1. Subjective

2.1 Basic Refreshers

1. Explain "if," "else if," and "switch" with real-time examples.
2. Explain qualifiers and modifiers with an example.
3. What is overflow, and what happens if a value exceeds the storage limit of a data type (e.g., storing beyond the range of char)?
4. Explain the difference between LIFO (Last In First Out) and FIFO (First In First Out) with real-time examples for each. Considering the scenario where water is poured into a jug, identify whether it represents LIFO or FIFO behavior, and explain your reasoning. If it reflects a LIFO scenario, how can you modify or interpret it to represent a FIFO scenario instead?

2.2 1D Pointers

2.3 Stdio

2.4 Storage classes

1. Can we define a variable with const volatile? If yes, explain how.
2. Can we use the const and volatile keywords at the same time?
3. Explain static variables and their linkage with an example.

2.5 2D pointers and DMA

1. Explain about double-pointer with an example
2. What is the use of function pointers?
3. Explain dangling pointers with an example program. How can we avoid them?

2.6 UDT

1. Explain Structure padding with an example
2. Explain the difference between structure and union with examples.
3. What is the difference between call by value and call by reference, and which one is preferred when passing a structure in C, and why?

2.7 Files

2.8 Projects

1. Explain the MP3 tag reader project.
2. Explain the steganography project.
3. Where is steganography used, and where is cryptography used?

2.9 Miscellaneous

1. Explain the volatile keyword with an example program.

2. Programming

1. WAP to get 'n' bits from a given position of a number (bitwise)
2. WAP to check whether the string is palindrome or not, if yes, return 1; otherwise, return 0 (strings)
3. WAP to count the number of set bits in a given integer (bitwise)
4. WAP to count the number of zeros in a given integer (bitwise or general)
5. Write a function to check if a number is a power of 2 or not (bitwise)
6. WAP to sort the array elements based on the number of set bits in each element, with elements having more set bits appearing later in the array. For example, given the array {10, 20, 30, 40, 50}, the sorted array should be {10, 20, 40, 50, 30}. (Arrays & Bitwise operators)
7. WAP to remove the substring and replace it with a newly entered string. (Strings)

8. WAP to move all zeroes to the end of the given array. (Array)
9. WAP to rotate the array by N positions. (Array)
10. WAP to create a structure named Book to store book details like title, author, and price. Write a C program to input details for three books. Find the most expensive and the lowest-priced books and display their information. (Struct)
11. Take a 2D array int a[m][n] and write a function to replace each array element with the 5th prime number starting from the corresponding array element.
12. Write a function in C to reverse each word in a given string without changing the word order. The function should work within constrained memory limits, avoiding dynamic memory allocation (e.g., malloc) and using only a limited number of additional variables. Assume that the string can fit within a standard fixed-length buffer, as is typical in embedded systems.
13. Write a program to print the following pattern based on the input N, where N is a positive integer: example

```
        A
      A B A
    A B C B A
  A B C D C B A
A B C D E D C B A
```

14. Write an embedded C program to implement a circular buffer (ring buffer) to store bytes of data. The buffer should operate in a First-In-First-Out (FIFO) manner and be efficient in terms of memory and processing. This type of buffer is commonly used in embedded systems for serial communication.
15. Write a program to generate n random numbers and calculate the 5-point weighted moving average. In a weighted moving average, each point in the window has a different weight. Use the weights {1, 2, 3, 2, 1} for the

5-point window, where the center point has the highest weight.

16. Write a program to count the number of set bits in an integer, and if the count is even, swap the nibbles of each byte.

17. Write a Function in C to print the following pattern
(For n=4)
1
5 2
8 6 3
10 9 7 4

18. Write a C program to dynamically allocate memory for N strings, take input, sort them, and print both the original and sorted lists. Ensure proper memory allocation and deallocation
Example
Enter number of strings: 3
Enter 3 strings:
Controller
Processor
Command
Given array is:
0: Controller
1: Processor
2: Command
Sorted array is:
0: Command
1: Controller
2: Processor

19. Implement a library management system in C using structures and dynamic memory allocation. Structure should contain book ID, title, and author. The system should allow:
Adding books dynamically.
Searching for a book by ID.

