



KENDRIYA VIDYALAYA SANGATHAN AHMEDABAD REGION

**STUDY MATERIAL
CLASS: XI**

SUBJECT: INFORMATICS PRACTICES (065)

SESSION: 2022-23



CHIEF PATRON



SMT SHRUTI BHARGAV
(Deputy Commissioner, KVS RO Ahmedabad)

PATRON



SMT VINITA SHARMA
(Assistant Commissioner KVS RO Ahmedabad)

MENTOR



SMT SUNITA KUMARI
(Principal. KV No.2 EME, Baroda)

CONTENT CREATOR TEAM



RAKSHA PARMAR, PGT (CS)
KV SABARMATI



**ASHISH KUMAR
CHAURDIA, PGT (CS)**
KV NO. 2 EME, BARODA



**MAYURI PATEL, PGT
(CS)**
KV SILVASSA



MANGI LAL KARWA, PGT (CS)
KV NO. 1 SURAT



**ABHISHEK ARYA, PGT
(CS) KV NO.3 AFS
MAKARPURA, VADODARA**



**KAMLESH AMIN, PGT
(CS)**
KV NO. 3 SU JAMNAGAR

INDEX

UNIT NO.	UNIT NAME	PAGE NO.	
		FROM	TO
1.	<i>Introduction to Computer System</i>	5	8
2.	<i>Introduction to Python</i>	9	41
3.	<i>Data Handling</i>	42	57
4.	<i>Database Concepts and the SQL</i>	58	114
5.	<i>Introduction to Emerging Trends</i>	115	130

KENDRIYA VIDYALAYA SANGATHAN, AHMEDABAD REGION
SPLIT-UP OF SYLLABUS

SUB: INFORMATICS PRACTICE (065)

CLASS - XI (NEW SYLLABUS)

SESSION 2022-23

DISTRIBUTION OF MARKS

UNIT	UNIT NAME	THEORY MARKS	Periods	
			THEORY	PRACTICAL
1	Introduction to computer system	10	10	-
2	Introduction to Python	25	35	28
3	Database concepts and the Structured Query Language	30	23	17
4	Introduction to Emerging Trends	5	7	-
	TOTAL	70	75	45

MONTH- WISE DISTRIBUTION

Month	Topics to be covered	Th.	Pr.
JULY	Unit 1: Introduction to Computer System Introduction to computer and computing: evolution of computing devices, components of a computer System and their interconnections, Input/output devices. Computer Memory: Units of memory, types of memory – primary and secondary, data deletion, its recovery and related security concerns. Software: purpose and types – system and application software, generic and specific purpose software.	15	15
	Unit 2: Introduction to Python Basics of Python programming, Python interpreter - interactive and script mode, the structure of a program, indentation, identifiers, keywords, constants, variables, types of operators, precedence of operators, data types, mutable and immutable data types.		

AUGUST	<p>Statements, expressions, evaluation and comments, input and output statements, data type conversion, debugging.</p> <p>Control Statements: if-else, for loop.</p> <p>Lists: list operations - creating, initializing, traversing and manipulating lists, list methods and built-in functions.</p>	15	10
SEPTEMBER	<p>Dictionary: concept of key-value pair, creating, initializing, traversing, updating and deleting elements, dictionary methods and built-in functions.</p> <p>Unit 3: Database concepts and the Structured Query Language</p> <p>Database Concepts: Introduction to database concepts and its need, Database Management System.</p> <p>Relational data model: Concept of domain, tuple, relation, candidate key, primary key, alternate key.</p>	18	8
OCTOMBER	Advantages of using Structured Query Language, Data Definition Language, Data Query Language and Data Manipulation Language.	5	-
NOVEMBER	HALF YEARLY EXAMINATION	5	5
	<p>Introduction to MySQL, creating a database using MySQL, Data Types</p> <p>Data Definition: CREATE TABLE</p>		
DECEMBER	<p>Data Query: SELECT, FROM, WHERE.</p> <p>Data Manipulation: INSERT</p> <p>Unit 4: Introduction to the Emerging Trends</p> <p>Artificial Intelligence, Machine Learning, Natural Language Processing, Immersive experience (AR, VR), Robotics, Big data and its characteristics, Internet of Things (IoT)</p>	10	7
JANUARY	<p>Sensors, Smart cities, Cloud Computing and Cloud Services (SaaS, IaaS, PaaS); Grid Computing, Block chain technology.</p> <p>• Revision, Project work</p>	7	-
FEBRUARY	• Revision, Project work, Session Ending Practical Examination		

PRACTICAL WORK

S. No.	UNIT NAME	MARKS
1	Lab Test (18 marks)	
	Python program (60% logic + 20% documentation + 20% code quality)	11
	Creating database using MySQL and performing Queries	7
2	Report File + Viva (12 marks)	
	Report file: Minimum 14 Python programs and 14 SQL queries	7
	Viva voce (based on the report file)	5

Suggested Practical List Python Programming

Programming in Python

1. To find average and grade for given marks.
2. To find sale price of an item with given cost and discount (%).
3. To calculate perimeter/circumference and area of shapes such as triangle, rectangle, square and circle.
4. To calculate Simple and Compound interest.
5. To calculate profit-loss for given Cost and Sell Price.
6. To calculate EMI for Amount, Period and Interest.
7. To calculate tax - GST / Income Tax.
8. To find the largest and smallest numbers in a list.
9. To find the third largest/smallest number in a list.
10. To find the sum of squares of the first 100 natural numbers.
11. To print the first 'n' multiples of given number.
12. To count the number of vowels in user entered string.
13. To print the words starting with a alphabet in a user entered string.
14. To print number of occurrences of a given alphabet in each string.
15. Create a dictionary to store names of states and their capitals.
16. Create a dictionary of students to store names and marks obtained in 5 subjects.
17. To print the highest and lowest values in the dictionary.

Data Management: SQL Commands

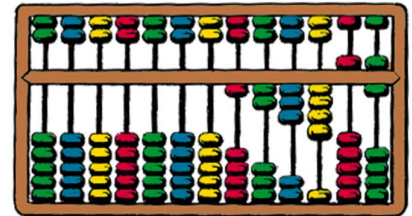
18. To create a database.
19. To create student table with the student id, class, section, gender, name, dob and marks as attributes where the student id is the primary key.
20. To insert the details of at least 10 students in the above table.
21. To display the entire content of table.
22. To display Rno, Name and Marks of those students who are scoring marks more than 50.
23. To find the average of marks from the student table.
24. To find the number of students, who are from section 'A'.
25. To display the information all the students, whose name starts with 'AN'.
(Examples: ANAND, ANGAD,..)
26. To display Rno, Name, DOB of those students who are born between '2005-01-01' and '2005-12-31'.
27. To display Rno, Name, DOB, Marks, Email of those male students in ascending order of their names.
28. To display Rno, Gender, Name, DOB, Marks, Email in descending order of their marks.
29. To display the unique section available in the table.

Introduction to computer system

A computer is an electronic device which consists of hardware and software. Hardware components describe the physical parts of the computer and software controls hardware and run the operating system, different programs & applications like Microsoft Windows, Microsoft word, Microsoft Excel etc.

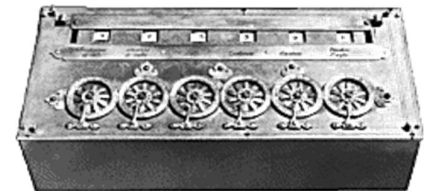
Review of brief history of Computer

1. The history of computers starts out about 2000 years ago in Babylonia (Mesopotamia), at the birth of the **abacus**, a wooden rack holding two horizontal wires with beads strung on them.

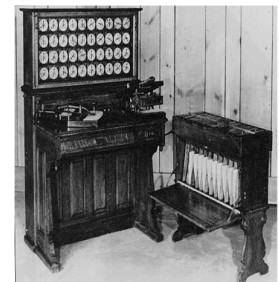


2. Blaise Pascal has credited the **first digital computer** in 1642. The basic principle of his calculator is still used today in water meters and modern-day odometers.

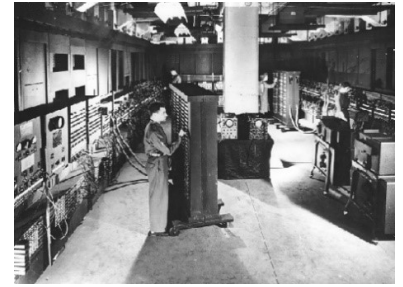
3. **Punched cards**, which were first successfully used with computers in 1890 by Herman Hollerith and James Powers. They developed devices that could read the information that had been punched into the cards automatically, without human help.



4. **Punch-card using computers** created by International Business Machines (IBM), Remington, Burroughs. These computers used electromechanical devices in which electrical power provided mechanical motion -- like turning the wheels of an adding machine. Such systems included features to:
 - feed in a specified number of cards automatically
 - add, multiply, and sort



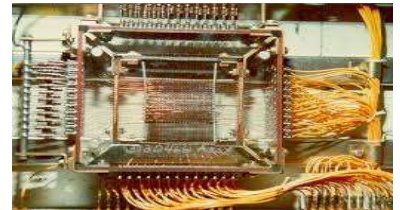
5. In 1942, John P. Eckert, John W. Mauchly, and their associates at the Moore school of Electrical Engineering of University of Pennsylvania decided to build a high - speed electronic computer to do the job. This machine became known as **ENIAC** (Electrical Numerical Integrator And Calculator) and it occupied the entire thirty by fifty feet room.



6. ENIAC was therefore about 1,000 times faster than the previous generation of relay computers. ENIAC used **18,000 vacuum tubes** about 1,800 square feet of floor space, and consumed about 180,000 watts of electrical power.



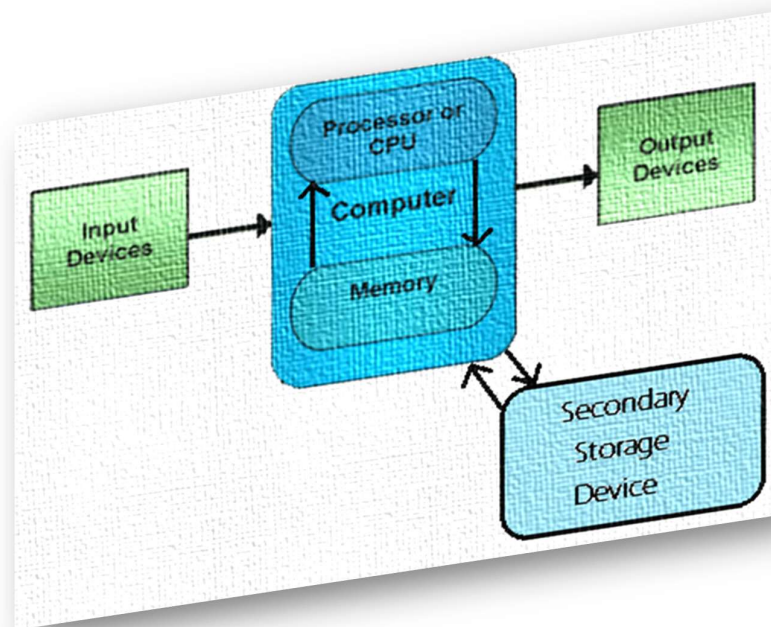
7. In 1950 **magnetic core memory** and the **Transistor** - Circuit Element of digital computers. RAM capacities increased from 8,000 to 64,000 words in commercially available machines by the 1960s, with access times of 2 to 3 MS (Milliseconds).



8. In the 1980's manufacturing of **semiconductor chips** was done, The Intel and Motorola Corporations were very competitive into the 1980s, although Japanese firms were making strong economic advances, especially in the area of memory chips. By the late 1980s, some personal computers were run by microprocessors that, handling 32 bits of data at a time.

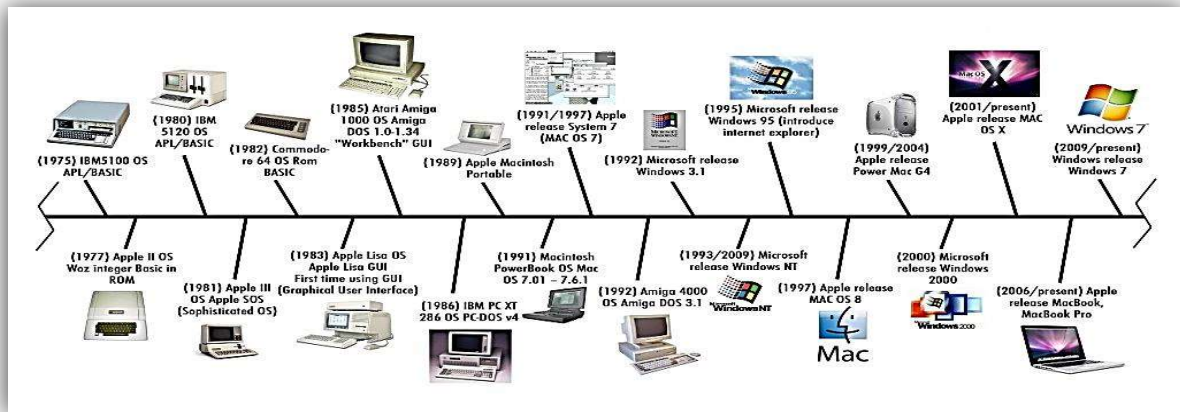


Basic components of Computer:



INPUT UNIT

The devices which are used to input the data and programs in the computer are known as "Input Devices". Input unit accepts instructions and data from the user and converts these instructions and data in computer acceptable format which are sent to computer system for processing.



Keyboard: Keyboard is most common input device. The data and instructions are input by typing on the keyboard. The message typed on the keyboard reaches the memory unit of a computer. It is connected to a computer via a cable. Apart from alphabet and numeral keys, it has other function keys for performing different functions.



Mouse: It is a pointing device. The mouse is rolled over the mouse pad, which in turn controls the movement of the cursor in the screen. You can click, double click or drag the mouse. Its sensor notifies the speed of its movements to the computer, which in turn moves the cursor/pointer on the screen.



