Adv C Module

1. Subjective

2.1 Basic Refreshers

1. Explain the bitwise operation.
2. How are characters stored in C?

2.2 1D Pointers and Functions

1. Where are function arguments stored in memory?

2.3 String

1. What is the code segment in memory, and what does it contain during the execution of a program?
2. What are the inbuilt functions you have used from <string.h>?
3. What happens if a string in C does not end with a null character ('\0)?
4. How does the printf() function work in C, even though it accepts a variable number of arguments?

2.4 Storage classes and memory segments

1. Where are static variables stored in memory?
2. What is meant by storage class?
3. What is the difference between the static and extern keywords?

2.5 2D Pointers and DMA

1. What is a function pointer?
2. Explain recursion with an example.
3. Explain the callback function with an example program.
4. Explain Memory segment.
5. Explain the dynamic memory allocation.
6. What is the difference between malloc and calloc?

2.6 Preprocessing

1. Explain preprocessor and preprocessor directive.
2. Explain the compilation stages.

2.7 UDT

1. Explain the structure in detail.
2. What is the difference between structure and union?

2.8 Miscellaneous

1. Explain the volatile keyword.

2. Programming

1. Write a program to reverse the order of words in a given sentence.
2. Write a macro to check if a given number is a power of 2 or not.
3. Write a program to rotate a given 2D matrix by 180 degrees.
4. There are n cities labeled from 0 to n - 1. You are given an n x n matrix isConnected where isConnected[i][j] = 1 means the i-th city and the j-th city are directly connected, and isConnected[i][j] = 0 otherwise.
   A province is a group of directly or indirectly connected cities and no other cities outside of the group.
   Return the total number of provinces.
   Example:
   Input: isConnected = [[1,1,0],[1,1,0],[0,0,1]]
   Output: 2
   Explanation: There are two provinces: [0, 1] and [2]
5. Given an integer array nums, return the length of the longest strictly increasing subsequence.
   A subsequence is a sequence that can be derived from an array by deleting some or no elements without changing the order of the remaining elements.
   Example:
   Input: nums = [10, 9, 2, 5, 3, 7, 101, 18]
   Output: 4
   Explanation: The longest increasing subsequence is [2, 3, 7, 101]; therefore, the length is 4.

MC Module

1. Basic electronics

1. What type of electronic device is a transistor—active or passive?

2. Basics

1. What are the Required Connections for the PIC18F4580 Microcontroller?
2. How Do You Configure a Matrix Keypad?
3. Explain the GPIO pins.
4. What is a register in the context of microcontrollers or computer architecture?
5. Is the 8051 an 8-bit or 16-bit processor?

3. Interrupts

1. Which interrupt has the highest priority in the 8051 microcontroller?

4. Projects

1. Explain the car black box project.
2. Explain the digital timer project.

5. ADC

1. What is the difference between ADC and DAC?

6. Protocols

1. Explain the I2C protocol with a frame format.
2. Explain the CAN protocol with a frame format.
3. Explain the SPI protocol with a frame format.
4. Explain the UART protocol with a frame format.
5. What Are the SPI Protocol Pin Names?

7. PWM

1. What is Pulse Width Modulation (PWM), and how is it used to control devices such as motors or LEDs in embedded systems?
2. What is the difference between PWM frequency and PWM duty cycle?

8. Programming

1. How are LEDs and Seven Segment Displays (SSD) Connected to the Board?
2. Write a Program to Blink LEDs in Series.

CPP Module

1. What is the difference between C and c++.
2. What is meant by function overloading?
3. Explain Runtime Polymorphism with Example.
4. Explain V-Table (Virtual Table).
5. What is the difference between an interface and a class?

Ds Module

1. Basics

1. What is meant by data structure?

2. Linked lists

1. WAP to reverse the array using a linked list.
2. What is a linked list, and how does it differ from an array in terms of memory allocation and performance?

3. Stack

1. What is a stack in computer memory, and how does it work?
2. What is the difference between stack and heap memory?
3. Why is the stack referred to as a LIFO (Last In, First Out) structure?

4. Searching and Sorting Techniques

1. What is the difference between linear search and binary search in terms of time complexity and application?
2. How does binary search work, and what is the key condition for using it?
3. Write a program to perform a binary search on a sorted array.
4. Can you perform a binary search on a linked list? Why or why not?

5. Queue

1. What is a queue in data structures, and how does it differ from a stack?
2. What is the FIFO (First-In, First-Out) principle in queues?
3. List real-world examples of queue usage in computer systems.

6. Hashing

1. What is hashing, and how is it used in data structures?
2. What is a hash function, and what are the desirable properties of a good hash function?
3. What is a hash table, and how does it improve search efficiency compared to arrays or linked lists?

7. Trees

1. What is a tree in data structures, and how is it different from a graph?
2. Write a program to find the height or depth of a binary tree.

