Adv C Module

1. MCQ

1) Which of the following is the correct syntax to declare a variable in C?
    a) int a;
    b) variable a;
    c) integer a;
    d) declare int a;
2) What is the size of an int data type in C (on a 32-bit system)?
    a) 2 bytes
    b) 4 bytes
    c) 8 bytes
    d) 1 byte
3) Which of the following is NOT true about #define macros?
    a) They are processed by the compiler
    b) They can be used to define constants
    c) They don’t consume memory
    d) They are replaced before compilation
4) What will be the output of this code, if the a = 5 and b = 9 based on user input?

```
#include <stdio.h>
int main()
{
        int a, b;
        scanf("%d%d", &a, &b);
        printf("%d%d%d", a, b, a + b);
}
```

    a) 5 9 14
    b) 9 5 14
    c) 5914
    d) 9514

5) If a function does not have a return statement, but its return type is int, what happens?
    a) The compiler fills 0 as the return value
    b) It returns garbage value
    c) Compilation error
    d) Segmentation fault at runtime
6) What is the result of freeing the same memory block twice using free() in C?
    a) Program terminates safely
    b) Nothing happens
    c) Memory leak
    d) Undefined behavior
7) How to declare a 2D array?
    a) int arr[3,3];
    b) int arr[3][3];
    c) int arr[][];
    d) array<int,3,3> arr;
8) What will be the output of the following code?

```
#include <stdio.h>
int main() {
        int a = 10, *p;
        p = &a;
        printf("%d", *p);
        return 0;
}
```

    a) 10
    b) Address of a
    c) Garbage value
    d) Compilation error
9) What is the output of the following code?

```
#include <stdio.h>
int main() {
        int arr[] = {10, 20, 30, 40};
        int *p = arr;
        printf("%d", *(p + 2));
```

```
        return 0;
}
```

a) 10
b) 20
c) 30
d) 40

10) What will break inside a loop?
a) Skip the next iteration
b) Exit the loop
c) Exit the program
d) Cause an error

11) What is the output of the following code?

```
#include <stdio.h>
int main() {
        if (0)
                printf("Hello");
        else
                printf("World");
        return 0;
}
```

a) Hello
b) World
c) No output
d) Compilation error

12) What is the output?

```
#include <stdio.h>
int main() {
        int *p = NULL;
        printf("%d", *p);
        return 0;
}
```

a) 0
b) Garbage value
c) Segmentation fault

d) Compilation error

13) What is the output?

```
#include <stdio.h>
int main() {
        int arr[] = {1, 2, 3, 4, 5};
        printf("%d", *(arr + 2));
        return 0;
}
```

a) 1
b) 2
c) 3
d) 5

14) What will happen when a function is called before its declaration?
a) Compilation error
b) Undefined behavior
c) Runtime error
d) Works fine if implicitly declared

15) Which of these statements is true about const variables in C?
a) You can’t modify them
b) They must be initialized
c) They are stored in read-only memory
d) They can be modified via pointers

16) Which of the following is a correct way to declare a pointer?
a) int *ptr;
b) int ptr*;
c) ptr int;
d) int ptr;

17) What will be the output of the following code?

```
#include <stdio.h>
int main() {
        char str[] = "Hello";
        printf("%c", *(str + 1));
```

```
    return 0;
}
```

a) H
b) e
c) l
d) o

18) How many times will the following loop execute?

```
int i = 0;
while (i < 5) {
 printf("%d ", i);
}
```

a) 5 times
b) 4 times
c) Infinite times
d) Compilation error

19) What will happen if you declare two global variables with the same name in two different C files and link them together?

a) Compiler error
b) Linker error
c) No error — they'll refer to the same variable
d) They will be treated as separate variables

20) Which of the following C statements about static is FALSE?

a) A static variable retains its value between function calls
b) A static global variable is visible only to that file
c) Static functions cannot be recursive
d) Static variables are initialized only once

2. Subjective

2.1 Basic Refreshers

1. What is the difference between a++ and a=a+1? Which one will execute fast?
2. What is the purpose of using void main() in a program?
3. Explain data types in detail
4. What is C?

