

Adv C Module

1. Subjective

2.1 Basic Refreshers

1. What are data types in C, and what are the different types of data types available?
2. Write the equation to set a bit in C.
3. Explain the ternary operator in C.
4. What is the difference between little-endian and big-endian memory representation?
5. Find the size of the following union:

```
union u {  
    int x;  
    struct s {  
        int y;  
        char c;  
    };  
};
```

6.

```
main()  
{  
    int i;  
    for (i=0; i<5; i++)  
    {  
        int i = 10;  
        printf ("%d", i);  
        i++;  
    }  
    return 0;  
}
```

 Find the output.

7. Which data type in embedded C is used to store an 8-bit integer?
 - a) char
 - b) int
 - c) short
 - d) long
8. What does the "const" keyword indicate in embedded C?
 - a) The variable is a constant pointer.

- b) The variable cannot be modified.
 - c) The variable is allocated in flash memory.
 - d) The variable is reserved for interrupt handling.
9. Which statement is used to exit a loop prematurely in embedded C?
- a) break
 - b) exit
 - c) return
 - d) continue
10. Which operator is used to access the value at the address stored in a pointer variable in embedded C?
- a) *
 - b) &
 - c) ->
 - d) #
11. What is the purpose of the "volatile" keyword in embedded C?
- a) It indicates that the variable cannot be modified.
 - b) It indicates that the variable is constant.
 - c) It indicates that the variable may change unexpectedly.
 - d) It indicates that the variable is stored in flash memory.
12. Give the output for the following Snippet of code
- ```
#include <stdio.h>
int main()
{
 int a=10,b=20,*p,s=0;
 p = &a;
 a++; (*p)++;
 s = a+b+ *p;
 print("%d\n",s);
 return 0;
}
```
13. #include <stdio.h>
- ```
int main()
{
    int a=-1,b= -a;
```

- ```

 int x,y;
 x = (a > 0) && (b < 0) || (a < 0) && (b > 0);
 y = (a <= 0) || (b >= 0) && (a >= 0) || (b <= 0);
 printf("%d\n", x == y);
 return 0;
 }
14. #include <stdio.h>
 main()
 {
 int i;
 for (i=0; i<5; i++)
 {
 int i= 10;
 printf("%d", i);
 i++;
 }
 return 0;
 }
15. #include<stdio.h>
 int main()
 {
 int arr[]={12, 13, 14, 15, 16};
 printf("%d,%d, %d\n", sizeof(arr), sizeof(*arr),
 sizeof(arr[0])); return 0;
 }
16. How is the 3rd element in an array accessed based on
 pointer notation?
 a. *a+3
 b. ("a+3)
 c. &(a+3)
 d. *(a+3)
17. main()
 {
 char x [10], "ptr = x;
 scanf("%s", x);
 change(&x[4])
 }

```

```

change(char a[])
{
 puts(a);
}

```

If abcdefg is the input, the output will be

18. What will be the size of the following structure?

```

#include <stdio.h>
struct temp {
 int a[10];
 char p;
};

```

19. What will be the output of the following piece of code?

```

for(i=0; i<10; i++);
printf("%d", i);

```

20. Union differs from structure in the following way

- A. All members are used at a time.
- B. Only one member can be used at a time.
- C. Union cannot have more members.
- D. Union initialized all members as structure.

21. The size of the following union, where an int occupies 4 bytes of memory is

```

union demo
{
 float x;
 int y;
 char z[10];
};

```

22. #include<stdio.h>

```

int main(void)
{
 int a = 1;
 int b = 0;
 b = ++a + ++a;
 printf("%d %d",a,b);
 getchar();
 return 0;
}

```

23. `#include<stdio.h>`  
`int main()`  
`{`  
`int a[] = {1, 2, 3, 4, 5, 6};`  
`int *ptr= (int)(&a+1);`  
`printf("%d", "(ptr-1));`  
`getchar();`  
`return 0;`  
`}`
24. Predict the output  
`#include<stdio.h>`  
`int main()`  
`{`  
`int n;`  
`for(n = 7; n!=0; n--)`  
`printf("n = %d", n--);`  
`getchar();`  
`return 0;`  
`}`
25. `#include <stdio.h>`  
`int mul(int a, int b, int c)`  
`{`  
`return a*b*c;`  
`}`  
`void main()`  
`{`  
`int (*function_pointer)(int, int, int);`  
`function_pointer = mul;`  
`printf("The product of three numbers is: %d",`  
`function_pointer(2, 3, 4));`  
`}`
26. Find the value of x that satisfies the condition:
- a.  $(x \& 0xF5) == 0x33$
  - b.  $(x \& 0xF3) == 0x33$
  - c.  $(x \& 0xF7) == 0x33$

## **2.2 1D Pointers and Functions**

1. What are pointers in C, and how are they used in memory access and manipulation?
2. How are arrays and loops used together in C programming? Provide an example.
3. What is the difference between pointers to int and pointers to char?
4. Explain about const pointer and pointer const (const int \*ptr & int \* const ptr).