8. Programming

1. You are given a list of words, where each word has a corresponding price (an integer). Also, you are given a prefix string. Your task is to:
    - Find all words that start with the given prefix
    - Return them sorted in ascending order of their prices
    - Use a tree-based data structure (Trie) for efficient prefix searching

    Sample Input:

    5
    apple 100
    app 50
    applet 80
    bat 30
    banana 40
    app

    Sample Output:

    app
    applet
    apple

LI Module

1. Basics

1. What is the difference between a kernel and a shell in Linux?

2. System call

1. Explain the system call.
2. Why do we need system calls in an operating system?
3. How does a system call switch from user mode to kernel mode?

3. Networking

1. Explain the concept of network synchronization and the protocols used for it. Also, explain network management concepts and protocols.
2. What is an IP address, and why is network byte order important in socket programming?

4. Process

1. What is a process in an operating system, and what states does a process transition through during its lifecycle?
2. What is scheduling in operating systems, and what are the different types of CPU scheduling algorithms?

5. IPC

1. What are IPC (Inter-Process Communication) mechanisms? Can you explain them?

6. Signal

1. What is a signal in Linux? How is it used in process management?
2. What is the difference between a software interrupt and a signal?
3. List any five standard signals and their meanings (e.g., SIGINT, SIGTERM, SIGKILL).
4. How can a signal be sent to a process? Mention different commands and system calls.

7. Socket

1. What are the key differences between TCP and UDP, and where are each of them used in real-world applications?
2. What is a socket file descriptor, and how is it used for network communication between processes or systems?

8. Threads

1. What are threads, and how do they work?
2. How to pass arguments to threads.
3. Explain the spinlock, deadlock, and race conditions.
4. What is the difference between mutexes and semaphores?
5. What is the difference between threads and processes?
6. What problems can occur when multiple threads access the same shared resource concurrently, and what synchronization techniques can be used to prevent them?
7. What are Pthreads, and which system calls are commonly used to create and manage threads in POSIX systems?

9. Process and Memory Management

1. What is memory management in an OS and why is it important?
2. What is paging, and how does it work?
3. What is segmentation in memory management?

General Questions
Aptitude

1. What is meant by routers and router protocols?
2. Draw a Circuit Diagram to Convert 230V AC into 6V DC.
3. What are operational amplifiers (op-amps), and what are their primary applications?
4. What are the characteristics of an op-amp when it is in an idle (no input) condition?
5. Which of the following equations follows the rules of Boolean Algebra?

6. The N-Queens problem is to place N queens on an N x N chessboard such that:
    - No two queens attack each other
    - That means no two queens share the same row, column, or diagonal

   Your task is to return the total number of distinct valid arrangements of the N queens.

   Sample Input:

   4

   Sample Output:

   2
7. Capeta is working part-time for an animal shipping company. He needs to pick up animals from various zoos and deliver them to other zoos.

   The company ships four types of animals:

   E: Elephant

   D: Dog

   C: Cat

   M: Mouse

   There are z zoos, numbered from 1 to z. There are a animals to transport. For each animal i, Capeta is given:

   - The type of the animal (t[i])
   - The source zoo (s[i]) where he has to pick it up
   - The destination zoo (d[i]) where he has to drop it off

   Capeta's truck has unlimited capacity, so it can carry any number of animals. However, there are certain rules he must follow:

   1. Capeta must visit zoos in strictly increasing order. He cannot skip any zoos.
   2. Once an animal is picked up, it must stay in the truck until it is dropped at its destination zoo.
   3. Due to compatibility constraints between animal types, Capeta cannot carry certain animals together:
      - Dogs are scared of Elephants → can't carry Dogs and Elephants at the same time
      - Cats are scared of Dogs → can't carry Cats and Dogs at the same time
      - Mice are scared of Cats → can't carry Mice and Cats at the same time

- Elephants are scared of Mice → can't carry Elephants and Mice at the same time

Due to these constraints, it may not be possible to carry all the animals. Capeta is allowed to ignore some animals. For the company's report, he needs to answer the following:
For each value of k from 1 to a, what is the minimum number of zoos that Capeta must reach in order to successfully transport k animals?
If it is impossible to transport k animals under the given constraints, return -1.
Sample Input:

```
2
10 3
E D C
4 1 4
7 5 8
10 6
E D C M E D
1 1 1 2 9 7
2 2 2 4 10 10
```

Sample Output:

```
5 8 -1
2 2 4 10 -1 -1
```

8. Tapan and Divya have a rectangular chocolate bar, consisting of M rows and N columns. Each cell of the bar contains a piece of chocolate labeled either:
   - 'T' (for Tapan),
   - 'D' (for Divya), or
   - 'U' (unlabeled and can go to either).

   They want to split the bar into exactly two pieces following these rules:
   Rules for Splitting:
   1. No Conflict Labels:
      - Tapan’s piece must not contain any 'D' chocolates.
      - Divya’s piece must not contain any 'T' chocolates.
   2. Connected Components:
      - Each piece must form one connected component — i.e., all chocolates in a piece must be reachable from one another using 4-directional connectivity (up, down, left, right).

3. Balanced Pieces:
    - The absolute difference in the number of chocolates between the two pieces should be at most K.
4. No 2×2 Squares:
    - Neither piece can contain 4 adjacent chocolates that form a 2×2 block, like this:

        XX
        XX

Sample Input:

2 2 4
UU
UU

Sample Output:

12