Displaying all books stored in the system.

20. Write a C program to find a pair of numbers in an array that sum up to a given target value.
21. Write a C program to determine if a given number can be expressed as the sum of two prime numbers. The program should take an integer input and check all possible prime pairs that sum to the given number.
22. WAP to find all the non-repeating characters in a given string and print them.
23. Write a program to print n prime numbers (it should work for any big number).

3. MCQ

1)

```
#include <stdio.h>

int main()
{
   int x[5] = { 10, 20, 30 };
   printf("%d", x[-1]);
   return 0;
}
```

0

10

Garbage value

Error

2)

The members of union can be accessed using ___.

Dot Operator (.)

And Operator (&)

Asterisk Operator (*)

Right Shift Operator (>)

3)

Output of the following code

```
#include <stdio.h>
 int  main()
   {
       if (sizeof(int) > -1)
           printf("True");
       else
           printf("False");
   }
```

True
False

4)

Will the following C code compile without any error?

```
 #include <stdio.h>
    int main()
    {
        for (int k = 0; k < 10; k++);
            return 0;
    }
```

Yes
No
Depends on the C standard implemented by compilers
Error

5)

What will be the output of the following C code?

```
#include <stdio.h>
int main(){
    int i, j;
    for (i = 2; i < 10; i++) {
        for (j = 2; j <= (i / j); j++)
            if (!(i % j))
                break;
        if (j > (i / j))
            printf("%d ", i);
    }

    return 0;
}
```

1 2 3 4

2 3 5 7

Compiler error

Nothing will be printed

6)

What will be the final value of x in the following C code?

```
#include <stdio.h>
    void main()
    {
        int x = 5 * 9 / 3 + 9;
    }
```

3.75

Depends on compiler

24

3

7)

What is the output of the following code snippet?

```
#include <stdio.h>
int main()
{
    int x = 5;
    int y = (x++) + (++x);
    printf(“%d”, y);
    return 0;
}
```

9

10

11

12

8)

```
int main()
{
    int _ = 10;
    int __ = 20;
    int ___ = _ + __;
    printf("__%d",___);
    return 0;
}
```

Compilation Error

Runtime Error

__0

__30

9)

What will be the output of the following code snippet?

```
#include <stdio.h>
void solve()
{
     int x = 2;
     printf("%d", (x << 1) + (x >> 1));
}
int main()
{
    solve();
    return 0;
}
```

4

5

2

1

10)

What will be the result of the following code snippet?

```
#include <stdio.h>
void solve() {
    char ch[10] = "abcdefghij";
    int ans = 0;
    for(int i = 0; i < 10; i++) {
        ans += (ch[i] - 'a');
    }
    printf("%d", ans);
}
int main() {
    solve();
    return 0;
}
```

45

36

20

100

11)

What will be the output

```
int main()
{
     int x = 10;
    {
        int x = 0;
        printf("%d",x);
    }
    return 0;
}
```

10

0

compilation error

Runtime error

12)

What will be the output of the following C code?

```
#include <stdio.h>

int main()
{
    printf((43 > 43) ? "value 1 is greater!" : "value 1 is not greater!");
    return 0;
}
```

value 1 is not greater

value 1 is greater

Error

None of the Above

13)

What will be the output of the following code snippet?

```
#include <stdio.h>
union School {
    int age, rollNo;
    double marks;
};
void solve() {
    union School sc;
    sc.age = 19;
    sc.rollNo = 82;
    sc.marks = 19.04;
    printf("%d", (int)sizeof(sc));
}
int main()
{
    solve();
    return 0;
}
```

4
8
16
12

14)

Will compiler produce any compilation error if same header file is included two times?