## **2.3 Storage classes and memory segments**

1. What are storage classes in C? Explain each type with examples.
2. Explain the const keyword.
3. Explain memory allocation for initialized and uninitialized variables.
4. What is a static variable, and what is its importance?

## **2.4 2D Pointers and DMA**

1. What is the prototype of a function pointer in C?
2. What is Dynamic Memory Allocation (DMA) in C? Explain functions like malloc, calloc, realloc, and free.
3. What are memory segments? Explain with examples.
4. Explain the NULL, void, wild pointer, and dangling pointer.

## **2.5 Preprocessing**

1. What are the different stages of the compilation process in C?
2. During which compilation stage is memory for local variables allocated?
3. During which compilation stage is memory for global variables allocated?
4. How do you debug a segmentation fault in C? What steps or tools do you use?

## **2.6 UDT**

1. What is a macro, and how is it used?
2. What is the purpose of typedef, and how does it work?
3. What is the difference between macros and typedef?
4. What is a structure, and how is it different from a union?

5. What is structure padding, and how does it affect memory alignment?
6. What is an enum, and how is it typically used?
7. What are the key differences between structures, unions, enums, and macros?
8. Explain the bitfield.

## **2.8 Miscellaneous**

1. Explain the volatile keyword with an example.
2. What is the difference between volatile and const in C?

## **2.9 Project**

1. Explain the steganography project.
2. Explain the address book project.

## **2.10 FILE I/O**

1. What are file operations in C? Explain functions like fopen, fread, fwrite, and fclose.

## **2. Programming**

1. Write a program in C to check whether a number is prime.
2. Write a C program to check whether a number is even or odd.
3. Write a program to print the characters of a string if the 5th bit and 1st bit are set.
4. Write a program to sort a character array.
5. Write a program to find duplicates in an array with a time complexity of  $O(n)$ .
6. Write a program in C to find the largest of three numbers.
7. Write a program to swap two numbers using pointers in C.
8. Write logic in C to read one byte (character) from a file and write it to another file.
9. Write a C program to remove duplicates from an array.
10. In an array containing elements from 1 to n, how do you find the missing number?
11. Write a C program to print a simple pattern.
12. WAP to print unique elements in an array.
13. WAP to convert 0x1234 to 0x3421. Explain the steps involved.

14. Write a function to toggle the bits of a value at a given position (for values in the range 0–255).
15. WAP to find the second largest in an array.
16. Write a program to print the characters of a string if the 5th bit and 1st bit of their ASCII value are set.
17. Write a program to toggle the 5th bit in an 8-bit register.
18. Write a program to count the number of set bits in a given number.
19. Write a function to count the number of set bits in a number.
20. Write a C program to create a list using an integer array (strictly without using linked lists). Which data structure would you use?
21. Write code for strlen() and draw a perfect flowchart explaining its working. Iterate through the flowchart step-by-step.

## ***MC Module***

### **1. Basic electronics**

1. What is feedback in electronics?
2. Explain the working principle of a transformer.
3. What are the different types of power supplies?

### **2. Basics**

1. What is a seven-segment display? Explain its working and draw its circuit connections.
2. What is the difference between a microcontroller (MC) and a microprocessor (MP)?
3. What are the common features between a microcontroller and a microprocessor?
4. What is the difference between I/O and GPIO?



### **3. Interrupts**

1. What is a timer, and what are its types?
2. If an interrupt occurs while the main function is executing, and before its execution completes, another interrupt occurs, what will happen, and how will it be executed?

### **4. Projects**

1. Explain the car black box project.
2. What is a CAN-based project? Explain its working and real-time use cases.

### **5. ADC**

1. What is the difference between ADC and DAC? Why are they used, and where are they used?

### **6. Embedded Systems**

1. What is meant by embedded system?
2. How do you design an embedded system to monitor room temperature when it reaches threshold temperature?
3. How do you optimize code in embedded systems and power in firmware?
4. Why do you want to work in embedded systems?