Scanner: Scanners are used to enter information directly into the computer memory. This device works like a Xerox machine. The scanner converts any type of printed or written information including photographs into digital pulses, which can be manipulated by the computer.



Track Ball: Track ball is similar to the upside-down design of the mouse. The user moves the ball directly, while the device itself remains stationary. The user spins the ball in various directions to effect the screen movements.



Light Pen: This is an input device which is used to draw lines or figures on a computer screen. It is touched to the CRT screen where it can detect raster on the screen as it passes.



Optical Character Reader (OCR): It is a device which detects alpha numeric characters printed or written on a paper. The text which is to be scanned is illuminated by a low frequency light source. The light is absorbed by the dark areas but reflected from the bright areas. The reflected light is received by the photocells.



Bar Code Reader: This device reads bar codes and converts them into electric pulses to be processed by a computer. A bar code is nothing but data coded in form of light and dark bars.



Voice Input Systems: It converts spoken words to machine language form. A microphone is used to convert human speech into electric signals. The signal pattern is then transmitted to a computer when it is compared to a dictionary of patterns that have been previously placed in a storage unit of computer. When a close match is found, the word is recognized.



Digital Camera: It converts graphics directly into digital form. An electronic chip is used in camera, when light falls, on the chip through the lens, it converts light waves into electrical waves.



OUTPUT UNIT

Output Device produces the final results of computer into human understandable form. Output unit accepts the results produced by the computer which are in coded form and it converts these coded results to human readable form.

Monitor: The monitor looks like a television screen. It is also called Visual Display Unit (VDU) and it is used to display information from the computer. There are coloured as well as black and white monitors. The monitor displays text and graphics. Based on the technology used, monitor is classified into two types. They are Cathode Ray Tube (CRT) monitor and Liquid Crystal Display (LCD) monitor.



Printer: A printer is used for transferring data from the computer to the paper. There are colour printers as well as black and white printers. The different types of printers are Dot Matrix Printers, Inkjet Printers and Laser Printers.



Dot matrix Printer



Inkjet Printer



Laser Printer

Types of Printers

Dot Matrix Printer	Inkjet Printer	Laser Printer
Prints characters in the form of dots.	Prints fully formed characters.	Prints fully formed characters.
Speed: 200-540 characters per second.	Speed: 4-8 pages per minute.	Speed: 4-20 pages per minute.
Not very expensive.	Moderately priced.	Expensive.

Touch Screen: A touch screen is a display screen that is sensitive to human touch. It allows the user to interact with the computer by touching pictures or words displayed on the screen. Touch screens are used to display the output information on the screen as well as to give input by touching the words or pictures on the screen. Thus it is an input/output device. Touch screens are normally used when information has to be accessed with minimum effort.



Plotter: Plotter is very large in size and is used to produce graphical output on papers. It uses single or multi color pens to draw pictures as blue print, posters etc.



STORAGE UNIT

The data and instructions that are entered into the computer system through input units have to be stored inside the computer before the actual processing starts. Similarly, the results produced by the computer after processing must also be kept somewhere inside the computer system before being passed on to the output units. The **Storage Unit** or the **primary / main storage** of a computer system is designed to do all these things. It provides space for storing data and instructions, intermediate results and for the final results.



Central Processing Unit (CPU)

The main unit inside the computer is the **CPU**. This unit is responsible for all events inside the computer. It consists of **Control Unit** and **Arithmetic and Logic unit**. The CPU is the brain of any computer system. In a human body, all major decisions are taken by the brain and the other parts of the body function as directed by the brain. Similarly, in a computer system, all major calculations and comparisons are made inside the CPU and it activates and controls the operations of other units like Hard disk, Printer etc. of a computer system.



Arithmetic and Logic Unit (ALU): The arithmetic and logic unit (ALU) is the part where actual computations take place. It consists of circuits that perform arithmetic operations (e.g. addition, subtraction, multiplication, division over data received from memory and capable to compare numbers (less than, equal to, or greater than etc)).

Control Unit: The control unit acts as a central nervous system for the components of the computer. It manages and coordinates the entire computer system. It obtains instructions from the program stored in main memory,

interprets the instructions, and issues signals that cause other units of the system to execute them. the control unit directs and controls the activities of the internal and external devices.

COMPUTER LANGUAGE

A language is defined as the medium of expression of thoughts. All the human beings in this world communicate with each other by a language. Similarly, computer also needs some expression medium to communicate with others.

A computer follows the instructions given by the programmer to perform a specific job. To perform a particular task, programmer prepares a sequence of instructions, known as program. A program written for a computer is known as Software.

1) First Generation Languages -1GLs (Machine language):When the human being started programming, the computer the instruction were given to it in a language that it could easily understand is called as machine language. The binary language of 1 and 0 is known as Machine language. Any instruction in this language is given in the form of 1s and 0s.

2) Second Generation Languages- 2GLs (Assembly Language):The first language similar to English was developed in 1950 which was known as Assembly Language or Symbolic Programming Languages. An assembly language is a low-level programming language for microprocessors (CPU) and other programmable devices.

3) Third Generation Languages- (3GLs) (High Level Languages):The languages developed which were nearer to the English language in 1960 were known as High Level languages. The different high level languages are FORTRAN, COBOL, BASIC, PASCAL, PL-1 and many others.

4) Fourth Generation Languages- (4GLs) (Query languages):The 3GLs are procedural in nature means HOW the problem get programmed and the procedures require the knowledge of HOW the problem will be solved. .4GLs are non-procedural that focuses on WHAT of the problem is coded means 'WHAT is required'. The main aim of 4GLs is to be cut down on developed and maintenance time and making it easier for users. Example of 4GL is SQL (Structured Query Language).

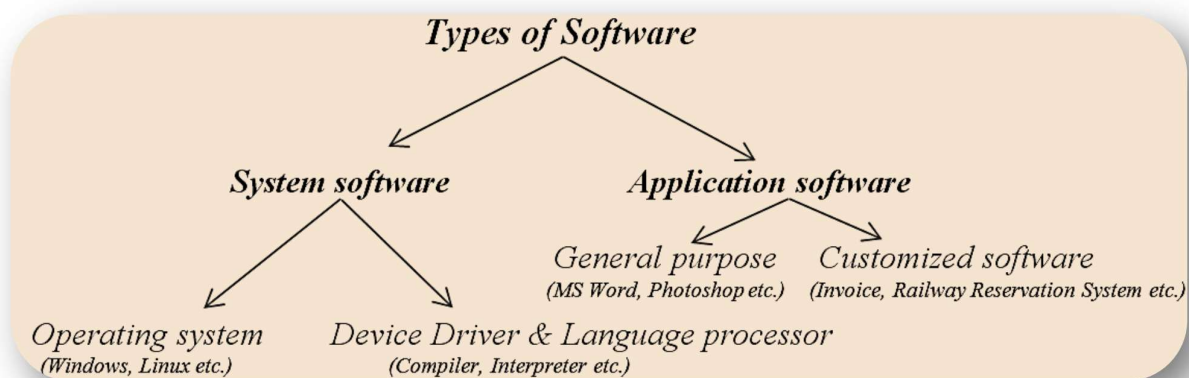
5) Fifth Generation Language-(5GLs):The 5GLs are designed to make the computer "Smarter". The use of 5GLs language touches on expert systems, computerized collection of the knowledge of many human experts, artificial intelligence and independently smart computer systems. Examples of 5GL are Prolog, OPS5 and Mercury.

Graphical User Interface (GUI) based languages: These are the languages which use a Graphical User Interface to write, compile & execute the program with ease. Some of them are-

- ▶ Visual Basic
- ▶ Visual C++
- ▶ C# (Pronounced as C sharp)
- ▶ Visual Basic.NET
- ▶ Visual Basic 2005

SOFTWARE

Computer software also called program is a set of instructions that directs a computer to perform specific tasks or operations. Computer software consists of computer programs and libraries.



System software: Software that directly operates the computer hardware to provide basic functionality needed by users and other software and to provide a platform for running application software. System software includes:

- ▶ **Operating system (OS):** Operating system manages resources of computer system like memory, CPU, hard disk, printer etc. also provides an interface between user and computer system & provides various services to other software.
- ▶ **Language Processor & Device drivers:** All the devices like mouse, keyboard, modem etc needs at least one corresponding device driver. A device driver is a program that controls a device. A language processor is a hardware device designed or used to perform tasks, such as processing program code to machine code. Language processors are found in languages such as Fortran and COBOL

Application software: Software that performs special functions or provides entertainment functions beyond the basic operation of the computer itself. There are many different types of application software.

- ▶ **General purpose:** Microsoft Word, Microsoft Excel, MS PowerPoint, Photoshop etc.
- ▶ **Customized:** Invoice Management System, Airline Reservation System etc.
- ▶ **Utilities:** Antivirus, Memory tester, Disk partitioning and Disk defragmenter etc.

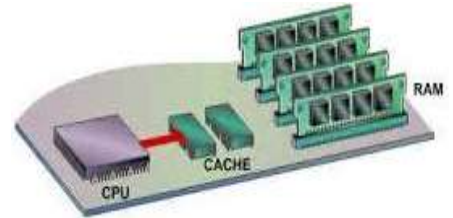
COMPUTER MEMORY

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.

Memory is primarily of three types

- ▶ Cache Memory
- ▶ Primary Memory or 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.



Advantages

The advantages of cache memory are as follows:

- 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.

Disadvantages

The disadvantages of cache memory are as follows:

- Cache memory has limited capacity.
- It is very expensive.

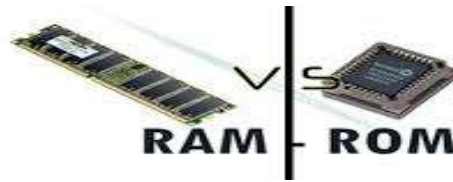
Primary Memory (Main Memory): Primary memory is also known as main memory. It holds only those data and instructions on which computer is currently working. It has limited capacity and data is lost when power is switched off. It is generally made up of semiconductor device.



Characteristics 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.
- Faster than secondary memories.
- A computer cannot run without primary memory.

Primary memory can be divided into RAM (Random Access Memory) and ROM (Read Only Memory).



RAM	ROM
Random Access Memory.	Read Only Memory.
It is Volatile or temporary	It is Non Volatile or permanent
Data gets erased when power	Data stored permanently.
Faster memory.	Slow memory.
It is used in the normal operations of a computer after starting up and loading the operating system.	A ROM chip is used primarily in the start-up process of a computer.

Secondary Memory

It is also known as external memory or non-volatile memory. It is slower than main memory. These are used for storing data or information permanently. For example: hard disk, CD-ROM, DVD etc.

Characteristic of Secondary memory

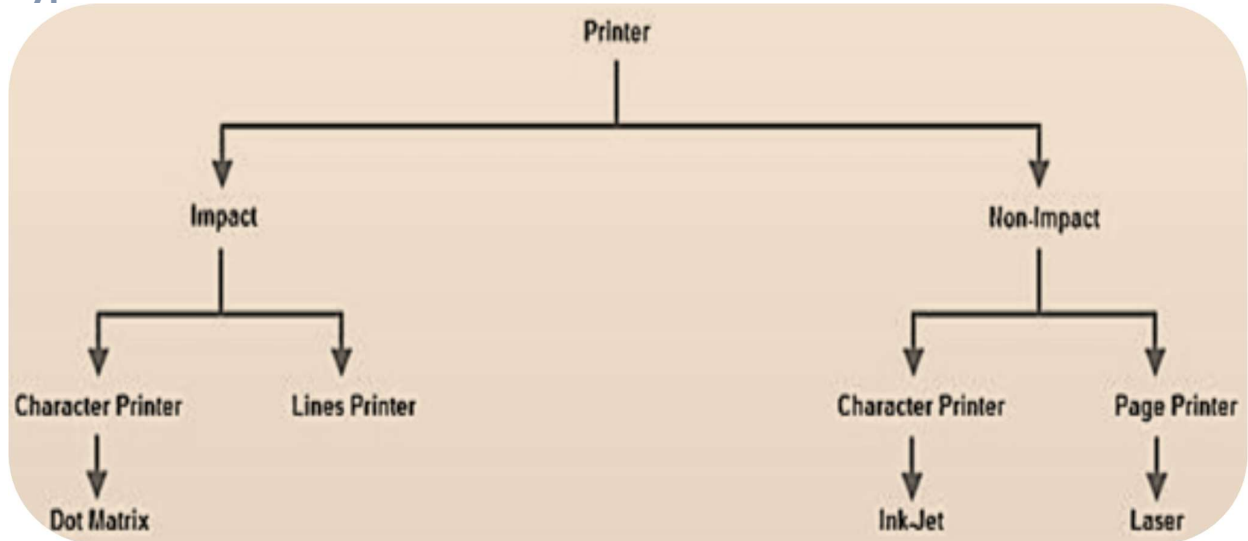
- These are magnetic and optical memories
- It is non-volatile memory which is used for storage of data in a computer.
- Data is permanently stored even if power is switched off.
- Computer may run without secondary memory.
- Slower than primary memories.



PRINTERS

Printers are output devices which are used to prepare permanent output on paper.

Types of Printers



Printers can be divided into two main categories:

Impact Printers: It uses hammers or pins strike against a ribbon and paper to print the text. This mechanism is known as electro-mechanical mechanism. They are of two types.

- ▶ **Character Printer**
- ▶ **Line Printer**

Character Printer: It prints only one character at a time. It has relatively slower speed. e.g. Dot matrix printers.

Dot Matrix Printer: It prints characters as combination of dots. These have a matrix of pins on the print head of the printer which form the character. There is a carbon between the pins & the paper. The words get printed on the paper when the pin strikes the carbon. There are generally 9 or 24 pins.



Line Printer The line printer is an impact computer printer that prints one entire line of text at a time. It is mostly associated with unit record equipment and the early days of digital computing, but the technology is still in use.

