

# Adv C Module

## 1. Subjective

### 2.1 Basic Refreshers

1. Explain the break and continue in detail.
2. Why is int the return type for main?
3. Difference between C and Embedded C

### 2.2 1D Pointers and Functions

1. Explain the recursion function
2. What is a wild pointer?
3. What is a NULL pointer?
4. Explain the const pointer and its syntax.
5. What is call by reference and call by value?
6. When the function prototype is needed?
7. Explain about the segmentation fault in a pointer.

### 2.3 String

1. Explain the difference between a string and a char array.
2. What is the last character of the string?
3. Where will the memory be allocated for this "hello" string?

### 2.5 Storage classes and memory segments

1. Explain the static variable in detail. With an example.
2. Explain the extern keyword in detail.
3. Explain the storage class in detail.
4. Explain the segments in c.
5. where the memory will be allocated, for const static.
6. Explain the register keyword.
7. What is the difference between const int ptr and int const ptr.
8. Explain the memory segment.
9. How to assemble a file from the C file.
10. How to create an executable file with our own name in GCC.

11. Do you know memory leakage tools?
12. Explain the data segments in detail.
13. Explain the stage of compilation.
14. Explain about command line arguments.
15. How can you judge the segmentation fault and solve it.

## **2.6 2D Pointers and DMA**

1. Explain the difference between calloc and malloc.
2. Explain the inline function in detail.
3. What is a command-line argument?
4. What is a NULL pointer?
5. Explain the smart pointer in detail.
6. Explain the static keyword. static function.
7. Explain the function pointer.
8. How realloc knows how many bytes to allocate, and how free knows how many bytes to delete.

## **2.7 Preprocessing**

1. Explain the use of macro.

## **2.8 UDT**

1. Explain the difference between a struct and a union.
2. How to reduce the memory size of the structure.
3. Explain about structure padding and its alignment.
4. What is typedef in detail?
5. Declare the structure and the size of this structure.
6. What is the difference between a structure and union?  
Explain with a padding
- 7.

## **2.9 Miscellaneous**

1. Explain the volatile keyword.

## **2.10 FILE I/O**

1. Explain about file operation functions like fopen, fread, and fwrite.
2. What is the use of ftell?

## **2. Programming**

1. WAP to find the 2nd largest number in an array.
2. WAP to find the 2nd smallest number in an array.

3. WAP to clear the nth bit of a number.
4. WAP to print the n prime numbers.
5. WAP to print the fibonacci series.
6. WAP to reverse the array.
7. WAP to find the target element in a sorted array.
8. WAP to implement the strcpy and strncpy.
9. WAP to create a file and put some content into the file using the command line.
10. WAP to check whether the given number is even or not.
11. WAP to check if the given number is prime or not.
12. WAP to set, clear and toggle the nth bit in the number.
13. WAP to remove the duplicate element from the string.
14. WAP to swap two numbers using bitwise operator.
15. WAP to find the little endian or big endian using a normal variable and using a pointer.
16. WAP to transpose the matrix.
17. WAP to reverse the string.
18. WAP to find the unique element in an array using the bitwise operator.
19. WAP to transpose the 2d array.

## ***MC Module***

### **1. Basic electronics**

1. Explain the rectifier and inductor.
2. Explain the relay.

### **2. Basics**

1. Difference between a microcontroller and a microprocessor
2. Why did you choose the PIC microcontroller specifically?
3. How to dump code in a PIC controller.
4. Which compiler are you using for compilation?

### **3. Interrupts**

1. Explain the interrupt latency.
2. What is EEPROM, and its application?
3. Explain the watchdog timer.

## **4. Projects**

1. Explain about car black box project.
2. Explain about pick-to-light project.
3. Explain digital timer project.

## **5. ADC**

1. Explain about ADC in detail.
2. Explain the acquisitions and conversation time.

## **6. Protocols**

1. Explain the I2C protocol in detail.
2. What is the function of the stop bit in UART communication, and what happens if it's missing or corrupted?
3. Which parameter ensures both the transmitter and receiver are synchronized in UART communication?
4. What is the role of parity in UART, and how does it help in error detection?
5. What is the difference between full-duplex and half-duplex UART communication?
6. How does UART detect the start of a data frame during transmission?
7. Why is UART considered asynchronous, and how does it differ from synchronous serial communication?
8. Explain the CAN protocol.
9. List out the hardware which are used in the CAN automotive project
10. How the communication happen in I2C.

