**Set One:**

1. Define a microcontroller and a microprocessor. List any four key differences between them.
2. What is an embedded system? List and explain any four of its characteristics.
3. State the advantages of Harvard architecture over Von Neumann architecture.
4. Describe the role of the compiler and assembler in embedded system development.
5. List the limitations of 8-bit microcontrollers.
6. Draw the block diagram of Von Neumann and Harvard architecture
7. List the pin's of RS-232 protocol with explanation

## Set Two:

1. Explain the functional differences between RISC and CISC architectures with suitable examples.
2. Discuss the role of microcontrollers in embedded systems with appropriate examples.
3. Compare Harvard and Von Neumann architectures in terms of speed and memory access.
4. Explain how SPI and I2C protocols differ in terms of hardware and communication mechanism.
5. Describe the importance of IDE and debugger in embedded system software development.

## Set Three:

1. Choose a real-world embedded application (e.g., washing machine or traffic light). Explain how a microcontroller-based system would control it.
2. Design a small system that uses RS232 communication to send data from a microcontroller to a PC. Briefly explain the interfacing and data flow.
3. Apply the knowledge of 8-bit microcontroller limitations to justify the use of 16-bit microcontrollers in industrial automation.

4. Using your understanding of SPI protocol, explain how to connect a microcontroller to an external EEPROM chip.
5. Identify and describe the software tools required to develop an LED blinking application on PIC microcontroller.

## Set Four:

1. Analyze the suitability of RISC architecture in embedded systems compared to CISC.
2. Compare RS232, RS485, and I2C protocols based on number of wires, speed, and range. Which is best for long-distance industrial communication and why?
3. Analyze how Harvard architecture contributes to increased performance in embedded systems.
4. Discuss how debugging tools (like hardware debugger or emulator) help in reducing errors during microcontroller program development.
5. Compare the usage of assembler and compiler in development of embedded firmware. When would you use each?

## Set Five:

1. Evaluate the choice of using an 8-bit microcontroller in a home automation project. Is it justified or would you recommend a higher-bit MCU? Explain.
2. Critically evaluate RS232 for modern embedded applications. Suggest a better alternative if required and justify your answer.
3. Judge the efficiency of Von Neumann architecture in real-time embedded applications.
4. Given a complex embedded application requiring multiple sensor inputs and memory access, assess whether RISC or CISC architecture is more suitable.
5. Evaluate the impact of using an integrated IDE with debugging support versus using separate tools for coding, compiling, and testing.

## Set Six:

1. Design a simple embedded system for automatic temperature monitoring using a microcontroller and suitable communication protocol (I2C/SPI). Briefly explain your architecture.
2. Create a comparison table of RS232, RS485, I2C, and SPI highlighting their key parameters (lines required, distance, speed, master/slave, full/half duplex).
3. Propose a microcontroller-based solution for monitoring and controlling classroom lights. Mention the role of embedded characteristics in your design.
4. Draft a development flow using tools like compiler, assembler, IDE, and emulator for a microcontroller-based motor speed controller system.
5. Design an experiment to test the data transmission accuracy of RS485 protocol in a noisy industrial setup using an embedded microcontroller.