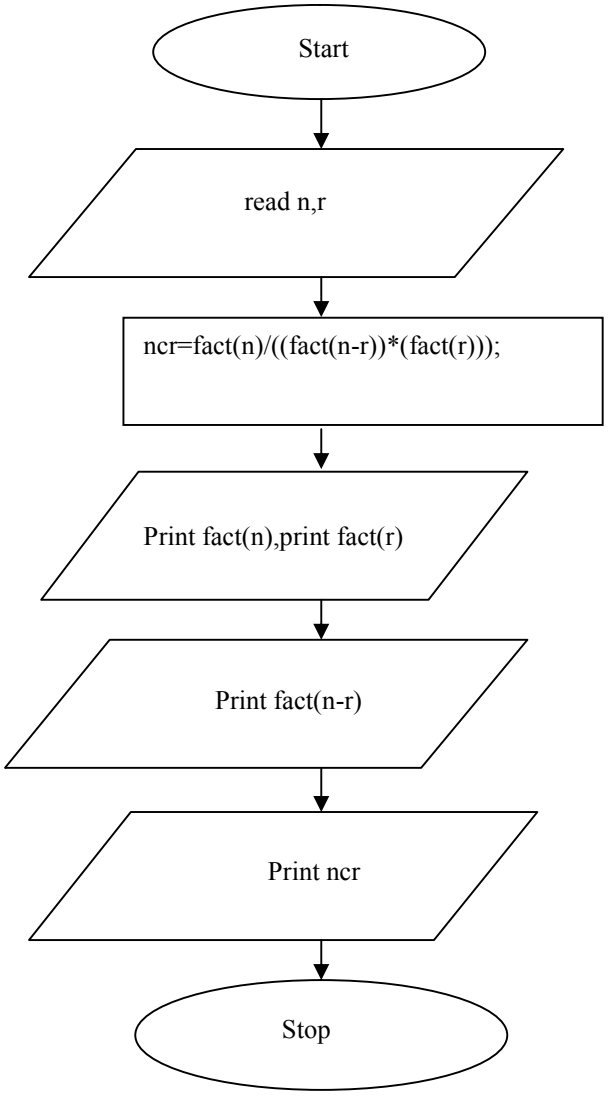
Algorithm

1. Start
2. Read n and r
3. Call function fact(n),fact(n-r) and fact(r)
4. Calculate ncr=fact(n)/((fact(n-r))*(fact(r)));
5. Print ncr
6. Stop.

Algorithm of factorial function

```
if n==0
 return1
else
return (n*fact(n-1))
```

Flowchart


Start
read n,r
ncr=fact(n)/((fact(n-r))*(fact(r)));
Print fact(n),print fact(r)
Print fact(n-r)
Print ncr
Stop

```
/*Program to compute binomial coefficient*/
#include<stdio.h>
int fact(int n);
main()
{
        int r[10],n[10],result,i,num;
        printf("\n Computing nCr\n");
        printf("\n Enter the number of computations:");
        scanf("%d",&num);
        printf("\n Enter %d different values for n:",num);
        for(i=0;i<num;i++)
        scanf("%d",&n[i]);
        printf("\n Enter %d different values for r:",num);
        for(i=0;i<num;i++)
        scanf("%d",&r[i]);
        printf("\n\t n\t r\t nCr");
        for(i=0;i<num;i++)
        {
        if(n[i]==0)
        {
        printf("\n\t %d \t %d",n[i],r[i]);
        printf("\t Please enter valid value for n");
```

```
        }
        else if(n[i]<r[i])
        {
        printf("\n\t %d \t %d",n[i],r[i]);
        printf("\t Please enter valid values for n and r");
        }
        else
        {
        result=fact(n[i])/(fact(r[i])*fact(n[i]-r[i]));
        printf("\n\t %d \t %d \t %d",n[i],r[i],result);
        }
        }
}
        int fact(int n)
        {
        if(n==0)
        return 1;
        else
        return(n*fact(n-1));
        }
```

12. Given two university information files "studentname.txt" and "usn.txt" that contains student name and usn respectively. Write a C program to create a new file called "output.txt" and copy the content of files "studentname.txt" and "usn.txt" into output file in the sequence shown below. Display the output file "output.txt" onto the screen.