5. What is the difference between int and float
6. Explain what is meant by array and how to define an array.

2.2 1D Pointers and Functions

1. What is a Segmentation fault? Explain with an example.
2. What is the difference between pass by value and pass by reference?
3. Explain pointers with an example
4. Explain the NULL pointer
5. Explain what is meant by function, with advantages and an example program.
6. What is a function?

2.3 String

1. Explain the strcat in detail.

2.5 Storage classes and memory segments

1. Explain Storage Classes in detail.
2. Explain the Static and Volatile keywords with examples.
3. Explain the Compilation stages in detail.
4. Explain the Static function with examples.

2.6 2D Pointers and DMA

1. What is a Memory leak? Explain with an example.
2. How will you free the memory without using free()?
3. Explain the function pointers with examples.
4. Explain the dynamic memory allocation.
5. Explain DMA with examples.
6. Explain the Dangling pointer with an example.
7. Explain the difference between malloc and calloc
8. Write the syntax for the realloc function.
9. Write the syntax for the malloc function. What is the return type?

2.7 Preprocessing

1. Explain the compilation stages with relevant commands.

2.8 UDT

1. Explain the differences between Structure and Union with examples.
2. Explain the bitfields.
3. What is structure? Explain the examples

2.9 Miscellaneous

1. Explain the volatile keyword
2. Explain the difference between const and volatile.

3. Programming

1. WAP to set, clear, and toggle the nth bit.
2. WAP to find the endianness of the system using both pointer and union methods.
3. WAP to store the student details and display the result.
4. WAP to reverse the array elements using a pointer.
5. WAP to find the maximum and minimum element in an array using a pointer.
6. Write a program in C for a 4-player chess game.
7. Write a program to swap two numbers using pointers (with and without using a temporary variable).
8. Write a program to determine if a system is little-endian.
9. Given a nibble, write a C program to print each character (or digit) separately.
10. WAP to find the second largest element in an array using pointers.
11. WAF to print the series of numbers from 1 up to a given number N.

12. WAP that accepts input from the user and prints the entered data directly.
13. WAP word reverse of a String
14. WAP to find HCF and LCM using recursion.

MC Module

1. Basic electronics

1. What is an IGBT?
2. What are harmonics in a 3-phase power supply?
3. Draw the circuit diagram for a full-load rectifier.
4. What is a harmonic filter?

2. Basics

1. What is a voltage conversion device?
2. What is a MOSFET? How can it be switched, and what are its terminals? Is the gate terminal similar to a regular logic gate?

3. Interrupts

1. Draw the block diagram for a microcontroller.
2. How is data stored from a microcontroller to EEPROM?
3. Define a register and its usage in microcontrollers, mentioning specific registers you've used.
4. Explain timers.
5. Explain the counters.
6. What is the difference between volatile and non-volatile memory?
7. What is the difference between a microcontroller and a microprocessor?
8. What is a watchdog timer?
9. What are the components used in a microcontroller?
10. What are the peripherals used in a microcontroller?
11. Explain the EEPROM.

4. Projects

1. Explain the car black box project and draw the block diagram.
2. Which protocol did you use in the car block project, and how does it work?
3. About the implementation of timers in the project.
4. Microwave oven project

5. ADC

1. Explain the ADC in detail.

6. Embedded Systems

1. Explain about embedded system.

7. Protocols

1. Explain the UART protocol with a frame format.
2. Explain the SPI protocol with a frame format.
3. Explain the I2C protocol with a frame format.
4. Explain the CAN protocol with a frame format.
5. Which protocol is faster, I2C or SPI?
6. Which communication protocol offers higher speed: SPI or I2C?
7. Which communication protocols are commonly used for controlling light blinking and video display in embedded systems?

