SECTION A (15 Marks)

1. Multiple Choice Questions

i. B – The study of computers and their applications
ii. B – Printer
iii. B – CPU
iv. C – Printer
v. C – Operating System
vi. B – Fixing computer problems
vii. C – Input → Processing → Output → Storage
viii. B – Frequent backups
ix. C – Hardware, software, and users working together
x. B – Following algorithms and logical steps

2. Matching Items

i. Operating System → B (Main software controlling computer operations)
ii. Algorithm → A (Step-by-step instructions for solving a problem)
iii. Hardware → D (Physical components of a computer)
iv. Software → C (Non-physical parts of a computer)
v. Troubleshooting → E (Process of identifying and fixing problems)

SECTION B (70 Marks)

3. Computer Science Basics

(a) **Definition:** Computer Science is the study of computers, their components, software, and how to use them to solve problems efficiently.

(b) **Branches:**

1. **Software Engineering** – Developing software applications.
2. **Artificial Intelligence** – Designing machines that can simulate human intelligence.

(c) **Importance in Daily Life:**

1. Facilitates communication (emails, messaging, video calls).
2. Simplifies business operations (accounting, stock management).
3. Supports education (e-learning, simulations, research).
4. Aids in problem solving and decision making.

4. Computer System and Components

(a) **Definition:** A computer system is a combination of hardware, software, and users working together to process data into meaningful information.

(b) **Four Basic Components and Functions:**

1. **Input Devices** – Accept data from users (e.g., keyboard).
2. **Processing Unit (CPU)** – Processes and executes instructions.
3. **Output Devices** – Displays results of processing (e.g., monitor).
4. **Storage Devices** – Stores data permanently or temporarily (e.g., hard disk, USB).

5. Hardware vs Software

(a) **Differences:**

Hardware	Software
Physical and tangible components	Non-physical programs and instructions
Can be touched	Cannot be touched

Examples: Monitor, Printer	Examples: Microsoft Word, Operating System

(b) **Examples:**

- **Application software:** Microsoft Word, Excel
- **System software:** Windows OS, Linux

6. Troubleshooting

(a) **Definition:** Troubleshooting is the process of identifying, diagnosing, and fixing problems in a computer system.

(b) **Common Computer Problems:**

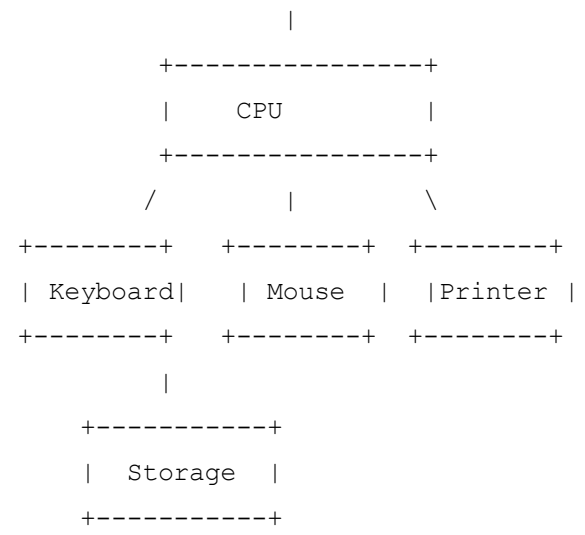
1. Computer not starting (power failure or hardware issue)
2. Software crashes or errors
3. Virus or malware infections
4. Slow system performance

7. Preventive Maintenance Practices

(a) **Practices:**

1. Regularly updating antivirus and software to prevent infections.
2. Cleaning hardware components (keyboard, monitor, CPU) to prevent dust accumulation.
3. Performing regular backups to prevent data loss.

(b) **Computer System Diagram:**



8. Algorithms

(a) **Definition:** An algorithm is a step-by-step procedure or set of instructions for solving a problem or performing a task.

(b) **Characteristics of a Good Algorithm:**

1. **Finiteness** – Must terminate after a finite number of steps.
2. **Definiteness** – Each step must be clear and unambiguous.
3. **Input** – Takes zero or more inputs.
4. **Output** – Produces at least one output.

SECTION C (15 Marks)

9. Importance of Proper Maintenance and Troubleshooting

- Ensures computers operate efficiently and reliably.
- Reduces the risk of hardware failure and data loss.
- Minimizes downtime in schools and offices.
- Extends the lifespan of computer equipment.

- Enhances user productivity by preventing unexpected system crashes.

10. Basic Principles of Computer Science

- **Algorithmic thinking** – Solving problems using step-by-step procedures.
- **Logical reasoning** – Making decisions based on logical analysis.
- **Abstraction** – Focusing on essential information while ignoring irrelevant details.
- **Modularity** – Dividing complex problems into smaller, manageable modules.

Application in Problem Solving and Programming:

- Algorithms guide program development.
- Logical reasoning ensures correct outputs.
- Abstraction simplifies complex programs.
- Modularity improves code maintenance and reusability.

✅ **END OF ANSWERS**