Student Name	**USN**
Name1	**USN1**
Name 2	**USN2**
….	**…**

Variables:

fp1,fp2,fp3 : File pointers to student.txt,usn.txt and output.txt respectively name[ ] ,

usn :Used to store name and usn from files respectively.

Definition: File is a storage unit on the secondary memory. Any file which present in the hard disk would contain file name and its extensions.

Ex: .doc,.docx,.jpeg,.mov,.mp3 etc.

Procedure

1. create a file by name "studentname.txt" in read mode
2. write students names into created file
3. create a another file by name "USN.txt" in read mode.
4. Write few USNs into created file.
5. Check whether files are exits by file function given as if(fp1==NULL)?
6. Read content of the files by proper file commands
7. Copy the read contents into another file name "OUTPUT.txt", which was created in write mode.
8. Print the copied content from "OUTPUT.txt" n display on Monitor.

```
/*Program to print student name and usn*/
#include<stdio.h>
#include<stdlib.h>
void main()
{
        FILE *fp1,*fp2,*fp3;
        char name[20];
        int USN;
        fp1=fopen("student.txt","r");
        if(fp1==NULL)
        {
        printf("File not found\n");
        exit(0);
        }
        fp2=fopen("USN.txt","r");
        if(fp2==NULL)
        {
        printf("File 2 not found\n");
        exit(0);
        }
        fp3=fopen("Output.txt","w");
        if(fp3==NULL)
        {
        printf("File 3 not found\n");
        exit(0);
        }
        while(1)
        {
        if(fscanf(fp1,"%s",name)>0)
        if(fscanf(fp2,"%d",&USN)>0)
        fprintf(fp3,"%s\t \t %d \n",name,USN);
        else
```

```
break;
        else
        break;
        }
        fclose(fp1);
        fclose(fp2);
        fclose(fp3);
        fp3=fopen("Output.txt","r");
        if(fp3==NULL)
        {
        printf("File 3 not found\n");
        exit(0);
        }
        printf("Student name\t USN \n");
        while(fscanf(fp3,"%s %d",name,&USN)>0)
        printf("%s \t \t %d \n",name,USN);
        fclose(fp3);
}
```

13. Write a C program to maintain a record of ‘‘n’’ student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Each field is of an appropriate data type. Print the marks of the student given student name as input.

Variables

s :array of structure student.
roll :Used to store student roll number.
name :Used to store name of student.
marks :Used to store marks of student.
grade :Used to store grade of student.
found :Used in searching student information.
n :Used to store number of students.
i :counter.
choice :Used to store user choice.

Procedure

1. Create a structure, the members of the structure are
   Rollno[];
   Name[];
   Marks;
   Grade[];
2. Read the number of students for which record should be maintained.
3. Enter the information of each student’s, repeat the same step from i=0 to n;
4. Display the students information on screen from i=0 to n;
5. Enter the key student name for which information should be displayed.
6. Check whether student details are available by condition
   if(strcmp(key,s[i].name))==0)
7. Print “No record found” in case student details are not available.

```
/*Program to maintain a record*/
#include<stdio.h>
struct student
{
        char roll[15];
        char name[20];
        float marks;
        char grade[3];
};
int main()
{
        struct student s[100];
        int i,n,found=0,choice=1;
        char key[50];
        printf("\nEnter number of students:");
        scanf("%d",&n);
        printf("Enter information of %d students:\n",n);
        for(i=0;i<n;i++)
        {
           printf("Enter the details of student %d\n ",i+1);
        printf("Enter the roll number:");
        scanf("%s",s[i].roll);
```

```
printf("Enter name:");
scanf("%s",s[i].name);
printf("Enter marks:");
scanf("%f",&s[i].marks);
printf("Enter grade:");
scanf("%s",s[i].grade);
printf("\n");
}
printf("Displaying information of students:\n\n");
printf("Roll no\t Name\t Marks\t Grade\n");
for(i=0;i<n;i++)
{
printf("%s\t",s[i].roll);
printf("%s\t",s[i].name);
printf("%4.2f\t",s[i].marks);
printf("%s\n",s[i].grade);
}
while(choice)
{
printf("\nEnter the name of student to display marks:");
scanf("%s",key);
```

```
for(i=0;i<n;i++)
        {
        if(strcmp(key,s[i].name)==0)
        {
        found=1;
        printf("\n Marks of %s student:%4.2f",s[i].name,s[i].marks);
        }
        }
        if(found==0)
        {
        printf("\n No record found of %s student",s[i].name);
        }
        found=0;
        printf("\n\n Press 0 to exit and 1 to continue:");
        scanf("%d",&choice);
        }
        return 0;
}
```