8. PWM

1. Explain how the PWM is working.

CPP Module

1. How will you create a user-defined template in C++? Give an example.
2. Explain the Pillars of OOPS.
3. What is STL in C++? Why do we use it?
4. What are the main components of STL?

5. Name a few commonly used STL containers.
6. What is the difference between a vector, a list, and a deque?
7. How does a map differ from an unordered_map?
8. What is the difference between a set and a multiset?
9. What is the difference between struct and class in C++?
10. What is the difference between new and malloc?
11. What are access specifiers in C++? Name them.
12. What is function overloading? Give an example

Ds Module

1. Basics

1. Explain linear and non-linear data structures.

2. Linked lists

1. Explain the types of linked lists
2. WAP to reverse an SLL.
3. WAP to insert a node at the end in SLL
4. WAP to remove duplicates in SLL.
5. Explain the usage of Makefile.
6. What is a Self-referential structure? Explain with an example.
7. WAP to check whether the linked list contains a loop or not.
8. WAP to delete the nth last element in an SLL.
9. What is the difference between algorithms and data structures?

3. Stack

1. What is a Stack? Explain with an example
2. Explain the complete code on the stack implementation and applications
3. WAP to demonstrate push and pop operations in the stack.

4. Searching and Sorting Techniques

1. WAP to sort the array by using the bubble sort method
2. WAP to sort the given array by using merge sort
3. Write a binary search function using iterative/recursive method.
4. Explain inverted search in detail.

5. WAP to perform Insertion sort.
6. WAP to sort the given array by using quick sort and its time complexity
7. Explain quick sort algorithms.
8. Explain searching algorithms.

5. Queue

1. What is a Queue? Explain with an Example.
2. WAP to Implement a Queue using an array
3. WAP to Implement a Queue using a Linked List
4. Explain the applications of a Queue.
5. What is a circular queue? WAP to implement a Circular queue.

6. Hashing

1. What is a hash table, and what is the use of the hash table? Give an example of a hash table briefly about with an example. What is the time complexity of the hash program?

7. Trees

1. WAP to print the pre-order output of a given BST
2. Explain the real-time Implementation of the tree.
3. WAP to find the height of a tree.
4. WAP to print in-order output of a given BST.

8. Project

1. Explain the arbitrary precision calculator project.
2. Explain the inverted search project.
3. Write a C program to create a hotel billing system by which data structure of this problem can be implemented. The hotel has breakfast, lunch, and dinner

   A-> the maximum billed meal in a month and mention the day.

   For example, this month, breakfast is sold at the maximum on Monday.

   B-> The minimum billed meal in a month, and mention the day.

For example, this month, breakfast is sold at a minimum on Wednesday.

LI Module

1. Basics

1. Explain about kernel.
2. Explain the use of OS in detail.

2. System call

1. Explain the difference between a Library call and a system call.

3. Networking

1. Explain about TCP and UDP.
2. Which is reliable between TCP and UDP?
3. What are the TCP and UDP protocols? What is the difference between the TCP and UDP protocols?

4. Process

1. Explain how to create the process.
2. Explain the use of exec system and its variation in detail with examples.

5. IPC

1. What is a pipe?
2. What is shared memory?
3. Why? (with real-time Example) IPC mechanism brief Explanation (PIPE, FIFO, SHM).
4. Explain the IPC mechanism.

6. Signal

1. Explain how to handle the SIGKILL signal.
2. Explain when the SIGCHLD signal will occur.

7. Socket

1. Explain socket programming.

8. Threads

1. What is a semaphore?
2. Race condition? Critical section Implementation with mutex method and Semaphore (Using real-time example)
3. When we are using the mutex method and when we are using the semaphore. (with real-time example)
4. Instead of a mutex, if you use a semaphore? Instead of a semaphore, if you use a mutex, what happens? Explain in detail.
5. What is the difference between a binary mutex and a mutex with an example?
6. What is the difference between thread and pthread?
7. Explain about process and thread. Can we create threads or processes in Kernel Space?
8. What is the difference between a mutex and a semaphore?
9. What is the difference between a process and a thread?