Yes
NO

15)

What will be the output of the following C code considering the size of a short int is 2, char is 1 and int is 4 bytes?

```
 #include <stdio.h>
    int main()
    {
        short int i = 20;
        char c = 97;
        printf("%d, %d, %d\n", sizeof(i), sizeof(c), sizeof(c + i));
        return 0;
    }
```

2, 1, 2
2, 1, 1
2, 1, 4
2, 2, 8

16)

What will be the output of the following C function?

```
  #include <stdio.h>
```

```
enum birds {SPARROW, PEACOCK, PARROT};
enum animals {TIGER = 8, LION, RABBIT, ZEBRA};
int main()
{
    enum birds m = TIGER;
    int k;
    k = m;
    printf("%d\n", k);
    return 0;
}
```

0
compile time error
1
8

17)

What is the output of the following code snippet?

```
#include <stdio.h>
#include<stdlib.h>
void set(int *to) {
    to = (int*)malloc(5 * sizeof(int));
}
void solve() {
    int *ptr;
    set(ptr);
    *ptr = 10;
    printf("%d", *ptr);
}
int main() {
    solve();
    return 0;
}
```

10
Garbage value
Cannot Say

Segmantation fault

18)

Below code output

```
int main()
{
   int a = 320;
   char *ptr;
   ptr =( char *)&a;
   printf("%d",*ptr);
   return 0;
}
```

320
64
60
160

19)

What will be the value of the following assignment expression?
(x = foo())!= 1 considering foo() returns 5

2
true
1
0

20)

What is the output of the following code snippet?

```
#include <stdio.h>
int main()
 {
   int x = 5;
   int y = x << 2;
   printf(“%d”, y);
   return 0;
}
```

5

10

15

20

21)

What is the difference between the following 2 C codes?

```
#include <stdio.h> //Program 1
 int main()
 {
    int d, a = 1, b = 2;
    d =  a++ + ++b;
    printf("%d %d %d", d, a, b);
 }

 #include <stdio.h> //Program 2
 int main()
 {
    int d, a = 1, b = 2;
    d =  a++ +++b;
    printf("%d %d %d", d, a, b);
 }
```

No difference as space doesn’t make any difference, values of a, b, d are same in both the case

Space does make a difference, values of a, b, d are different

Program 1 has syntax error, program 2 is not

Program 2 has syntax error, program 1 is not

22)

What will be the output of the following code snippet?

```
#include <stdio.h>
int get(int n) {
   if(n <= 1) {
      return n;
   }
   return get(n - 1) + get(n - 2);
}
```

```
void solve() {
    int ans = get(6);
    printf("%d", ans);
}
int main() {
    solve();
    return 0;
}
```

5
1
0
8

23)

What will be the output of the following code snippet?

```
#include <stdio.h>
void solve() {
    int a = 3;
    int res = a++ + ++a + a++ + ++a;
    printf("%d", res);
}
int main() {
    solve();
    return 0;
}
```

12
24
20
18

24)

What will be the output of the following C code on a 64 bit machine?

```
  #include <stdio.h>
    union Sti
    {
        int nu;
        char m;
    };
    int main()
    {
        union Sti s;
        printf("%d", sizeof(s));
        return 0;
    }
```

8
3
5
4

25)

What is the result after execution of the following code if a is 10, b is 5, and c is 10?

a = 10, c = 10
a = 11, c = 10
a = 10, c = 11
a = 11, c = 11

26)

What will be the final values of i and j in the following C code?

```
#include <stdio.h>
int x = 0;
int f()
{
    if (x == 0)
        return x + 1;
    else
        return x - 1;
}
int g()
{
    return x++;
}
```

```
int main()
{
   int i = (f() + g()) | g(); //bitwise or
   int j = g() | (f() + g()); //bitwise or
   printf(“%”)
}
```

i value is 1 and j value is 1
i value is 0 and j value is 0
i value is 1 and j value is undefined
i and j value are undefined

27)

What will happen if the following C code is executed?

```
#include <stdio.h>
int main()
{
   int main = 3;
   printf("%d", main);
   return 0;
}
```

It will cause a compile-time error
It will cause a run-time error
It will run without any error and prints 3
It will experience infinite looping

28)

What does the following declaration indicate?

```
int x: 8;
```

x stores value of 8
x is an 8 bit integer
Both A and B
None of the Above

29)

Which one of the following is invalid macro in C programming?