### **7. Protocols**

1. What is the difference between the I2C and UART protocols and the frame format?
2. What is the CAN Protocol? Explain its features and applications. Draw the frame format.
3. Explain the SPI protocol in microcontrollers.

### **8. Programming**

1. Write the logic to transfer temperature data from one microcontroller board to another.
2. Write pseudocode to debounce a push button using software logic.
3. Write a program to create a delay using a for loop.
4. Write a function to make a CLCD glow ON and OFF for 50 ms.

## **CPP Module**

1. How can you implement polymorphism, encapsulation, and abstraction using C data structures?
2. Write a program to create a class using structures in C.

## ***Ds Module***

### **1. Basics**

1. What is a data structure?
2. What are the different types of data structures? Give one example for each type.
3. What is time complexity?

### **2. Linked lists**

1. What is meant by linked list?
2. What is the difference between a single linked list and a double linked list?
3. WAP to Implement a singly linked list and perform insertion, deletion, and traversal.
4. WAP to reverse a linked list (iterative & recursive).
5. WAP to detect a loop in a linked list.
6. WAP to find the middle node of a linked list.

### **3. Stack**

1. What is meant by stack?
2. WAP to implement a stack using an array and linked list.
3. WAP to implement 2 stacks in a single array.
4. Evaluate a postfix expression using a stack.

### **4. Searching and Sorting Techniques**

1. WAP to implement linear search and binary search.
2. WAP to find the first and last occurrence of an element in a sorted array.
3. WAP to implement bubble, selection, insertion, merge, quick, and heap sort.

## **5. Queue**

1. WAP to implement a queue using an array and linked list.
2. WAP to implement a circular queue.
3. WAP to implement deque (double-ended queue).
4. WAP to implement a priority queue.

## **6. Hashing**

1. WAP to implement a hash table with linear and quadratic probing.
2. WAP to check for duplicates in an array using hashing.
3. WAP to find the first non-repeating character in a string.
4. WAP to count the frequency of elements in an array.
5. WAP to implement the two-sum problem using hashing.

## **7. Trees**

1. WAP to implement a binary tree and perform inorder, preorder, and postorder traversals.
2. WAP to find the height of a binary tree.
3. WAP to check if a binary tree is balanced.

## **8. Projects**

1. What is the Inverted Search project? Explain its working and purpose.
2. Explain the APC project.

# ***LI Module***

## **1. Basics**

1. What is the purpose of the time command in Linux, and how is it used?
2. What does the date command do in Linux?
3. How does the top command help in monitoring system processes?
4. What is the function of the ifconfig command, and how is it used?
5. What is the use of the ifconfig and ping commands?
6. What are routers and IP addresses?

## **2. System call**

1. What is a system call? Explain its role in OS.
2. What is the difference between a system call and a function call?
3. What are pthread system calls?
4. Write a program to demonstrate the fork() system call and explain its output.
5. What is the difference between the exec() family and fork()?

## **3. Networking**

1. What is an IP address, and why is it important in networking?
2. Explain the OSI model.
3. What is network byte order?

## **4. Process**

1. Define process and its various states (with diagram).
2. What is the difference between a process and a thread?
3. Explain the process control block (PCB) and its contents.
4. What is context switching? Explain with steps.

## **5. IPC**

1. What are pipes in Linux? Can you explain with some commonly used pipe-related commands?

## **6. Signal**

1. What is a signal in OS?
2. Explain the signal handling mechanism in UNIX/Linux.
3. What is the difference between synchronous and asynchronous signals?
4. Explain the working of the kill(), raise(), and alarm() functions.

## **7. Socket**

1. What is the difference between TCP and UDP protocols?
2. What is the difference between FTP and TFTP?
3. What is the difference between a process and a thread?

## **8. Threads**

1. What is a semaphore?
2. Explain mutexes and semaphores.

## **9. Process and Memory Management**

1. Explain memory hierarchy in computer systems.
2. What is logical vs. physical address space?
3. Explain paging and segmentation with diagrams.
4. What are internal and external fragmentation? Give examples.