Non-Impact Printers: These printers use non-Impact technology such as ink-jet or laser technology. These printers provide better quality of output at higher speed. These printers are of two types:



Ink-Jet Printer: It prints characters by spraying patterns of ink on the paper from a nozzle or jet. It prints from nozzles having very fine holes, from which ink is pumped out to create various letters and shapes.



Laser Printer: It utilizes a laser beam to produce an image on a drum. The light of the laser alters the electrical charge on the drum wherever it hits. The drum is then rolled through a reservoir of toner, which is picked up by the charged portions of the drum. Finally, the toner is transferred to the paper through a combination of heat and pressure. They are much quieter than dot-matrix. They are also relatively fast. The speed of laser printers ranges from about 4 to 20 pages of text per minute (ppm).



WORKSHEET

Identify the following peripherals as INPUT device or OUTPUT device.

1. Monitor _____
2. Keyboard _____
3. Scanner _____
4. Laser Printer _____
5. Mouse _____
6. Speakers _____
7. Digital Camera _____

Fill in the blank with the correct answer from the box. Some may be used more than once or not at all.

CPU	BIOS	power supply	hard drive	network card
Motherboard	RAM	USB Port	ROM	video card

1. I connect computers and allow them to talk to each other. _____
2. I wake up the computer and remind it what to do. _____
3. I am the brain of the computer. _____
4. Information is stored on my magnetic cylinders. _____
5. I hold all of the other circuit boards. _____
6. I handle the graphics that are displayed on the monitor. _____
7. I am the type of port used by flash drives _____

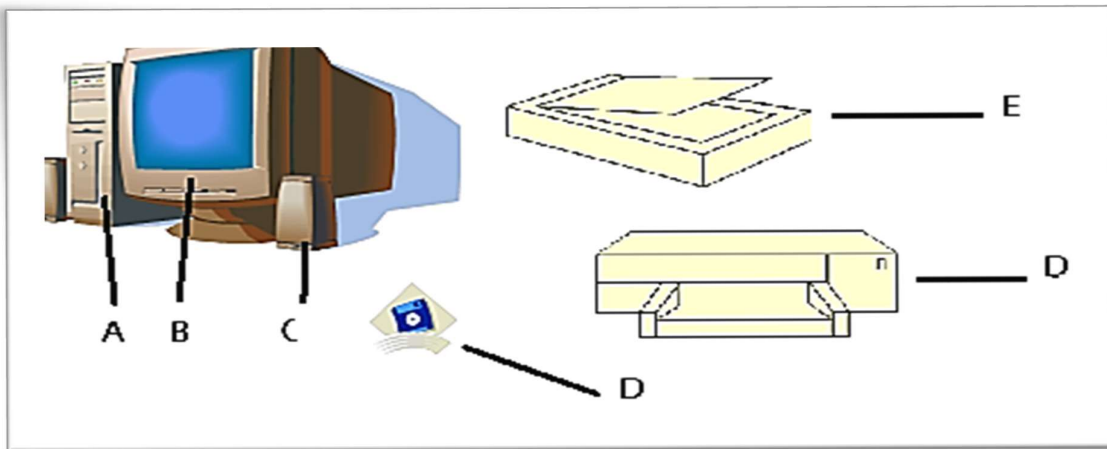
Fill in the blanks with the vocabulary words from the box. Use each word only once.

information	flash drive	CD primary	DVD	secondary
Windows operating system	graphics	upgraded	user friendly	

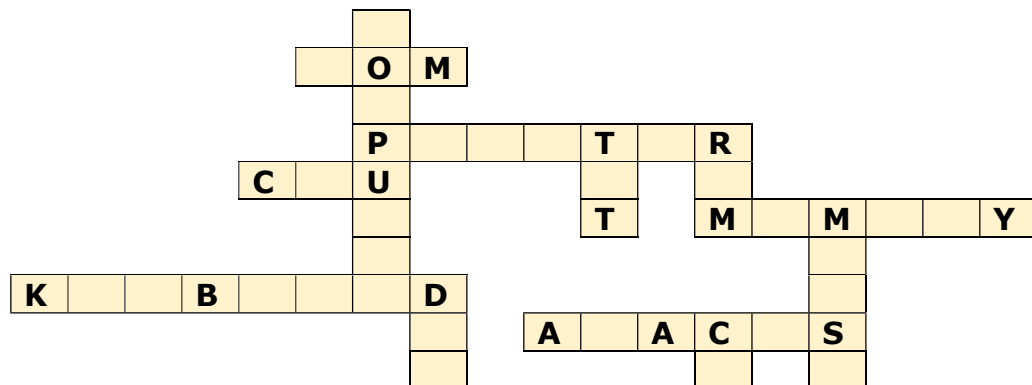
1. _____ memory is stored on chips located on the motherboard.
2. _____ memory is stored on the hard drive.
3. A _____ can hold information greater than a CD or DVD.
4. A _____ usually holds up to 650 to 700 MB.
5. A _____ holds even more information at least 7 GB.
6. The purpose of storage in a computer is to hold _____ or data.
7. A computer that is easy to operate is called _____

8. _____ is the most common operating system for PCs.
9. Operating systems are constantly being _____ as technology advances.
10. A Graphical User Interface (GUI) uses _____ to help the user navigate within the computer system.

Label the parts of the Computer System correctly.



Solve the crosswords using the words from the lesson



Match the following

- | | |
|----------------------|------------------------|
| 1 CD | a Impact printer |
| 2 Laser printer | b Output device |
| 3 Dot matrix printer | c Primary memory |
| 4 Abacus | d Non-Impact printer |
| 5 ROM | e Input device |
| 6 Monitor | f System software |
| 7 RAM | g Read only memory |
| 8 Mouse | h Application software |
| 9 Operating system | i Babylonia |
| 10 MS Word | j Optical media |

INTRODUCTION TO PYTHON

Basics of Python programming, Python interpreter - interactive and script mode, the structure of a program, indentation, identifiers, keywords, constants, variables, types of operators, precedence of operators, data types, mutable and immutable data types, statements, expressions, evaluation and comments, input and output statements, data type conversion, debugging.

Control Statements: if-else, for loop

Lists: list operations - creating, initializing, traversing and manipulating lists, list methods and built-in functions.

Dictionary: concept of key-value pair, creating, initializing, traversing, updating and deleting elements, dictionary methods and built-in functions.

Introduction :

Python is a General-Purpose object-oriented programming language, which means that it can model real-world entities. The distinctive feature of Python is that it is an **interpreted language**. The Python **IDLE** (Integrated Development Environment) executes instructions one line at a time. This also lets us use it as a calculator.

Guido van Rossum named it after the comedy group Monty Python Flying Circus a Comedy Serial of BBC.

Python Features :

- a. Easy
- b. Interpreted
- c. Object-Oriented
- d. Free and Open Source
- e. Portable
- f. Large Python Library

Interpreter : It executes / converts high level code into machine code (source to target code) line by line, and slow compare to compiler

Compiler : It executes / converts code in one go and faster compared to Interpreter.

MODES OF PYTHON WORKING

Interactive Mode

Interactive mode is running blocks or a single line of Python code. The code executes via the Python shell. Interactive mode is handy when you just want to execute basic Python commands or you are new to Python programming.

Script Mode

If you need to write a long piece of Python code or your Python script spans multiple files, interactive mode is not recommended. Script mode is the way to go in such cases. In script mode, You write your code in a text file then save it with a .py extension which stands for "Python". This file can be executed by Python Interpreter later on.

PYTHON KEYWORDS

Python keywords are special reserved words that have specific meanings and purposes and can't be used for anything but those specific purposes. An example of something you *can't* do with Python keywords is assign something to them.

Examples of keywords are : False, else, import, pass, in, is, True, and continue etc.

Python Identifiers

Identifiers are **names given to different entities such as constants, variables, structures, functions, etc.** Example: int, amount; True, totalbalance; In the above example, amount and totalbalance are identifiers and int, and double are keywords.

Rules for Identifiers

- Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore _
- An identifier cannot start with a digit. ...
- Keywords cannot be used as identifiers. ...
- We cannot use special symbols like !, @, #, \$, % etc. ...
- An identifier can be of any length.

BUILT-IN DATA TYPES

In programming, data type is an important concept. Variables can store data of different types, and different types can do different things. Python has the following data types built-in by default, in these categories:

Text Type:	Str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	Set
Boolean Type:	Bool (Only True and False Value)
None Type:	None

PYTHON OPERATORS

Operators are used to perform operations on variables and values. In the example below, we use the + operator to add together two values: Python divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators

PYTHON ARITHMETIC OPERATORS

Arithmetic operators are used with numeric values to perform common mathematical operations:

Operator	Name	Example	
+	Addition	x + y	
-	Subtraction	x - y	
*	Multiplication	x * y	
/	Division	x / y	
%	Modulus	x % y	
**	Exponentiation	x ** y	
//	Floor division	x // y	

PYTHON ASSIGNMENT OPERATORS

Assignment operators are used to assign values to variables:

Operator	Example	Same As	
=	x = 5	x = 5	
+=	x += 3	x = x + 3	
-=	x -= 3	x = x - 3	
*=	x *= 3	x = x * 3	
/=	x /= 3	x = x / 3	
%=	x %= 3	x = x % 3	
//=	x //= 3	x = x // 3	
**=	x **= 3	x = x ** 3	
&=	x &= 3	x = x & 3	
=	x = 3	x = x 3	
^=	x ^= 3	x = x ^ 3	
>>=	x >>= 3	x = x >> 3	
<<=	x <<= 3	x = x << 3	

PYTHON LOGICAL OPERATORS

Logical operators are used to combine conditional statements:

Operator	Description	Example	
and	Returns True if both statements are true	x < 5 and x < 10	
or	Returns True if one of the statements is true	x < 5 or x < 4	
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)	

PYTHON IDENTITY OPERATORS

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location:

Operator	Description	Example	
is	Returns True if both variables are the same object	x is y	
is not	Returns True if both variables are not the same object	x is not y	

PYTHON MEMBERSHIP OPERATORS

Membership operators are used to test if a sequence is presented in an object:

Operator	Description	Example	
in	Returns True if a sequence with the specified value is present in the object	x in y	
not in	Returns True if a sequence with the specified value is not present in the object	x not in y	

PYTHON BITWISE OPERATORS

Bitwise operators are used to compare (binary) numbers:

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and

WHAT ARE MUTABLE AND IMMUTABLE DATA TYPES IN PYTHON?

Python Mutable data types are those whose values can be changed in place whereas Immutable data types are those that can never change their value in place.

Here are Mutable data types :

1. Lists
2. Dictionaries
3. Sets

And Immutable data types in Python:

1. Integers
2. Floating-Point numbers
3. Booleans
4. Strings
5. Tuples

SAMPLE QUESTION AND ANSWERS

1. Who developed python programming language? Write any two name(s) of IDE for python language.

Ans: Guido Van Rossum in 1990 developed python programming language

2. Write Names of any two Python IDE

Ans. IDE- Spyder, Pycharm, IDLE

3. Identify the valid identifiers from following-
f@name , as , _tax , roll_no , 12class , totalmarks , addr1

Ans: _tax, roll_no, totalmarks, add1

4. What is difference between equality (==) and identity (is) operator?

Ans: Equality (==) compares values while identity (is) compares memory address of variables/objects.

5. Write python code to convert the time given in minutes into hours and minutes.

Ans:

```
min=int(input("Enter time in minutes &"))
h=min//60
m=min%60
print("Hours=", h)
print("Minutes=", m)
```

6. Evaluate (to true or false) each of the following expression:

14<=14, 14<14, -14 > -15, -15 >=15

1. True, False, True, False
2. False, False , True, True
3. True, False, True, True
4. False, True, False, False

Ans: True, False, True, True

7. What will be the output of the following python statements?

```
a,b,c = 10,40,20
a,c,b = b+10, a+20, c-10
print(a,b,c)
print(a+b//c**2)
Ans: 50 10 30 50
```

CONTROL FLOW STATEMENTS IN PYTHON

A program's **control flow** is the order in which the program's code executes. The control flow of a Python program is regulated by conditional statements, loops, and function calls.

Python has *three* types of control structures:

- **Sequential** - default mode
- **Selection** - used for decisions and branching
- **Repetition** - used for looping, i.e., repeating a piece of code multiple times.

1. SEQUENTIAL

Sequential statements are a set of statements whose execution process happens in a sequence. The problem with sequential statements is that if the logic has broken in any one of the lines, then the complete source code execution will break.

2. SELECTION/DECISION CONTROL STATEMENTS

In Python, the selection statements are also known as *Decision control statements* or *branching statements*. The selection statement allows a program to test several conditions and execute instructions based on which condition is true.

Some Decision Control Statements are:

- Simple if
- if-else
- nested if
- if-elif-else
-

Simple if: *If statements* are control flow statements that help us to run a particular code, but only when a certain condition is met or satisfied. A *simple if* only has one condition to check.

```
# If the number is positive, we print an appropriate message
num = 3
if num > 0:
    print(num, "is a positive number.")
print("This is always printed.")

num = -1
if num > 0:
    print(num, "is a positive number.")
print("This is also always printed.")
```

if-else: The *if-else statement* evaluates the condition and will execute the body of *if* if the test condition is True, but if the condition is False, then the body of *else* is executed.

#Take user input and find number is odd or even

```
inp_num = input("Enter a number: ")
```

#Convert string to int

```
inp_num = int(inp_num)
```

```
if inp_num == 0:
```

```
    print(inp_num, "is Even")
```

```
elif inp_num%2==0:
```

```
    print(inp_num, "is Even")
```

```
else:
```

```
    print(inp_num, "is Odd")
```

if-elif-else: The *if-elif-else statement* is used to conditionally execute a statement or a block of statements.

```
# '''In this program, we check if the number is positive or negative or zero  
and display an appropriate message'''
```

```
num = 3.4
```

```
# Try these two variations as well:
```

```
# num = 0
```

```
# num = -4.5
```

```
if num > 0:
```

```
    print("Positive number")
```

```
elif num == 0:
```

```
    print("Zero")
```

```
else:
```

```
    print("Negative number")
```

nested if: *Nested if statements* are an if statement inside another if statement.

```
#'''In this program, we input a number check if the number is positive  
or negative or zero and display an appropriate message This time we  
use nested if statement'''
```

```
num = float(input("Enter a number: "))
```

```
if num >= 0:
```

```
    if num == 0:
```

```
        print("Zero")
```

```
    else:
```

```
        print("Positive number")
```

```
else:
```

```
    print("Negative number")
```

3. REPETITION

A **repetition statement** is used to repeat a group(block) of programming instructions.