#pragma

#error
#ifndef
#elseif

30)

Output of the following code

```
#include <stdio.h>
int  main()
  {
     char *p = 0;
     *p = 'a';
     printf("value in pointer p is %c\n", *p);
  }
```

It will print a
It will print 0
Compile time error
Run time error

31) What is the output of the following code?

```
#include <stdio.h>
int main() {
   int arr[] = {1,2,3,4,5,6,7,8,9};
   int *ptr1 = arr;
   int *ptr2 = arr+2;
   printf("%d ", ptr2 - ptr1);
   printf("%ld", (char*)ptr2 - (char*)ptr1);
}
```

Options:

1. 5
2. 20
3. 2
4. 20 20

32) Which of the following is true?

Options:

1. gets() doesn’t do any bound testing and should not be used.

2. fgets() should be used in place of gets() only for files, otherwise gets() is fine.
3. gets() cannot read strings with spaces.
4. None of the above.

33) What will be the output of the following C function?

```
#include <stdio.h>
enum birds {SPARROW, PEACOCK, PARROT};
enum animals {TIGER = 8, LION, RABBIT, ZEBRA};
int main() {
   enum birds m = TIGER;
   int k;
   k = m;
   printf("%d\n", k);
   return 0;
}
```

Options:

1. 0
2. Compile time error
3. 1
4. 8

34) Which function is used to set file position to the beginning in C?

Options:

1. rewind()
2. fseek(fp, 0, SEEK_SET)
3. Both a and b
4. Iclseek()

35) Given the following code, which option correctly prints the second element of the first array?

```
int arr[2][4] = {{1,2,3,4}, {5,6,7,8}};
int (*p)[4] = arr;
```

Options:

1. printf("%d", *(p[0] + 1));
2. printf("%d", *(*p + 1));
3. printf("%d", p[0][1]);
4. All of the above

36) What will be the output of the following C code snippet?

```
#include <stdio.h>
void solve() {
    bool ok = false;
    printf("%s", ok ? "YES" : "NO");
}
int main() {
    solve();
    return 0;
}
```

Options:

1. YES
2. NO
3. Compilation error
4. None of the above

37) What will be the output of the following C code snippet?

```
#include <stdio.h>
void main() {
    1 ? 2 ? return 1 : return 2;
}
```

Options:

1. returns 1
2. returns 2
3. Varies
4. Compile time error

38) What is the worst-case time complexity of searching for an element in an unordered array?

Options:

1. O(1)
2. O(log n)
3. O(n)
4. O(n log n)

39) What is the output of this C code?

```
#include <stdio.h>
int main() {
    int n = 0, m = 0;
    if (n = 0)
        printf("True");
    else
        printf("False");
}
```

Options:

1. True
2. False
3. No output will be printed
4. Run time error

40) What is the output of the below program?

```
#include <stdio.h>
int main() {
    union {
        char x;
        int z;
    };
    myvar.z = (1 << 10);
    printf("%d, %d", myvar.x, myvar.z);
}
```

Options:

1. 4, 100
2. 0, 100
3. garbage value, 100
4. garbage value, 100

41) What will be the output of the following C code?

```
#include <stdio.h>
int main() {
    int y = 10000;
    int y = 34;
    printf("Hello World %d\n", y);
    return 0;
}
```

Options:

1. Hello World 34
2. Compile time error
3. Hello World 10000
4. Hello World followed by a junk value

42) If the price of a commodity is increased by 50%, by what fraction must its consumption be reduced so as to keep the same expenditure on its consumption?