## **7. PWM**

1. How will you use PWM?
2. Give a real-time example for PWM

## **8. Programming**

1. Write a program to blink the LED alternately.

## **CPP Module**

1. Explain the OOPS concepts in detail

2. Explain function overriding and overloading.
3. Function overriding should perform initialize an array size 100, then enter 1,2,3 upto 100 then print in reverse order but after printing 100 we need to skip 1 element then print 98 then we need to skip 2 elements then 3 elements after come back to again 1,2,3 skip but the overriding function must be in derived class.
4. If a class has having static variable, will it be inherited in derived classes?
5. Structure and class difference?

## ***DS Module***

### **1. Basics**

1. Write the program for the makefile.
2. What is the use of a data structure?

### **2. Linked lists**

1. WAP to link to a sorted linked list.
2. Tell the types of linked lists and explain about those
3. Explain the difference between an array and a linked list.
4. Explain the circular linked list.

### **3. Stack**

1. Explain the stack application.
2. Explain the difference between prefix and postfix.
3. Explain the stack functions like push, pop, peek, and peep.

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

1. Explain the sorting technique you know.
2. Explain bubble sorting.
3. Explain insertion sorting.
4. Explain the selection sorting.
5. Explain quick sort in detail.
6. Explain binary search and its time complexity.

### **5. Queue**

1. Explain enqueue and dequeue in detail.
2. Explain the use of a queue and give real real-time example.

## **6.Hashing**

1. Explain about hash tables and their need.
2. Explain why you used a hash table in the inverted search project.

## **7.Trees**

1. Explain the binary search tree.
2. What do you know about BST?

# ***LI Module***

## **1.Basics**

1. Explain the types of RTOS.
2. How to change the directory, how to print a string in the terminal using Linux functions
3. How do you compile a C program on a Linux system?
4. Explain the booting sequence.
5. How do you know the kernel version using a command?
6. Explain systemctl.

## **2.System call**

1. What is a system call? Explain with an example.
2. Tell the C library function and a system call to open a file
3. Which is having the advantage, open or fopen?
4. Is an open system call OS independent?
5. Explain which one is faster library call or system call.
6. Tell me what system calls you know.
7. What is lseek, library function or system call?

## **3.Networking**

1. Explain the TCP/IP protocol in detail.
2. Explain the UDP protocol in detail.

## **4.Process**

1. What is the use of the fork system call?

2. Tell the prototype of the open, fork system call.
3. If in the program we are writing three times fork(), how many processes will be created?

## **5. IPC**

1. Explain the SHM in detail.
2. Explain the IPC concept in detail.
3. Is pipe is bidirectional.
4. If we want to read and write at any time, then which IPC will you use?
5. Apart from Pipe, FIFO, and shm any other ways to communicate with another process?

## **6. Signal**

1. What is the difference between signal and sigaction?
2. How many signals are we having?
3. How will you ignore the signal?
4. How to perform a new action in signals.
5. Explain why the SIGKILL signal is handled by the signal and sigaction system calls.

## **7. Socket**

1. What is a socket? If we don't close it, what will happen?
2. Explain TCP socket programming.
3. Explain the use of the bind system call.
4. Explain the use of accept and connect, and tell where we need to use them.
5. Explain the difference between the recv, recvfrom, sendto, and send.

## **8. Threads**

1. What are threads?
2. What library will you use for multithread?
3. What is the difference between a mutex and a semaphore?
4. What is a semaphore?
5. Explain the multi-thread implementation.

## ***Linux Systems***

1. Linux commands - ls, mv, cp, ifconfig, rm, and sed.
2. Difference between touch and vi command.
3. What does the ls command do?
4. What echo commands will do?
5. In a shell script, which & when you will use for and while loops.
6. Explain the grep command in detail.
7. Explain the chmod command.

## ***General Question***

1. If you have an issue with the firmware, how will you find or solve it?
2. How to debug the code without an IDE.
3. Explain the academic project in detail.
4. Explain why you choose the Emertxe for embedded course.