In Python, we generally have two loops/repetitive statements:

- for loop
- while loop

for loop: A *for loop* is used to iterate over a sequence that is either a list, tuple, dictionary, or a set. We can execute a set of statements once for each item in a list, tuple, or dictionary.

```
# Program to find the sum of all numbers stored in a list List of
numbers
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
```

```
# variable to store the sum
sum = 0
```

```
# iterate over the list
for val in numbers:
    sum = sum+val
```

```
print("The sum is", sum)
```

```
# Program to iterate through a list using indexing
```

```
genre = ['pop', 'rock', 'jazz']
```

```
# iterate over the list using index
for i in range(len(genre)):
    print("I like", genre[i])
```

```
digits = [0, 1, 5]
```

```
for i in digits:
    print(i)
else:
    print("No items left.")
```

while loop: In Python, *while loops* are used to execute a block of statements repeatedly until a given condition is satisfied. Then, the expression is checked again and, if it is still true, the body is executed again. This continues until the expression becomes false.

```
# Program to add natural numbers up to
# sum = 1+2+3+...+n
```

```
# To take input from the user,
# n = int(input("Enter n: "))
```

```

n = 10

# initialize sum and counter
sum = 0
i = 1

while i <= n:
    sum = sum + i
    i = i+1    # update counter

# print the sum
print("The sum is", sum)

'''Example to illustrate the use of else statement with the while loop'''
counter = 0

while counter < 3:
    print("Inside loop")
    counter = counter + 1
else:
    print("Inside else")

```

SAMPLE QUESTION AND ANSWERS

1. What will this code print?

```

a = True
if a:
    print("A")
else:
    print("B")
print("C")

```

Ans:

A
C

2. What is empty statement? Which python statement can be termed as empty statement?

Ans: A statement that does nothing. Where the syntax of the language require the presence of statement but the logic of the program does not. Pass statement is an empty statement.

3. Re-write the following program using 'while' loop

```

sum = 0
for x in range(10):
    sum = sum + x**2
print(sum)

```


Ans:

```
sum=0
x=0
while x<10:
    sum=sum+x**2
    print(sum)
```

4 What will be the output of the following:

```
m,n,p=40,20,0
while m>n:
    m = m % 7
    n = n//2
    p += m+n
    print(m,n,p)
```

Ans:

5 10 15

LIST IN PYTHON

WHAT IS LIST IN PYTHON?

A list is a sequence datatype. Elements of list are enclosed in square brackets separated by comma. Elements of a list can be of any type. for example :

```
L = [1, 3, 5, 'python', 'game']
```

The above list has five elements and are enclosed in square brackets(separated by comma)

METHODS OF LIST

METHODS OF LIST (BUILT-IN FUNCTIONS)

Python offers various methods of list (built in functions) to perform operations on List. Let's discuss one by one

- append():
- extend()
- insert():
- reverse() :
- len():
- sort():
- count() :
- clear():
- pop():
- del Statement :
- remove():
- index():
- max() :
- min() :

APPEND():

THIS FUNCTION ADD A SINGLE ELEMENT AT THE END OF THE LIST. FOR EXAMPLE

```
L = [34, 45, 56, 23, 12]
L.append(77)
print(L)
```

OUTPUT : [34, 45, 56, 23, 12, 77]

L.append(28,9) will return type error as append function can take only one argument This function can add a list at the end of another list. for example

```
L.append([28, 29])
```

OUTPUT : [34, 45, 56, 23, 12, 77, [28, 29]]

Q1. WRITE A PROGRAM TO ACCEPT FIVE NUMBERS FROM THE USER AND ADD IN THE GIVEN

LIST. L = [34, 23, 12]

Ans.

```
L = [34, 23, 12]
k = 0
for i in range(5):
    k = int(input("Enter any number"))
    L.append(k)
print("List after appending is: ", L)
```

output

Execution of the program

```
Enter any number : 2
Enter any number : 3
Enter any number : 5
Enter any number : 6
Enter any number : 9
```

List after appending is: [34, 23, 12, 2, 3, 5, 6, 9]

Q2. WRITE A PROGRAM TO ACCEPT 10 NUMBERS FROM THE USER AND IF THE NUMBER IS EVEN THEN ADD THE ELEMENTS IN LIST L1 OTHERWISE ADD IN LIST L2.

Ans.

```
L1 = [ ]
L2 = [ ]
for i in range(10):
    k = int(input("Enter any number : "))
    if k%2==0:
        L1.append(k)
    else:
        L2.append(k)
```

```
print("Even number List :",L1)
print("Odd number List :",L2)
```

OUTPUT : Execution of the program:

```
Enter any number : 1
Enter any number : 2
Enter any number : 3
Enter any number : 4
Enter any number : 5
Enter any number : 6
Enter any number : 7
Enter any number : 8
Enter any number : 9
Enter any number : 10
```

```
Even number List : [2, 4, 6, 8, 10]
Odd number List : [1, 3, 5, 7, 9]
```

EXTEND()

THIS FUNCTION ADDS ALL THE ELEMENTS OF ONE LIST AT THE END OF THE OTHER LIST. FOR EXAMPLE :

```
L1 = [1, 6, 3, 9]
L2 = [12, 13, 18, 19]
L1.extend(L2)
print(L1)
```

Output : [1, 6, 3, 9, 12, 13, 18, 19]

All the elements of list L2 are added at the end of the list L1

Q3. WRITE THE OUTPUT OF THE FOLLOWING TWO CODING:

```
L1 = [1, 2, 3]
L2 = [7, 8 ,9]
L1.append(L2)
print(L1)
```

```
OUTPUT : [1, 2, 3, [7, 8, 9]]
L1 = [1, 2, 3]
L2 = [7, 8 ,9]
L1.extend(L2)
print(L1)
```

OUTPUT : [1, 2, 3, 7, 8, 9]

INSERT():

THIS FUNCTION HELP US TO ADD AN ELEMENT AT A SPECIFIC INDEX VALUE. FOR EXAMPLE

```
L1 = [23, 12, 45, 32]
L1.insert(3, 77) # This statement will add element 77 at index
value 3 ie position number 4
print(L1)
```

OUTPUT : [23, 12, 45, 77, 32]

```
L1.insert(1, 91) # This statement will add element 91 at index
value 1.
print(L1)
```

OUTPUT : [23, 91, 12, 45, 77, 32]

REVERSE() :

THIS FUNCTION SIMPLY REVERSE THE ORDER OF ALL THE ELEMENTS IN THE LIST. FOR EXAMPLE :

```
L1 = [23, 12, 45, 32]
L1.reverse( )
print(L1)
```

OUTPUT : [32, 45, 12, 23] #The given list is reversed.

LEN():

THIS FUNCTION RETURNS THE LENGTH OF THE LIST. FOR EXAMPLE

```
L1 = [12, 34, 43, 23, 78, 90, 1]
print(len(L1))
```

OUTPUT : 7

```
L2 = ["Sumit", "Naman", "Parth", 76, 90, "Mini"]
print(len(L2))
```

OUTPUT : 6

SORT():

THIS FUNCTION ARRANGE ALL THE ELEMENTS IN INCREASING ORDER (BY DEFAULT). FOR EXAMPLE

```
L1 = [12, 34, 43, 23, 78, 90, 1]
L1.sort( )
print(L1)
```

OUTPUT : [1, 12, 23, 34, 43, 78, 90]

```
L2 = ["Sumit", "Naman", "Parth", 76, 90, "Mini"]
L2.sort( )
print(L2)
```

OUTPUT : TypeError: '<' not supported between instances of 'int' and 'str'

```
L3 = ["Sumit", "Naman", "Parth", "Mini"]
L3.sort( )
print(L3)
```

OUTPUT : ['Mini', 'Naman', 'Parth', 'Sumit']

This function arranges String elements according to their ASCII value.

TO ARRANGE ELEMENTS IN DECREASING/DESCENDING ORDER.

```
L1 = [12, 34, 43, 23, 78, 90, 1]
L1.sort( reverse = True)
print(L1)
```

OUTPUT : [90, 78, 43, 34, 23, 12, 1]

OR

```
L1 = [12, 34, 43, 23, 78, 90, 1]
L1.sort( )
L1.reverse( )
print(L1)
```

OUTPUT : [90, 78, 43, 34, 23, 12, 1]

COUNT() :

THIS FUNCTION RETURNS THE FREQUENCY OF AN ELEMENT IN THE LIST. IN OTHER WORDS, WE CAN SAY THAT THIS FUNCTION RETURNS HOW MANY TIMES AN ELEMENT HAS OCCURRED IN THE LIST. FOR EXAMPLE

```
L1 = [1, 2, 5, 2, 6, 1, 7, 9, 1]
print(L1.count(1))
```

OUTPUT : 3
`print(L1.count(2))`

OUTPUT : 2

CLEAR():

THIS FUNCTION REMOVE ALL THE ELEMENTS OF THE LIST. FOR EXAMPLE

```
L1 = [1, 2, 5, 2, 6, 1, 7, 9, 1]
L1.clear( )
print(L1)
```

OUTPUT : []

DELETION OPERATION :

POP():

THIS FUNCTION DELETE/REMOVE THE ELEMENT FROM THE SPECIFIED INDEX. THIS FUNCTION ALSO RETURN THE DELETED ELEMENT. FOR EXAMPLE

```
L1 = [90, 56, 87, 98, 23, 6, 78]
```

```
print(L1.pop(3 )) #Output of this statement shows that it return the
element.
print(L1)
```

OUTPUT:

98

[90, 56, 87, 23, 6, 78]

NOTE : If we don't give any index value in pop() function then it will delete the last element.

```
L1 = [90, 56, 87, 98, 23, 6, 78]
print(L1.pop( )) #Output of this statement shows that it return the
element.
print(L1)
```

OUTPUT :

78

[90, 56, 87, 98, 23, 6]

```
L1 = ["Amit", "Sumit", "Naman", "Manan", "Kapil"]
print(L1.pop(-2)) # pop( ) function work with negative index value also.
print(L1)
```

OUTPUT :

Manan

['Amit', 'Sumit', 'Naman', 'Kapil']

DEL STATEMENT :

THE DEL STATEMENT DELETE THE ELEMENT FROM THE SPECIFIED INDEX. THIS STATEMENT DOES NOT RETURN THE ELEMENT. FOR EXAMPLE

```
L1 = [90, 56, 87, 98, 23, 6, 78]
del L1[3]
print(L1)
```

OUTPUT: [90, 56, 87, 23, 6, 78]

More than one element can be deleted by using del statement. for example

```
L1 = [90, 56, 87, 98, 23, 6, 78]
del L1[2 : 5] # will delete three elements of index value 2, 3 and 4
print(L1)
```

OUTPUT: [90, 56, 6, 78]

Q. What is the difference between pop () function and del statement?

Ans.

pop ()	del statement
This function returns the deleted element	This statement does not return the deleted element
One element can be deleted at one time	Multiple elements can be deleted. for example, del L1[2:5] will delete 3 elements from list L1
Index value of the element to be deleted is written in parenthesis. for example, L1.pop (2)	Index value of the element to be deleted is written in square brackets. for example, del L1[2]

REMOVE():

THIS FUNCTION DELETE THE FIRST OCCURRENCE OF SPECIFIED ELEMENT. THIS FUNCTION IS USED WHEN WE KNOW THE ELEMENT TO BE DELETED BUT NOT THE INDEX VALUE. FOR EXAMPLE:

```
L1 = [90, 56, 87, 98, 23, 6, 78, 98]
L1.remove(98) #will remove element 98 from the list which is at index
val 3
print(L1)
```

OUTPUT : [90, 56, 87, 23, 6, 78, 98]

Q. WRITE A PROGRAM TO REMOVE ALL EVEN NUMBERS FROM THE LIST.

Ans.

```
L1 = [90, 56, 87, 98, 23, 6, 78, 98]
l = len(L1)
i=0
while i < l:
    if L1[i]%2==0:
        del L1[i]
        l=l-1
        i=i-1
    i = i+1
print(L1)
```

SEARCHING : METHODS OF LIST :

ELEMENTS IN THE LIST CAN BE SEARCHED USING INDEX() METHOD.

INDEX():

THIS FUNCTION SIMPLY RETURNS THE INDEX VALUE OF THE SPECIFIED ELEMENT. FOR EXAMPLE

```
L1 = [90, 56, 87, 98, 23, 6, 78, 98]
print(L1.index(87))
```

OUTPUT : 3

If we specified an element which is not present in the list, then it will return an error. for example

```
L1 = [90, 56, 87, 98, 23, 6, 78, 98]
print(L1.index(77))
```

OUTPUT : ValueError: 77 is not in list

Q4. WRITE A FUNCTION INDV(L, N) WHICH TAKES A LIST AND A NUMBER AS AN ARGUMENT AND RETURN THE INDEX VALUE OF NUMBER PASSED AS AN ARGUMENT.