14. Write a C program using pointers to compute the sum, mean, and standard deviation of all elements stored in an array of n real numbers.

Variables

n : Used to store number of elements.

a :Array to store n elements.

i :counter.

sum :Used to store sum of n elements.

mean :Used to store the mean.

sd :Used to store standard deviation.

ptr :pointer

Definition: A pointer is a variable that holds a address of another variable.

Algorithm

Step1: read the size of the array

Step2: read the elements of the array.

Step3: repeat the step 2 from i=0 to n;

Step4: assign the address of the array to pointer by

ptr=a;

step5: compute sum by

sum=sum+*ptr;

step6: repeat step 5 from i=0 to n;

step7: compute mean by mean=sum/n;

step8: compute variance by

variance= variance+ pow((*ptr-mean),2);

step9: repeat the step from i=0 to n

step10: compute standard deviation by sd=sqrt(variance);

step11: end;

```
#include<stdio.h>
#include<math.h>
void main()
{
int i,n;
float a[100],sum=0,variance=0,mean,sd;
float *ptr;
printf("\nEnter no. of elements: ");
scanf("%d",&n);
printf("\nEnter %d elements",n);
for(i=0;i<n;i++)
scanf("%f",&a[i]);
ptr=a; //ptr=&a[0]
for(i=0;i<n;i++)
{
sum=sum+ *ptr;
ptr++;
}
mean=sum/n;
ptr=a;
for(i=0;i<n;i++)
{
variance=variance+pow((*ptr-mean),2);
ptr++;
}
variance=variance/n;
sd=sqrt(variance);
printf("\nSum=%3f \nMean=%3f \nStandard Deviation=%3f \n",sum,mean,sd);
}
```

VIVA QUESTIONS

1) What is an algorithm?
2) What is high level language?
3) What is compiler?
4) What are tokens?
5) What are identifiers?
6) What are keywords? How many keywords is their in C programming language?
7) What is a variable?
8) What are the rules to be followed while declaring a variable?
9) What is a constant?
10) What is a datatype? What are the different dataypes?
11) What are escape sequence characters?
12) List the size and range of basic datatypes.
13) What is the difference between a character and string?
14) What is implicit type conversion and explicit type conversion (type casting)?
15) What is precedence of an operator means?
16) What is the difference between printf() and puts() functions.
17) What is function? What are the advantages of functions?
18) What are the different types of functions?
19) What is a library function?
20) What is calling function and called function?
21) What is the meaning of actual parameter and formal parameter?
22) What is the purpose of switch statement? Explain with syntax.
23) What is loop? List the differences between pre-test and post-test loop.

24) What is the meaning of event controlled loop and counter controlled loop?

25) What are the advantages of loops?

26) What is control statement? What are the various types of control statements available in C language?

27) Explain for loop with syntax.

28) What is the difference between while and do-while loop?

29) What are unconditional control statements?

30) What is the use of break statement?

31) What is an array? What is the difference between an ordinary variable and an array variable?

32) What are the differences between recursion and iteration?

33) What is a pointer?

34) What is a NULL pointer?

35) What is a Structure? What are the differences between structures and arrays?

36) What is memory leak? Why it should be avoided.

37) Which header file should be included to use functions like malloc() and calloc()?

38) List string.h Library functions in C.