## ***General Questions Aptitude***

1. What type of display is used in trains for displaying information?
2. Explain the working of an operational amplifier (Op-Amp) with a circuit diagram.
3. What is a voltage controller? Explain with a suitable circuit.
4. What is a chopper in power electronics? Draw its circuit and explain its working.
5. If an LED is burnt out, what could be the possible reasons behind it? How would you identify and troubleshoot the issue?
6. How do you approach debugging an error in an electronic circuit? Describe your step-by-step method.
7. What tools or instruments would you use to find faults in a circuit (e.g., short circuits, open connections, incorrect voltages)?
8. If a microcontroller-based circuit is not responding, what checks will you perform to isolate the problem?
9. How would you determine whether a component (like a transistor or resistor) is faulty or not?
10. What is a seating arrangement puzzle, and how do you approach solving it?
11. Can you explain the 'crossing the bridge' puzzle and the logic used to solve it?
12. What are matching-type puzzles based on clues, and how should they be approached in logical reasoning tests?
13. Two people bet \$1 on each badminton match they played. The girl won 3 bets, and the boy ended up winning \$5. How many matches did they play in total?

14. A number is increased by 20% and then decreased by 10%. What is the overall percentage change?
15. If 40% of a number is 120, what is the number?
16. A student scores 450 marks out of 600. What is the percentage of marks obtained?
17. I am a 3-digit number. My tens digit is five more than my ones digit, and my hundreds digit is eight less than my tens digit. What number am I?
18. If yesterday's tomorrow is Thursday, what is today?
19. Five friends, A, B, C, D, and E, are sitting in a row. A is to the left of B but to the right of C. D is at the right end, and E is to the left of D. Who is in the middle?
20. Four people need to cross a bridge at night. They have one torch, and the bridge can hold only two at a time. Their crossing times are 1, 2, 7, and 10 minutes. How can they cross in the shortest time?
21. What is Android?
22. In which products is the Android OS used, and why?
23. What is the latest version of Android? Explain its latest features.
24. In computer architecture, what is the role of the CPU?
25. Background: In the vehicle ecosystem, indicators play a very important role by nudging the other riders/drivers. Left and Right indicators are represented concerning the User.
  - The left indicator represents the User shall take the left turn/line.
  - The right indicator represents the User shall take the right turn/line.
  - In hazard light, both left and right indicators toggle simultaneously. The Hazard Light represents that the user is facing some problems in the commute.

Task:

1. Develop the software for the indicator module according to the requirements below and test it on the evaluation board.
2. Base Software development shall be developed over the C programming language
3. Application software shall be developed over MATLAB Simulink and then use the code generator to generate automatic C code and merge with Base Software to generate binary files.

#### Required Materials :

- MicroController Kit to execute the code
- UART TTL which is used to log the UART data
- Two Push Buttons which are used as left & right indicator buttons
- Two LEDs which are used as left and right indicators
- Flasher tools for the controller
- Jumper wires may be required to connect the Buttons / LEDs

#### Technical Requirements:

- If the left indicator LED is OFF, then pressing the left indicator push button for 1 sec will turn ON the Left indicator LED. The same is the requirement for the right indicator.
- If the left indicator LED is ON, then pressing the left indicator push button for 1 sec will turn OFF the Left indicator LED. The same is the requirement for the right indicator.
- If the right indicator LED is already ON, then pressing the left indicator push button for 1 sec turn OFF the right indicator LED, and turn ON the left indicator LED.
- If the left indicator LED is already ON, then pressing the right indicator push button for 1 sec turn OFF the left indicator LED, and turn ON the right indicator LED.
- Switch ON any of the indicators will toggle the LED (s) ON-OFF-ON for every 300 msec
- Button press events shall be captured over GPIO and LED shall be driven over PWM
- Button press events, LED power status, Left / Right Indicator engagement status shall be published over UART
- 100 msec Scheduler shall be used to call the tasks in Base and Application software/
- If both left and right indicator buttons are pressed simultaneously for 1 sec then hazard light activation shall take place.
- Deactivation of the hazard light shall be performed by pressing the left/right indicator button press for 1sec.
- UART, GPIO, PWM, and timer driver development shall be implemented under the base software category. It shall be

modular, scalable, and developed in C Programming Language.

- Indicator engagement, disengagement, and toggle functionality shall be implemented in the Application category and it shall be modular, scalable, and Developed in MATLAB Simulink. MATLAB-generated code shall be integrated with Base software.

- Button press signal/pulse should be generated only once for each button press event. Multiple signals shall not be triggered if the button is pressed and held for a longer duration.

Submission Requirement: Create the software document for “indicators control over mobile phones(using BLE Protocol)” covering below

1. High-level Architecture and Software Requirement Specification
2. Your Github link of the code that you generated and checked-in.
3. Google link of the Video Recording of the working module in the zipped format to be added in the document
4. Link of UART Log