Ans.

```
L1 = [23, 12, 45, 32, 12]
def indv(L, n):
    if n in L:
        print(L.index(n))
    else:
        print("number not available in list")
indv(L1, 45)
```

OUTPUT : 2

FINDING LARGEST AND SMALLEST VALUE

MAX() :

This function returns the largest value from the list. for example

```
L1 = [90, 56, 87, 98, 23, 6, 78, 98]
print(max(L1))
```

OUTPUT : 98

```
L1 = ["Amit", "Sumit", "Naman", "Manan", "Kapil"]
print(max(L1)) # For character value, it checks according to ASCII
Value.
```

OUTPUT : Sumit

MIN() :

This function return the smallest value from the list. for example

```
L1 = [90, 56, 87, 98, 23, 6, 78, 98]
```

```
print(min(L1))
```

OUTPUT : 6

```
L1 = ["Amit", "Sumit", "Naman", "Manan", "Kapil"]
```

```
print(min(L1)) # For character value, it checks according to ASCII Value.
```

OUTPUT : Amit

PYTHON DICTIONARY

A python dictionary is a collection of elements where each element is a combination of Key-Value pair. Each value/values is associated with a unique key. All the Key-Value pairs are enclosed in Curly braces. In other words we can say that "Dictionaries are mutable, unordered collection of elements in the form of Key-Value Pairs which are enclosed in curly braces".

EXAMPLE OF DICTIONARY:

```
A = {1 : "One", 2 : "Two", 3 : "Three"}
B = {"A" : "Apple", "B" : "Ball", "C" : "Cat"}
```

Dictionary A has numeric Keys (1, 2, 3) and Values("One", "Two", "Three") are in String, while dictionary B has both Keys('A', 'B', 'C') and Values("Apple", "Ball", "Cat") are in string.

CHARACTERISTICS OF PYTHON DICTIONARY:

1. The combination of Key and Value is called Key-Value Pair.
2. Keys and it's values are separated by colon(:)
3. Different Key-Value pairs are separated by comma(,).
4. Keys are unique for each Value.
5. Keys of dictionary must be of immutable type like string, number etc.

METHOD TO CREATE EMPTY DICTIONARY

There are two ways to create an empty dictionary which are as follows

1. A = { } # A is an empty dictionary
2. A = dict() # dict() method will create an empty dictionary.

METHOD TO CREATE DICTIONARY AT RUN TIME :

Q1. Write a program to enter roll number and names of five students and store the data in dictionary.

```
d = { }
for i in range(5):
    rno=int(input("Enter roll number"))
    name=input("Enter name of student")
    d[rno] = name
print(d)
```

OUTPUT #Execution of Above Program

```
Enter roll number: 1
Enter name of student: Amit
Enter roll number: 2
Enter name of student: Sunil
```

Enter roll number: 3
Enter name of student: Lata
Enter roll number: 4
Enter name of student: Suman
Enter roll number: 5
Enter name of student: Ravi

```
{1: 'Amit', 2: 'Sunil', 3: 'Lata', 4: 'Suman', 5: 'Ravi'}
```

Q2. Write a program to store book id, book name and price of three books and store the data in dictionary named "dict"

```
dict = {}

for i in range(3):
    b_id = int(input("Enter book id : "))
    b_name = input("Enter book name : ")
    b_price = int(input("Enter price of book : "))
    temp=[b_name,b_price]
    dict[b_id] = temp

print(dict)
```

OUTPUT #Execution of Above Program

```
Enter book id : 1
Enter book name : CS
Enter price of book : 350
Enter book id : 2
Enter book name : GK
Enter price of book : 100
Enter book id : 3
Enter book name : IP
Enter price of book : 250
{1: ['CS', 350], 2: ['GK', 100], 3: ['IP', 250]}
```

TRAVERSING A PYTHON DICTIONARY

It means to access each element of dictionary by using loop. for example

```
A = {1 : "One", 2 : "Two", 3 : "Three"}
for k in A:
    print(k, "--->", A[k])
```

OUTPUT

```
1--->One
2--->Two
3--->Three
```

HOW TO APPEND VALUE IN PYTHON DICTIONARY

We can easily add new element in dictionary by the following way

```
A = {1 : "One", 2 : "Two", 3 : "Three"}
A[4] = "Four"
print(A)
```

OUTPUT

{1 : "One", 2 : "Two", 3 : "Three", 4 : "Four"}

We can also join two dictionaries into one by using update() method. It merge the keys and values of one dictionary into other and overwrites the values of the same key.

```
A = {1 : "One", 2 : "Two", 3 : "Three"}
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}
A.update(B)
print(A)
```

OUTPUT

{1: 'Amit', 2: 'Sunil', 3: 'Three', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}

#It over writes the values of same keys and add the values of different keys
UPDATE VALUES IN A PYTHON DICTIONARY :

We can not change the key of an element, but can change the value of a respective key as follows

```
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}
B[2] = 'Ram'
print(B)
```

OUTPUT:

{1: 'Amit', 2: 'Ram', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}

REMOVING AN ELEMENT FROM A DICTIONARY

There are two ways by which we can delete the elements of dictionary:

1.By using del statement :

Syntax of using del statement is : del <dictionary-name>[key of element]

```
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}
del B[2] # It will remove the element of key 2
print(B)
```

OUTPUT:

{1: 'Amit', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}

```
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}
del B[3]
```

```
''' It will return an error (KeyError) if the key given is not present in the dictionary'''  
print(B)
```

OUTPUT:
KeyError

2. By Using pop() function : This function not only delete the element of required key but also return the deleted value.

```
(i) B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}  
a=B.pop(2) #It returns the element of Key - 2  
print(a)  
print(B)
```

OUTPUT
Sunil

```
(ii) B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}  
a=B.pop(6) #It returns the element of Key - 6  
print(a)  
print(B)
```

OUTPUT

Suman
{1: 'Amit', 2: 'Sunil', 5: 'Lata', 7: 'Ravi'}

PYTHON DICTIONARY METHODS:

There are many methods of dictionary in python. Let's do some of the important function with example.

LEN() :

This function returns the length of Dictionary means, it returns the total number of Key-Value pairs present in the dictionary. for example:

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}  
print("Length of dictionary A is :", len(A))  
print("Length of dictionary B is :", len(B))
```

OUTPUT

Length of dictionary A is : 3
Length of dictionary B is : 4

CLEAR() :

This function simply removes all items from the dictionary. for example

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}  
A.clear()
```

```
print("Length of dictionary A is :", len(A))
print("Length of dictionary B is :", len(B))
```

OUTPUT

Length of dictionary A is: 0

Length of dictionary B is: 4

ITEMS():

This function returns all the key value pair of dictionary in the form of list of tuples. for example

```
A = {1 : "One", 2 : "Two", 3 : "Three"}
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}
print(A.items())
print(B.items())
```

OUTPUT

dict_items([(1, 'One'), (2, 'Two'), (3, 'Three')])

dict_items([('A', 'Apple'), ('B', 'Bat'), ('C', 'Cat'), ('D', 'Doll')])

Q1. What is the difference between print(A) and print(A.items()) Where 'A' is dictionary given above.

Ans. There is difference in the output of dictionary as print command print all key-value pair in curly braces while items() return all key value pairs as a list of tuples. for example

```
A = {1 : "One", 2 : "Two", 3 : "Three"}
print(A.items())
print(A)
```

OUTPUT:

dict_items([(1, 'One'), (2, 'Two'), (3, 'Three')])

{1: 'One', 2: 'Two', 3: 'Three'}

KEYS():

This function return all the keys of dictionary in the form of List. for example :

```
A = {1 : "One", 2 : "Two", 3 : "Three"}
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}
print(A.keys())
print(B.keys())
```

OUTPUT:

dict_keys([1, 2, 3])

dict_keys(['A', 'B', 'C', 'D'])

VALUES():

This function return all the values of dictionary in the form of List. for example :

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {'A' : "Apple", 'B' : "Bat", 'C' : "Cat", 'D' : "Doll"}  
print(A.values())  
print(B.values())
```

OUTPUT:

```
dict_values(['One', 'Two', 'Three'])  
dict_values(['Apple', 'Bat', 'Cat', 'Doll'])
```

UPDATE():

This function merge the key value pair of two dictionary into one. In simple words it simply join/merge the two dictionary. It merge the keys and values of one dictionary into other and overwrites the values of the same key. for example:

```
A = {1 : "One", 2 : "Two", 3 : "Three"}  
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}  
A.update(B)  
print(A)
```

OUTPUT

{1: 'Amit', 2: 'Sunil', 3: 'Three', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}

#It over writes the values of same keys and add the values of different keys

POP():

This function not only delete the element of required key but also return the deleted value.

```
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}  
a=B.pop(2) #It returns the element of Key - 2  
print(a)  
print(B)
```

OUTPUT

Sunil

{1: 'Amit', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}

```
B = {1: 'Amit', 2: 'Sunil', 5: 'Lata', 6: 'Suman', 7: 'Ravi'}  
a=B.pop(6) #It returns the element of Key - 6  
print(a)  
print(B)
```

OUTPUT

Suman

{1: 'Amit', 2: 'Sunil', 5: 'Lata', 7: 'Ravi'}

SAMPLE ASSIGNMENT QUESTIONS WITH ANSWERS:

- 1 Who developed python programming language? Write any two name(s) of IDE for python language. 2
Ans: Guido Van Rossum in 1990 developed python programming language
Name of IDE- Spyder, Pycharm, IDLE
- 2 Identify the valid identifiers from following- f@name , as , _tax , roll_no , 12class , totalmarks , addr1 2
Ans: _tax, roll_no, totalmarks, add1
- 3 Write a python program to find the area and perimeter of a rectangle by accepting inputs of length and breadth from user. 2
(area=length*breadth)
(perimeter=2*(length+breadth))
Ans: length=float(input("Enter length of rectangle"))
breadth=float(input("Enter breadth of rectangle"))
area=length*breadth
perimeter=2*(length+breadth)
print("Area of rectangle",area)
print("Perimeter of rectangle",perimeter)
- 4 Write python statement to check if **a** is less than **b**? 1
Ans: if a<b:
- 5 What will this code print? 1
a = True
if a:
 print("A")
else:
 print("B")
print("C")
Ans:
A
C
- 6 What is difference between equality (==) and identity (is) operator? 2
Ans: Equality (==) compares values while identity (is) compares memory address of variables/objects.
- 7 "Comments are useful and easy way to enhance readability and understandability of a program." Elaborate with examples. 2

- 8 Write the difference between compiler and interpreter with example? 3

Ans:

Interpreter	Compiler
✓ Translates program one statement at a time.	✓ Scans the entire program and translates it as a whole into machine code.
✓ It takes less amount of time to analyze the source code but the overall execution time is slower.	✓ It takes large amount of time to analyze the sourcecode but the overall execution time is comparatively faster.
✓ Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.	✓ It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.
Example Programming language like Python, Ruby use interpreters.	Example Programming language like C, C++ use compilers.

- 9 Write python code to convert the time given in minutes into hours and minutes. 3

Ans:

```
min=int(input("Enter time in minutes"))
h=min//60
m=min%60
print("Hours=",h)
print("Minutes",m)
```

- 10 (i)What is empty statement? Which python statement can be termed as empty statement? 4

Ans: A statement that does nothing. Where the syntax of the language require the presence of statement but the logic of the program does not. Pass statement is an empty statement.

1 mark for correct answer 1 mark for pass statement

(ii) Re-write the following program using 'while' loop

```
sum = 0
for x in range(10):
    sum = sum + x**2
print(sum)
```

Ans:

```
sum=0
x=0
while x<10:
    sum=sum+x**2
print(sum)
```

- 11 Evaluate (to true or false)each of the following expression: 1
- 14<=14 14<14 -14 > -15 -15 >=15
1. True, False, True, False

2. False, False , True, True
3. True, False, True, True
4. False, True, False, False

Ans: True, False, True, True

- 12 Which of the following statements is not true? 1
1. Python is a high level programming language.
 2. Python is an interpreted language.
 3. Python is an object-oriented language.
 4. None of the above.

Ans: None of the above

- 13 Which of the following is correct? 1
1. Variable name can start with an underscore.
 2. Variable name can start with a digit.
 3. Keywords can be used as a variable name.
 4. Variable name can have symbols like: @, #, \$ etc.

Ans: Variable name can start with an underscore.

- 14 What will be the output of the following python statements? 2
- ```
a,b,c = 10,40,20
a,c,b = b+10, a+20, c-10
print(a,b,c)
print(a+b//c**2)
```
- Ans: 50 10 30  
50

- 15 What will be the output of the following: 3
- ```
m,n,p=40,20,0
while m>n:
    m = m
    % 7
    n = n//2
    p +=
    m+n
print(m,n,p)
```

Ans: 5 10 15

- 16 What will be the output of the following: 3
- ```
x , y, sum= 1, 5, 10
for i in
 range(1,y+1,2):
 sum = sum + x
 x=x+1
print(sum)
```

Ans: 16



Ans:

```
age=int(input("Enter age"))
if age>=18:
 print("Eligible for Voting")
else:
 print("A Minor")
```

- 24 Write python code to calculate Division, Percentage and total marks of 5 subjects of any student, assume that maximum marks for each subject is 100. And division as per criteria given below

3

| Percentage   | Division |
|--------------|----------|
| >=60         | first    |
| <60 and >=45 | second   |
| <45 and >=33 | third    |
| <33          | fail     |

Ans:

```
sub1=int(input("Enter first "))
sub2=int(input("Enter second "))
sub3=int(input("Enter third "))
sub4=int(input("Enter fourth "))
sub5=int(input("Enter fifth "))
totalmarks=sub1+sub2+sub3+sub4+sub5
percentage=totalmarks/5
if percentage>=60:
 print("First Division")
elif percentage>=45:
 print("Second Division")
elif percentage>=33:
 print("Third Division")
else:
 print("Fail")
```

- 25 Write a python program using loop to print the following series up to n terms.

4

1, 2, 3, 5, 8, 13, 21, .....

Ans.