9. Process and Memory Management

1. Explain the Process and memory management

General Questions & Aptitude

1. Two trains, one from Howrah to Patna and the other from Patna to Howrah, start simultaneously. After they meet, the trains reach their destinations after 9 hours and 16 hours respectively. The ratio of their speeds is:
    1.1. 2:3
    1.2. 4:3
    1.3. 6:7
    1.4. 9:16

2. A, B, and C can do a piece of work in 20, 30, and 60 day,s respectively. In how many days can A do the work if he is assisted by B and C on every third day?
    2.1. 12 days
    2.2. 15days
    2.3. 16days
    2.4. 18days
3. A vendor bought toffees at 6 for a rupee. How many for a rupee must he sell to gain 20%?
    3.1. 3
    3.2. 4
    3.3. 5
    3.4. 6
4. A is two years older than B who is twice as old as C. If the total of the ages of A, B and C be 27, then how old is B?
    4.1. 7
    4.2. 8
    4.3. 9
    4.4. 10
5. The captain of a cricket team of 11 members is 26 years old and the wicket keeper is 3 years older. If the ages of these two are excluded, the average age of the remaining players is one year less than the average age of the whole team. What is the average age of the team?
    5.1. 23 years
    5.2. 24 years
    5.3. 25 years
    5.4. None of these
6. A bag contains 4 white, 5 red and 6 blue balls. Three balls are drawn at random from the bag. The probability that all of them are red, is:
    6.1. 1/22
    6.2. 22
    6.3. 2/91
    6.4. 2/77

7. Which of the following is a prime number?
    7.1. 33
    7.2. 81
    7.3. 93
    7.4. 97
8. 10, 25, 45, 54, 60, 75, 80
    8.1. 10
    8.2. 45
    8.3. 54
    8.4. 75
9. If a person walks at 14 km/hr instead of 10 km/hr, he would have walked 20 km more. The actual distance travelled by him is:
    9.1. 50km
    9.2. 56km
    9.3. 70km
    9.4. 80km
10. A person's present age is two-fifths of the age of his mother. After 8 years, he will be one-half of the age of his mother. How old is the mother at present?
    10.1. 32 years
    10.2. 36 years
    10.3. 40 years
    10.4. 48 years
11. What does the term RTOS stand for?
    11.1. Real-Time Operating Scheme
    11.2. Real-Time Operating System
    11.3. Reliable Time Operating System
    11.4. Rapid Task Optimization Software
12. What is the function of a pull-up resistor in a circuit?
    12.1. Reduce power
    12.2. Ensure a default high state
    12.3. Ensure a default low state
    12.4. Amplify signals

13. When using an analog sensor with a microcontroller, which pin is typically used?
    13.1. PWM
    13.2. RX
    13.3. TX
    13.4. ADC
14. In PCB layout, what is a ‘via’?
    14.1. A wire
    14.2. A resistor
    14.3. A hole connecting different layers
    14.4. A power source
15. What happens when an interrupt is triggered?
    15.1. The CPU ignores it
    15.2. The CPU completes the current instruction and jumps to the ISR
    15.3. System resets
    15.4. Power is cut off
16. Which of the following is a commonly used communication protocol in IoT?
    16.1. HTTP
    16.2. MQTT
    16.3. FTP
    16.4. SMTP
17. In an IoT system, which layer is responsible for sensing the environment?
    17.1. Network layer
    17.2. Application layer
    17.3. Perception layer
    17.4. Transport layer
18. Which software is most commonly used for professional PCB design?
    18.1. Notepad++
    18.2. Adobe Photoshop
    18.3. Kicad
    18.4. Microsoft word

19. What is the main purpose of using ground planes in PCB design?
    19.1. Control the PWM signal
    19.2. Increase clock speed
    19.3. Reset the system if the software hangs
    19.4. Power off the system