```
n=int(input("enter term value"))
a,b=0,1
for i in range(n):
 a,b=b,a+b
 print(b)
```

- 25 What are Python's built-in core data types?

1

Ans: int, float, string, bool, list, tuple, etc

- 27 What will be the output of the following python statements? 1
1.  $87//5$  2.  $87//5.0$
- Ans: 1. 17  
2. 17.0
- 28 What is the use of "continue" keyword in python language. 1
- Ans: *to continue a loop from beginning*
- 29 Write the following arithmetic expressions using operators in Python: 2
- (i)  $c = \frac{a+b}{2a}$  (ii)  $x = a^3 + b^3$
- (iii)  $A = \pi r(r+h)^2$  (iv)  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
- Ans:
1.  $c = (a + b) / (2 * a)$   
2.  $x = a^{**3} + b^{**3}$   
3.  $A = \text{math.pi} * r (r + h) ** 2$  or  $A = 3.14 * r (r + h) ** 2$   
4.  $x = (-b + \text{math.sqrt}(b*b - 4 * a * c)) / (2*a)$
- 30 What will the following code print? 2
- (i) 

```
str1=""Hello
o""
str2=""Hello\
o""
print(len(str1) > len(str2))
```
- Ans: True
- (ii) 

```
x,y=10, '10'
print(type(x)==type(y))
```
- Ans: False
- 31 Write a python program to check whether an input number is prime or not. 3
- Ans:
- ```
n=int(input("Enter any number"))
for i in range(2,n):
    if n%i==0:
        print("Given number is not a prime number")
        break
else:
    print("Given number is a prime number")
```

32 What will be the output of the following: 1

```
s = 'PYTHON'
s1= s[::-1]
print(s1)
```

Ans
NOHTYP

33 i) 1

```
list1 = [12,32,65,26,80,10]
list1.sort()
print(list1)
```

Ans. [
10, 12, 26, 32, 65, 80]

34 list1 = [1,2,3,4,5,6,7,8,9,10] 1

```
list1[: -2]
print(list1[ : 3] + list1[ 3 : ])
```

Ans.
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

35 list1 = [1,2,3,4,5] 1

```
print( list1 [ len ( list1 ) - 1 ] )
```

Ans. 5

36 Write a Python program to arrange the values in a dictionary in ascending order. For example

Original Dictionary = {1 : 25, 2 : 21, 3 : 23}

Expected Output = [21, 25, 32]

Ans. :

```
d={1:21,2:32,3:25}
print(sorted(d.values()))
print(d)
```

37 Write a function check(key) which takes a key as an argument and check whether that key is present in dictionary or not.

Ans :

```
d1 = {1 : "Amit", 2 : "Suman"}
```

```
def check(i):
    for k in d1:
```

```

    if k==i:
        print("key is present")
        break
    else:
        print("key is not present")
check(1) #This function is only to verify code

```

- 38 Write a program to accept a key from the user and remove that key from the dictionary if present.

```

d1 = {1 : 2, 2 : 90, 3 : 50}
k=int(input("Enter any key"))
if k in d1:
    d1.pop(k)
    print(d1)
else:
    print("Key not found")

```

- 39 What are the three important functions of dictionary elements (any two) 2

```

Keys()
Items()
Values()

```

- 40 Q1. Keys must be _____ in dictionary.(mutable/immutable) 1
 Ans Immutable

- 41 Q3. Write the code to print only keys of the following dictionary. 1

```

D = {"Amit" : 40, "Sunil" : 34, "Naina" : 30}
print(d.keys())

```

- 42 Q7. Write two differences between List and Dictionary. 2

Dictionary has key value pair, but list has index value pairs
 Dictionary key can be of any type but list key are pre-defined and only numeric values

- 43 Write the output of the following: 2

```

m = {1 : "Jan", 2 : 'Jun', 3 : 'Aug'}
print(m[1])
print(m["jan"])

```

Ans :

Jun

Error

44 Which of the following is a dictionary? 1

1. L1 = [34, 56, 64, 32]
2. L2 = (23, 34, 65)
3. L3 = {1 : 23, 2 : 45}

Ans : L3

45 Write two important features of dictionary. 2

Ans Two important features of dictionary are:

1. Dictionary is mutable.
2. Keys are always unique while values may not be.

46 Write the output of the following. 2

```
D = { }  
D[1] = "Amit"  
D[2] = "Suman"  
print(D)
```

Ans

```
{1: 'Amit', 2: 'Sumit'}
```

47 Write the output of the following in reference to the following dictionary 2

```
d={1:"Cricket", 2 : "Hockey",3 : "Football"}
```

- a.
for i in d:
print(i)
- b.
for i in d:
print(d[i])

Ans :

a.

```
1  
2  
3
```

b.

```
Cricket  
Hockey  
Football
```


- 48 Write a program to accept roll numbers and name of three students from the user, store the detail in dictionary by taking roll number as key and name as primary key. 3

Ans

```
d={ }
for i in range(3):
    rn=int(input("Enter roll number"))
    nm=input("Enter Name")
    d[rn]=nm
print(d)
```

- 49 Explain the following in reference to dictionary, with example 3
1. del statement
 2. pop() method

Ans

Del statement will delete an item from the dictionary by specifying its key. For example

```
d={1: 'Amit', 2: 'Sumit', 5: 'Kavita'}
del d[5]
print(d)
```

Output :

```
{1: 'Amit', 2: 'Sumit'}
```

pop() : This method not only delete the item from the dictionary but also return the deleted value.

```
d={1: 'Amit', 2: 'Sumit', 5: 'Kavita'}
print(d.pop(1))
print(d)
```

Output:

```
Amit
{2: 'Sumit', 5: 'Kavita'}
```

- 50 What do you mean by dictionary in python? 1

Ans

A dictionary is an unordered collection of items and each item is a pair of Key and Value.

MYSQL – SQL

DATABASE CONCEPTS:

- A database management system (DBMS) or database system in short, is a software that can be used to create and manage database
- DBMS allows to create a database, store, manage, update/modify and retrieve data from that database by users or application programs
- Examples of open source and commercial DBMS include MySQL, Oracle, PostgreSQL, SQL Server, Microsoft Access, MongoDB.
- The DBMS serves as an interface between the database and end users or application programs
- Real life applications of DBMS : Banking system, Inventory Management, Organization resource management, Online shopping etc.

KEY CONCEPTS IN DBMS:

- **Database Schema** : It is the skeleton of the database that represents the structure, the type of data each column can hold, constraints on the data to be stored (if any), and the relationships among the tables.
- **Data Constraint** : Sometimes we put certain restrictions or limitations on the type of data that can be inserted in one or more columns of a table. This is done by specifying one or more constraints on that column(s) while creating the tables. Eg: Not Null, Unique etc.
- **Meta-data or Data Dictionary** : The database schema along with various constraints on the data is stored by DBMS in a database catalog or dictionary, called meta-data. A meta-data is data about the data
- **Database Instance** : loading data, the state or snapshot of the database at any given time is the database instance, data can be accessed through it
- **Query** : A query is a request to a database for obtaining information in a desired way
- **Data Manipulation** : Modification of database consists of three operations viz. Insertion, Deletion or Update
- **Database Engine** : Database engine is the underlying component or set of programs used by a DBMS to create database and handle various queries for data retrieval and manipulation

RELATIONAL DATA MODEL:

- A data model describes the structure of the database, including how data are defined and represented, relationships among data, and the constraints. The most commonly used data model is Relational Data Model
- In relational model, tables are called relations that store data for different columns. Each table can have multiple columns where each column name should be unique. For example, each row in the table represents a related set of values

- ❖ **Attribute** : columns of a relation
- ❖ **Tuple** : rows of a relation
- ❖ **Domain** : It is a set of values from which an attribute can take a value in each row
- ❖ **Degree** : The number of attributes in a relation is called the Degree of the relation
- ❖ **Cardinality** : The number of tuples in a relation is called the Cardinality of the relation

KEYS IN A RELATIONAL DATABASE:

- ❖ **Candidate Key** : A relation can have one or more attributes that takes distinct values. Any of these attributes can be used to uniquely identify the tuples in the relation. Such attributes are called candidate keys as each of them are candidates for the primary key
- ❖ **Primary Key** : the attribute chosen by the database designer to uniquely identify the tuples in a relation is called the primary key of that relation
- ❖ **Alternate Key**: All remaining candidate keys is known as alternate key
- ❖ **Foreign Key**: A foreign key is used to represent the relationship between two relations. A foreign key is an attribute whose value is derived from the primary key of another relation

NOTE : There are many RDBMS such as MySQL, Microsoft SQL Server, PostgreSQL, Oracle, etc.

STRUCTURED QUERY LANGUAGE (SQL):

- ❖ is the most popular query language used by major relational database management systems such as MySQL, ORACLE, SQL Server,
- ❖ SQL Rules:
 - SQL is case insensitive. That means name and NAME are same for SQL.
 - Always end SQL statements with a semicolon (;).

DATA TYPE OF ATTRIBUTE:

- Data type indicates the type of data value that an attribute can have and the operations that can be performed on the data of that attribute

CHAR	<ul style="list-style-type: none"> • Fixed length • Declaring CHAR (10) implies to reserve spaces for 10 characters. If data does not have 10 characters MySQL fills the remaining characters with spaces padded on the right.
-------------	--

VARCHAR	<ul style="list-style-type: none"> • Variable length • declaring VARCHAR (30) means a maximum of 30 characters can be stored but the actual allocated bytes will depend on the length of entered string
INT	Integer value. Each INT value occupies 4 bytes of storage
FLOAT	Holds numbers with decimal points. Each FLOAT value occupies 4 bytes.
DATE	The DATE type is used for dates in 'YYYY-MM-DD' format. YYYY is the 4 digit year, MM is the 2 digit month and DD is the 2 digit date

CONSTRAINTS:

- Constraints are certain types of restrictions on the data values that an attribute can have. They are used to ensure the accuracy and reliability of data.

SNO	CONSTRaining	Description
1	NOT NULL	Ensures that a column cannot have NULL value
2	DEFAULT	Provides a default value for a column when none is specified
3	UNIQUE	Ensures that all values in a column are different
4	CHECK	Makes sure that all values in a column satisfy certain criteria
5	PRIMARY KEY	Used to uniquely identify a row in the table
6	FOREIGN KEY	Used to ensure referential integrity of the data

- **SQL Commands Classification :**
 - Data Definition Language – CREATE, ALTER, DROP command
 - Data Manipulation Language – INSERT, UPDATE, DELETE, SELECT command
 - Transaction Control Language – COMMIT, ROLLBACK, SAVEPOINT command

- **SQL queries:**

DDL Commands			
1.	CREATE DATABASE	Syntax : CREATE DATABASE <DBNAME>;	Example : CREATE DATABASE EMP_DATA;
2.	For using /opening Database	Syntax : USE <DBNAME>;	Example : USE EMP_DATA;
3.	To show all available databases	Syntax : SHOW DATABASES;	Example : SHOW DATABASES;
4.	To create table	Syntax : CREATE TABLE <table_name>	Example : CREATE TABLE EMP (EMPNO INT ,

		(<code><col_name></code> <code><datatype></code> <code><size></code> constraint, <code><col_name></code> <code><datatype></code> <code><size></code> constraint,-----);	ENAME VARCHAR(20), SALARY INT); CREATE TABLE EMP (EMPNO INT PRIMARY KEY, ENAME VARCHAR(20), SALARY INT);																														
<div>Example 8.1 Create table STUDENT.</div> <pre>mysql> CREATE TABLE STUDENT(-> RollNumber INT, -> SName VARCHAR(20), -> SDateofBirth DATE, -> GUID CHAR(12), -> PRIMARY KEY (RollNumber)); Query OK, 0 rows affected (0.91 sec)</pre>																																	
5.	Describe table – can view the structure of the table	Syntax : DESCRIBE <code><TABLE_NAME>;</code>	Example : <pre>mysql> DESC STUDENT;</pre> <table><thead><tr><th>Field</th><th>Type</th><th>Null</th><th>Key</th><th>Default</th><th>Extra</th></tr></thead><tbody><tr><td>RollNumber</td><td>int</td><td>NO</td><td>PRI</td><td>NULL</td><td></td></tr><tr><td>SName</td><td>varchar(20)</td><td>YES</td><td></td><td>NULL</td><td></td></tr><tr><td>SDateofBirth</td><td>date</td><td>YES</td><td></td><td>NULL</td><td></td></tr><tr><td>GUID</td><td>char(12)</td><td>YES</td><td></td><td>NULL</td><td></td></tr></tbody></table> <p>4 rows in set (0.06 sec)</p>	Field	Type	Null	Key	Default	Extra	RollNumber	int	NO	PRI	NULL		SName	varchar(20)	YES		NULL		SDateofBirth	date	YES		NULL		GUID	char(12)	YES		NULL	
Field	Type	Null	Key	Default	Extra																												
RollNumber	int	NO	PRI	NULL																													
SName	varchar(20)	YES		NULL																													
SDateofBirth	date	YES		NULL																													
GUID	char(12)	YES		NULL																													
6.	Alter table – to change the structure of the table like Adding new column, deleting a column etc.	Syntax : ALTER TABLE <code><TABLE_NAME></code> ADD/MODIFY/DROP <code><COLUMN_NAME>...;</code>	Example : (a) Add an attribute to an existing table ALTER TABLE STUDENT ADD MARKS INT; (b) Modify datatype of an attribute ALTER TABLE STUDENT MODIFY MARKS FLOAT; (c) Remove an attribute ALTER TABLE STUDENT DROP MARKS;																														
7.	Drop table - DROP statement to remove a database or a table permanently from the system.	Syntax : DROP TABLE <code><table_name>;</code> Syntax to drop a database: DROP DATABASE <code>db_name;</code>	Example : Command to delete a database EMP_DATA. DROP DATABASE EMP_DATA; Command to delete a table STUDENT. DROP TABLE STUDENT;																														

DML Commands			
8.	Insertion of Record	Syntax : INSERT INTO <TABLE_NAME> (<col_name>,<col_name>,...) VALUES (<val1>,<val2>,...);	Example : INSERT INTO STUDENT VALUES (11, 'MEENU', '2005-10-25', 'ABC'); OR INSERT INTO STUDENT (ROLLNO,SNAME,SDOB,GUID) VALUES (11, 'MEENU', '2005-10-25', 'ABC'); OR INSERT INTO STUDENT (ROLLNO,SNAME,SDOB) VALUES (20, 'SHEENA', '2009-10-13');
9.	Select statement	1) To display all record Syntax: SELECT * FROM tablename; 2) To display selected columns and all rows Syntax: SELECT <colname>,<colname>.. FROM <tablename>; 3) To display selected rows and all columns Syntax : SELECT * FROM <tablename> WHERE <condition>; 4) To display selected rows and selected columns Syntax : SELECT <colname>,<colname> > FROM <tablename> WHERE <condition>;	To display all details of students. Example : SELECT * FROM STUDENT; To display only ROLLNO and SNAME from STUDENT table. SELECT ROLLNO, SNAME FROM STUDENT; To display all details whose rollno is less than 15. SELECT * FROM STUDENT WHERE ROLLNO<15; To display name of student whose roll no is 20. SELECT SNAME FROM STUDENT WHERE ROLLNO=20;
10.	Column Aliases	Syntax:	Example: Display names of

	<ul style="list-style-type: none">- renaming a column	<pre>SELECT <COLNAME> AS "<new name> FROM <tablename>;</pre>	<p>all employees along with their annual salary (Salary*12). While displaying query result, rename EName as Name.</p> <p>mysql> SELECT EName AS Name, Salary*12 FROM EMPLOYEE;</p> <p>Output:-</p> <table><thead><tr><th>Name</th><th>Salary*12</th></tr></thead><tbody><tr><td>Aaliya</td><td>120000</td></tr><tr><td>Kritika</td><td>720000</td></tr><tr><td>Shabbir</td><td>540000</td></tr><tr><td>Gurpreet</td><td>228000</td></tr><tr><td>Joseph</td><td>408000</td></tr><tr><td>Sanya</td><td>576000</td></tr><tr><td>Vergese</td><td>180000</td></tr><tr><td>Nachaobi</td><td>348000</td></tr><tr><td>Daribha</td><td>504000</td></tr><tr><td>Tanya</td><td>600000</td></tr></tbody></table> <p>10 rows in set (0.02 sec)</p>	Name	Salary*12	Aaliya	120000	Kritika	720000	Shabbir	540000	Gurpreet	228000	Joseph	408000	Sanya	576000	Vergese	180000	Nachaobi	348000	Daribha	504000	Tanya	600000
Name	Salary*12																								
Aaliya	120000																								
Kritika	720000																								
Shabbir	540000																								
Gurpreet	228000																								
Joseph	408000																								
Sanya	576000																								
Vergese	180000																								
Nachaobi	348000																								
Daribha	504000																								
Tanya	600000																								
11.	<p>DISTINCT clause - The SELECT statement when combined with DISTINCT clause, returns records without repetition (distinct records)</p>	<p>Syntax :</p> <pre>SELECT DISTINCT <COLNAME> FROM <tablename>;</pre>	<p>To display unique department number for all the employees.</p> <p>mysql> SELECT DISTINCT DeptId FROM EMPLOYEE;</p> <p>Output:-</p> <table><thead><tr><th>DeptId</th></tr></thead><tbody><tr><td>D02</td></tr><tr><td>D01</td></tr><tr><td>D04</td></tr><tr><td>D03</td></tr><tr><td>D05</td></tr></tbody></table>	DeptId	D02	D01	D04	D03	D05																
DeptId																									
D02																									
D01																									
D04																									
D03																									
D05																									
12.	<p>WHERE clause - The WHERE clause is used to retrieve data that meet some specified conditions</p>	<p>Syntax :</p> <pre>SELECT * FROM <tablename> WHERE <condition>;</pre>	<p>To display distinct salaries of the employees working in the department number D01.</p> <p>mysql> SELECT DISTINCT Salary FROM EMPLOYEE WHERE Deptid='D01';</p> <p>Output:-</p> <table><thead><tr><th>Salary</th></tr></thead><tbody><tr><td>60000</td></tr><tr><td>45000</td></tr><tr><td>15000</td></tr></tbody></table> <p>3 rows in set (0.02 sec)</p>	Salary	60000	45000	15000																		
Salary																									
60000																									
45000																									
15000																									

13.	Membership Operator – IN The IN operator compares a value with a set of values and returns true if the value belongs to that set.	Syntax: SELECT * FROM <tablename> WHERE <colname> IN (<val1>,<val2>...);	<pre>mysql> SELECT * -> FROM EMPLOYEE -> WHERE DeptId IN ('D01', 'D02' , 'D04');</pre> <table><thead><tr><th>EmpNo</th><th>Ename</th><th>Salary</th><th>Bonus</th><th>DeptId</th></tr></thead><tbody><tr><td>101</td><td>Aaliya</td><td>10000</td><td>234</td><td>D02</td></tr><tr><td>102</td><td>Kritika</td><td>60000</td><td>123</td><td>D01</td></tr><tr><td>103</td><td>Shabbir</td><td>45000</td><td>566</td><td>D01</td></tr><tr><td>104</td><td>Gurpreet</td><td>19000</td><td>565</td><td>D04</td></tr><tr><td>106</td><td>Sanya</td><td>48000</td><td>695</td><td>D02</td></tr><tr><td>107</td><td>Vergese</td><td>15000</td><td>NULL</td><td>D01</td></tr><tr><td>109</td><td>Daribha</td><td>42000</td><td>NULL</td><td>D04</td></tr></tbody></table> <p>7 rows in set (0.00 sec)</p>	EmpNo	Ename	Salary	Bonus	DeptId	101	Aaliya	10000	234	D02	102	Kritika	60000	123	D01	103	Shabbir	45000	566	D01	104	Gurpreet	19000	565	D04	106	Sanya	48000	695	D02	107	Vergese	15000	NULL	D01	109	Daribha	42000	NULL	D04																																																											
EmpNo	Ename	Salary	Bonus	DeptId																																																																																																		
101	Aaliya	10000	234	D02																																																																																																		
102	Kritika	60000	123	D01																																																																																																		
103	Shabbir	45000	566	D01																																																																																																		
104	Gurpreet	19000	565	D04																																																																																																		
106	Sanya	48000	695	D02																																																																																																		
107	Vergese	15000	NULL	D01																																																																																																		
109	Daribha	42000	NULL	D04																																																																																																		
14.	ORDER BY clause – used to display data in an ordered (arranged) form with respect to a specified column. By default displays records in ascending order To display the records in descending order, the DESC keyword written	Syntax : SELECT <colname>... FROM <tablename> ORDER BY <colname>;	Example : SELECT * FROM EMP ORDER BY SALARY; Output:- <table><thead><tr><th>EmpNo</th><th>Ename</th><th>Salary</th><th>Bonus</th><th>Dept</th></tr></thead><tbody><tr><td>101</td><td>Aaliya</td><td>10000</td><td>234</td><td>D02</td></tr><tr><td>107</td><td>Vergese</td><td>15000</td><td>NULL</td><td>D01</td></tr><tr><td>104</td><td>Gurpreet</td><td>19000</td><td>565</td><td>D04</td></tr><tr><td>108</td><td>Nachaobi</td><td>29000</td><td>NULL</td><td>D05</td></tr><tr><td>105</td><td>Joseph</td><td>34000</td><td>875</td><td>D03</td></tr><tr><td>109</td><td>Daribha</td><td>42000</td><td>NULL</td><td>D04</td></tr><tr><td>103</td><td>Shabbir</td><td>45000</td><td>566</td><td>D01</td></tr><tr><td>106</td><td>Sanya</td><td>48000</td><td>695</td><td>D02</td></tr><tr><td>110</td><td>Tanya</td><td>50000</td><td>467</td><td>D05</td></tr><tr><td>102</td><td>Kritika</td><td>60000</td><td>123</td><td>D01</td></tr></tbody></table> <p>10 rows in set (0.05 sec)</p> The following query displays details of all the employees in descending order of their salaries. mysql> SELECT * FROM EMPLOYEE ORDER BY Salary DESC; Output:- <table><thead><tr><th>EmpNo</th><th>Ename</th><th>Salary</th><th>Bonus</th></tr></thead><tbody><tr><td>102</td><td>Kritika</td><td>60000</td><td>123</td></tr><tr><td>110</td><td>Tanya</td><td>50000</td><td>467</td></tr><tr><td>106</td><td>Sanya</td><td>48000</td><td>695</td></tr><tr><td>103</td><td>Shabbir</td><td>45000</td><td>566</td></tr><tr><td>109</td><td>Daribha</td><td>42000</td><td>NULL</td></tr><tr><td>105</td><td>Joseph</td><td>34000</td><td>875</td></tr><tr><td>108</td><td>Nachaobi</td><td>29000</td><td>NULL</td></tr><tr><td>104</td><td>Gurpreet</td><td>19000</td><td>565</td></tr><tr><td>107</td><td>Vergese</td><td>15000</td><td>NULL</td></tr><tr><td>101</td><td>Aaliya</td><td>10000</td><td>234</td></tr></tbody></table> <p>10 rows in set (0.00 sec)</p>	EmpNo	Ename	Salary	Bonus	Dept	101	Aaliya	10000	234	D02	107	Vergese	15000	NULL	D01	104	Gurpreet	19000	565	D04	108	Nachaobi	29000	NULL	D05	105	Joseph	34000	875	D03	109	Daribha	42000	NULL	D04	103	Shabbir	45000	566	D01	106	Sanya	48000	695	D02	110	Tanya	50000	467	D05	102	Kritika	60000	123	D01	EmpNo	Ename	Salary	Bonus	102	Kritika	60000	123	110	Tanya	50000	467	106	Sanya	48000	695	103	Shabbir	45000	566	109	Daribha	42000	NULL	105	Joseph	34000	875	108	Nachaobi	29000	NULL	104	Gurpreet	19000	565	107	Vergese	15000	NULL	101	Aaliya	10000	234
EmpNo	Ename	Salary	Bonus	Dept																																																																																																		
101	Aaliya	10000	234	D02																																																																																																		
107	Vergese	15000	NULL	D01																																																																																																		
104	Gurpreet	19000	565	D04																																																																																																		
108	Nachaobi	29000	NULL	D05																																																																																																		
105	Joseph	34000	875	D03																																																																																																		
109	Daribha	42000	NULL	D04																																																																																																		
103	Shabbir	45000	566	D01																																																																																																		
106	Sanya	48000	695	D02																																																																																																		
110	Tanya	50000	467	D05																																																																																																		
102	Kritika	60000	123	D01																																																																																																		
EmpNo	Ename	Salary	Bonus																																																																																																			
102	Kritika	60000	123																																																																																																			
110	Tanya	50000	467																																																																																																			
106	Sanya	48000	695																																																																																																			
103	Shabbir	45000	566																																																																																																			
109	Daribha	42000	NULL																																																																																																			
105	Joseph	34000	875																																																																																																			
108	Nachaobi	29000	NULL																																																																																																			
104	Gurpreet	19000	565																																																																																																			
107	Vergese	15000	NULL																																																																																																			
101	Aaliya	10000	234																																																																																																			
15.	Handling NULL	Syntax :	➤ Displays details of all																																																																																																			

	Values	<p>SELECT * FROM <tablename> WHERE <colname> IS NULL/ IS NOT NULL;</p>	<p>those employees who have not been given a bonus. This implies that the bonus column will be blank.</p> <p>mysql> SELECT * FROM EMPLOYEE WHERE Bonus IS NULL;</p> <p>Output:-</p> <table><tr><th>EmpNo</th><th>Ename</th><th>Salary</th><th>Bonus</th><th>DeptId</th></tr><tr><td>107</td><td>Vergese</td><td>15000</td><td>NULL</td><td>D01</td></tr><tr><td>108</td><td>Nachaobi</td><td>29000</td><td>NULL</td><td>D05</td></tr><tr><td>109</td><td>Daribha</td><td>42000</td><td>NULL</td><td>D04</td></tr></table> <p>3 rows in set (0.00 sec)</p> <p>➤ Displays names of all the employees who have been given a bonus. This implies that the bonus column will not be blank.</p> <p>mysql> SELECT EName FROM EMPLOYEE WHERE Bonus IS NOT NULL;</p> <p>Output:-</p> <table><tr><th>EName</th></tr><tr><td>Aaliya</td></tr><tr><td>Kritika</td></tr><tr><td>Shabbir</td></tr><tr><td>Gurpreet</td></tr><tr><td>Joseph</td></tr><tr><td>Sanya</td></tr><tr><td>Tanya</td></tr></table> <p>7 rows in set (0.00 sec)</p>	EmpNo	Ename	Salary	Bonus	DeptId	107	Vergese	15000	NULL	D01	108	Nachaobi	29000	NULL	D05	109	Daribha	42000	NULL	D04	EName	Aaliya	Kritika	Shabbir	Gurpreet	Joseph	Sanya	Tanya
EmpNo	Ename	Salary	Bonus	DeptId																											
107	Vergese	15000	NULL	D01																											
108	Nachaobi	29000	NULL	D05																											
109	Daribha	42000	NULL	D04																											
EName																															
Aaliya																															
Kritika																															
Shabbir																															
Gurpreet																															
Joseph																															
Sanya																															
Tanya																															
16.	Substring pattern matching	<ul style="list-style-type: none">● % (percent)— used to represent zero, one, or multiple characters● _ (underscore)— used to represent a single char	<p>➤ Displays details of all those employees whose name starts with 'K'.</p> <p>mysql> SELECT * FROM EMPLOYEE WHERE Ename LIKE 'K%';</p> <p>➤ Displays details of all</p>																												

			<p>those employees whose name ends with 'a'.</p> <pre>mysql> SELECT * -> FROM EMPLOYEE -> WHERE Ename LIKE '%a';</pre> <p>➤ Displays details of all those employees whose name consists of exactly 5 letters and starts with any letter but has 'ANYA' after that.</p> <pre>mysql> SELECT * FROM EMPLOYEE WHERE Ename LIKE '_ANYA';</pre> <p>➤ Displays names of all the employees containing 'se' as a substring in name.</p> <pre>mysql> SELECT Ename FROM EMPLOYEE WHERE Ename LIKE '%se%';</pre> <p>➤ Displays names of all employees containing 'a' as the second character.</p> <pre>mysql> SELECT Ename FROM EMPLOYEE WHERE Ename LIKE '_a%';</pre>
17.	<p>Data Updation – to make changes in the value(s) of one or more columns of existing records in a table.</p>	<p>Syntax: UPDATE table_name SET attribute1 = value1, attribute2 = value2, .. WHERE condition;</p>	<p>➤ To update the name of student to 'SANJAY' whose Roll no is 3 in table STUDENT.</p> <pre>mysql> UPDATE STUDENT SET SNAME = 'SANJAY' WHERE RollNumber = 3;</pre>
18.	<p>Data Deletion - The DELETE statement is used to delete one or more record(s) from a table</p>	<p>Syntax: DELETE FROM table_name WHERE condition;</p>	<p>➤ To delete the details of student whose Roll number is 2.</p> <pre>mysql> DELETE FROM STUDENT WHERE RollNumber = 2;</pre>

MYSQL – QUESTION BANK

Write SQL queries to perform the following task:

1.	To create database with name BOOK.	CREATE DATABASE BOOK;																																
2.	To Open above created database BOOK.	USE BOOK;																																
3.	To display list of all available databases.	SHOW DATABASES;																																
4.	To create following table: Table Name: EMP <table><tr><td>Column name</td><td>Datatype</td><td>Size</td><td>Constraint</td></tr><tr><td>EMPNO</td><td>INT</td><td>10</td><td>PRIMARY KEY</td></tr><tr><td>ENAME</td><td>VARCHAR</td><td>25</td><td>NOT NULL</td></tr><tr><td>JOB</td><td>VARCHAR</td><td>25</td><td>-</td></tr><tr><td>SALARY</td><td>INT</td><td>10</td><td>-</td></tr><tr><td>DOJ</td><td>DATE</td><td></td><td>-</td></tr><tr><td>DEPTNO</td><td>INT</td><td>5</td><td>-</td></tr></table>	Column name	Datatype	Size	Constraint	EMPNO	INT	10	PRIMARY KEY	ENAME	VARCHAR	25	NOT NULL	JOB	VARCHAR	25	-	SALARY	INT	10	-	DOJ	DATE		-	DEPTNO	INT	5	-	CREATE TABLE EMP (EMPNO INT (10) PRIMARY KEY, ENAME VARCHAR (25) NOT NULL, JOB VARCHAR (25), SALARY INT (10), DOJ DATE, DEPTNO INT (5));				
Column name	Datatype	Size	Constraint																															
EMPNO	INT	10	PRIMARY KEY																															
ENAME	VARCHAR	25	NOT NULL																															
JOB	VARCHAR	25	-																															
SALARY	INT	10	-																															
DOJ	DATE		-																															
DEPTNO	INT	5	-																															
<table><tr><td colspan="2"></td><td colspan="2">Table : GYM</td></tr><tr><td>ICODE</td><td>INAME</td><td>PRICE</td><td>BRANDNAME</td></tr><tr><td>G101</td><td>Power fit exerciser</td><td>20000</td><td>ABC</td></tr><tr><td>G102</td><td>Hand Grip</td><td>8000</td><td>ROMA</td></tr><tr><td>G103</td><td>Cycle bike</td><td>25000</td><td>ABC</td></tr><tr><td>G104</td><td>Cross trainer</td><td>4100</td><td>YES</td></tr><tr><td>G105</td><td>Massage belt</td><td>5600</td><td>ROMA</td></tr><tr><td>G106</td><td>Treadmill</td><td>52000</td><td>RUNFLOW</td></tr></table>					Table : GYM		ICODE	INAME	PRICE	BRANDNAME	G101	Power fit exerciser	20000	ABC	G102	Hand Grip	8000	ROMA	G103	Cycle bike	25000	ABC	G104	Cross trainer	4100	YES	G105	Massage belt	5600	ROMA	G106	Treadmill	52000	RUNFLOW
		Table : GYM																																
ICODE	INAME	PRICE	BRANDNAME																															
G101	Power fit exerciser	20000	ABC																															
G102	Hand Grip	8000	ROMA																															
G103	Cycle bike	25000	ABC																															
G104	Cross trainer	4100	YES																															
G105	Massage belt	5600	ROMA																															
G106	Treadmill	52000	RUNFLOW																															
5.	Insert the rows in above table GYM.	INSERT INTO GYM VALUES ('G101', 'Power fit exerciser' , 20000 , 'ABC');																																

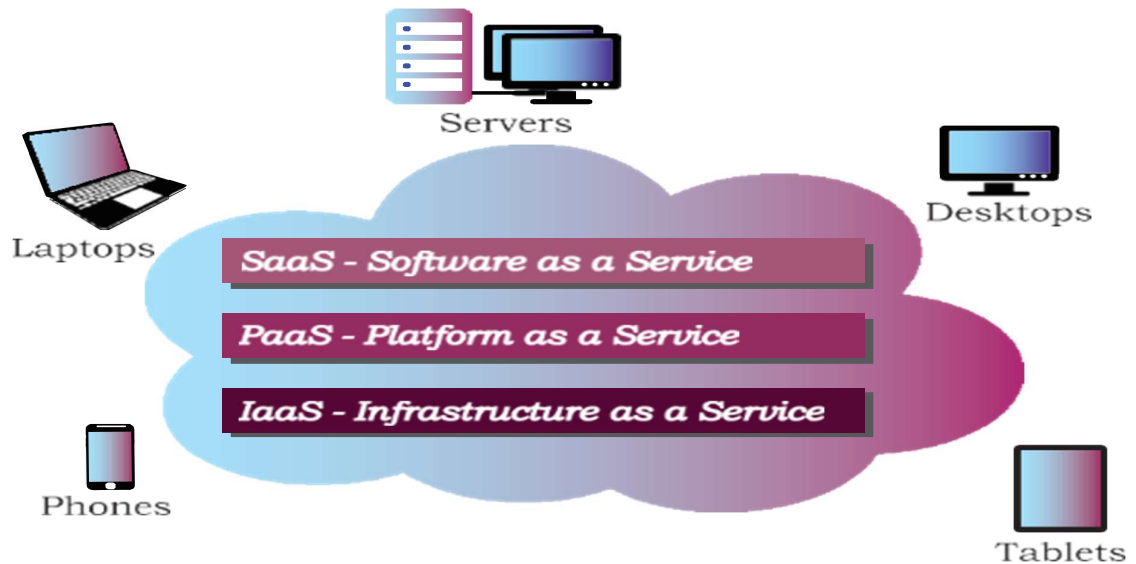
Write SQL query for the following on the basis of Table EMP given below:

ENAME	DNAME	JOB	EMPNO	HIREDATE	LOC
ADAMS	RESEARCH	CLERK	7876	23-MAY-87	DALLAS
ALLEN	SALES	SALESMAN	7499	20-FEB-81	CHICAGO
BLAKE	SALES	MANAGER	7698	01-MAY-81	CHICAGO
CLARK	ACCOUNTING	MANAGER	7782	09-JUN-81	NEW YORK
FORD	RESEARCH	ANALYST	7902	03-DEC-81	DALLAS
JAMES	SALES	CLERK	7900	03-DEC-81	CHICAGO
JONES	RESEARCH	MANAGER	7566	02-APR-81	DALLAS
KING	ACCOUNTING	PRESIDENT	7839	17-NOV-81	NEW YORK
MARTIN	SALES	SALESMAN	7654	28-SEP-81	CHICAGO
MILLER	ACCOUNTING	CLERK	7934	23-JAN-82	NEW YORK
SCOTT	RESEARCH	ANALYST	7788	19-APR-87	DALLAS
SMITH	RESEARCH	CLERK	7369	17-DEC-80	DALLAS
TURNER	SALES	SALESMAN	7844	08-SEP-81	CHICAGO
WARD	SALES	SALESMAN	7521	22-FEB-81	CHICAGO

6.	Write a command to display all details of table EMP.	SELECT * FROM EMP;
----	--	--------------------

7.	Write a command to display name of employee with their job type.	SELECT ENAME, JOB FROM EMP;
8.	Write a command to display all details of table EMP whose JOB type is CLERK.	SELECT * FROM EMP WHERE JOB='CLERK';
9.	Write a command to display name of employees whose loc is CHICAGO.	SELECT ENAME FROM EMP WHERE LOC='CHICAGO';
10.	Write a command to display details of employee whose job type is either CLERK or SALESMAN.	SELECT * FROM EMP WHERE JOB IN ('CLERK', 'SALESMAN');
11.	Write a command to display distinct job types from table emp.	SELECT DISTINCT JOB FROM EMP;
12.	Write a command to display details of Employee in ascending order of HIREDATE.	SELECT * FROM EMP ORDER BY HIREDATE;
13.	Write a command to display name of employees whose Department name is not given.	SELECT ENAME FROM EMP WHERE DNAME IS NULL;
14.	Write a command to display name of employees whose name starts with 's'.	SELECT ENAME FROM EMP WHERE ENAME LIKE 'S%';
15.	Write a command to display name of employees whose name ends with 's'.	SELECT ENAME FROM EMP WHERE ENAME LIKE '%s';
16.	Write a command to display name of employees whose name contains alphabet 's'.	SELECT ENAME FROM EMP WHERE ENAME LIKE '%S%';
17.	Write a command to display name of employees whose name contains 's' as second last alphabet.	SELECT ENAME FROM EMP WHERE ENAME LIKE '%s_';
18.	Write a command to display name of employees whose name length is 4 letters.	SELECT ENAME FROM EMP WHERE ENAME LIKE '____';
19.	Display details of emp in descending order of Location.	SELECT * FROM EMP ORDER BY LOC DESC;
20.	Write a command to change the job type to MANAGER whose empno is 7876.	UPDATE EMP SET JOB='MANAGER' WHERE EMPNO = 7876;
21.	Write a command to add a new column SALARY in above table emp.	ALTER TABLE EMP ADD SALARY INT;
22.	Write a command to modify the datatype of SALARY to float.	ALTER TABLE EMP MODIFY SALARY FLOAT;
23.	Write a command to delete a column salary from table emp.	ALTER TABLE EMP DROP SALARY;
24.	Write a command to delete a rows from table emp whose empno is either 7876 or 7902.	DELETE FROM EMP WHERE EMPNO IN (7876, 7902);
25.	Write a command to delete all the rows from table emp.	DELETE FROM EMP;
26.	Write a command to delete table emp.	DROP TABLE EMP;
27.	Write a command to delete database SCHOOL.	DROP DATABASE SCHOOL;
28.	Write the degree and cardinality of the above table emp.	Degree = no. of columns = 6 Cardinality = no of rows = 14

EMERGING TRENDS



ARTIFICIAL INTELLIGENCE

Artificial Intelligence endeavours to simulate the natural intelligence of human beings into machines, thus making them behave intelligently. An intelligent machine is supposed to imitate some of the cognitive functions of humans like learning, decision-making and problem solving. AI system can also learn from past experiences or outcomes to make new decisions.

MACHINE LEARNING

Machine Learning is a subsystem of Artificial Intelligence, wherein computers have the ability to learn from data using statistical techniques, without being explicitly programmed by a human being. It comprises algorithms that use data to learn on their own and make predictions. These algorithms, called models, are first trained and tested using a training data and testing data, respectively.

DEFINE AUGMENTED REALITY

The superimposition of computer-generated perceptual information over the existing physical surroundings is called as Augmented Reality (AR). It adds components of the digital world to the physical world.

BIGDATA

Big data is data that contains greater variety, arriving in increasing volumes and with more velocity. Our posts, instant messages and chats, photographs that we share through various sites, our tweets, blog articles, news items, opinion polls and their comments, audio/video chats etc. Big data not only represents voluminous data, it also involves various challenges like integration, storage, analysis, searching, processing, transfer, querying and visualisation of such data.

IOT THINGS

Internet of things (IoT), is an entity or physical object that has a unique identifier, an embedded system and the ability to transfer data over a network.

The Internet of Things refers to the rapidly growing network of connected objects that are able to collect and exchange data in real time using embedded sensors. **Thermostats**, cars, lights, refrigerators, and more appliances can all be connected to the IoT.

CLOUD COMPUTING

Cloud computing is an emerging trend in the field of information technology, where computer-based services are delivered over the Internet or the cloud. The services comprise software, hardware (servers), databases, storage, etc. These resources are provided by companies called cloud service providers. Cloud computing offers cost-effective, on-demand resources. A user can avail need-based resources from the cloud at a very reasonable cost. We already use cloud services while storing our pictures and files as backup on Internet, or host a website on the Internet.

CLOUD SERVICES AVAILABLE OVER NETWORK

There are three standard models to categorise different computing services delivered through cloud

- Infrastructure as a Service (IaaS),
- Platform as a Service (PaaS), and
- Software as a Service (SaaS).

INFRASTRUCTURE AS A SERVICE (IAAS)

The IaaS providers can offer different kinds of computing infrastructure, such as servers, virtual machines (VM), storage and backup facility, network components, operating systems or any other hardware or software. Using IaaS from the cloud, a user can use the hardware infrastructure located at a remote location to configure, deploy and execute any software application on that cloud infrastructure.

PLATFORM AS A SERVICE (PAAS)

Through this service, a user can install and execute an application without worrying about the underlying infrastructure and their setup. Suppose we have developed a web application using MySQL and Python. To run this application online, we can avail a pre-configured Apache server from cloud having MySQL and Python pre-installed. Thus, we are not required to install MySQL and Python on the cloud, nor do we need to configure the web server.

SOFTWARE AS A SERVICE (SAAS)

SaaS provides on-demand access to application software, usually requiring a licensing or subscription by the user. While using Google doc, Microsoft Office 365, Drop Box, etc., to edit a document online, we use SaaS from cloud. A user is not concerned about installation or configuration of the software application as long as the required software is accessible.

BLOCKCHAINS

We can define blockchain as a system that allows a group of connected computers to maintain a single updated and secure ledger. Each computer or node that participates in the blockchain receives a full copy of the database. It maintains an 'append only' open ledger which is updated only after all the nodes within the network authenticate the transaction.