Options:

1. 1/4
2. 1/3
3. 1/2
4. 1/2

43) What will be the output of the following code snippet?

```
#include <stdio.h>
void solve() {
   printf("%d %d", 023, (23));
}
int main() {
   solve();
   return 0;
}
```

Options:

1. 023 23
2. 23 23
3. 19 23
4. 23 19

44) What is the output of the following C code?

```
#include <stdio.h>
void main() {
   printf("%d", sizeof('7'));
}
```

Options:

1. 1

2. 2
3. 4
4. 8

45) What is the output of the following C code?

```
#include <stdio.h>
int main() {
    char *s = "Hello";
    s[0] = 'h';
    printf("%s", s);
    return 0;
}
```

Options:

1. hello
2. Hello
3. Undefined behavior
4. Segmentation fault

46) What happens if the following program is executed in C and C++?

```
#include <stdio.h>
int main() {
    int new = 5;
    printf("%d", new);
}
```

Options:

1. Error in C and successful execution in C++
2. Error in both C and C++
3. Error in C++ and successful execution in C
4. A successful run in both C and C++

47) What will be the output of the following C code?

```
#include <stdio.h>
int main() {
    for (int g = 2; g < 10; g++) {
        if (g % 3 == 0)
            break;
        printf("%d ", g);
    }
    return 0;
}
```

Options:

1. 2 3 5 7
2. 2 3 4 5 6 7 8 9
3. 2 3 5
4. 2

48) What is the purpose of __attribute__((packed)) in C?

Options:

1. To align structures to word boundaries
2. To pack structure members with minimum padding
3. To enable structure padding
4. To declare a function inline

49) Find out the output of the below code snippet?

```
#include <stdio.h>
int fun(void)
{
        return 50;
}

int main()
{
        static int var fun();
        printf("%d\n", var);
        return 0;
)
```

options:

1. Compiler erro
2. 50
3. undefined behaviour
4. No error

50) 4. A sum of Rs. 1360 has divided among A,B, and C such that A gets 2/3 of what B gets and B gets 1/4 of what C gets. B's share is

Options:

1. Rs 120

2. Ra 160
3. Rs 240
4. Rs 300

51) What is the output of the fallowing code snippet?

```
#include <stdio.h>
int main()
{
        Int x = 5;
        Int y = ++x + x++;
        printf("%d", y);
        return 0;
}
```

Options;

1. 10
2. 11
3. 12
4. 13

52) What will be the output of the following code snippet?

```
#include <stdio.h>
void solve()
{
        bool ok = false;
        printf(ok ? “YES” : “NO”);
}
int main()
{
        solve();
        return 0;
}
```

Options:

1. YES
2. NO
3. Compilation error
4. None of the above

53) In a class, there are 15 boys and 10 girls. Three students are selected at random. The probability that 1 girl and 2 boys are selected is:

1. 21/46
2. 25/117
3. 1/50
4. 3/25

54) What is the output of the below program?

```
#include <stdio.h>
int main()
{
        Int local = 5;
        if(local == 3, 4)
        {
                printf(“hello”\n”);
        }
        else
        {
                printf(“world\n”);
        }
        return 0;
}
```

1. Hello
2. World
3. Compilation error
4. Runtime error

55) Present ages of Sameer and Anand are in the ratio of 5 : 4 respectively. Three years hence, the ratio of their ages will become 11 : 9 respectively. What is Anand’s present age in years?

1. 24
2. 27
3. 40
4. 22

56) What will be the result of the following code snippet?

```
#include <stdio.h>
void solve()
{
        char ch[10] = “abcdefghij”;
        int ans = 0;
        for(int i = 0; i<10;i++)
```

```
    {
        ans +=(ch[i]-’a’);
    }
    printf(“%d”, ans);
}
int main()
{
    solve();
    Return 0;
}
```

1. 36
2. 100
3. 45
4. 20

57) What is the output of the following code?

```
#include <stdio.h>
#define SQUARE(x) x*x
int main()
{
    int a = 3;
    printf(“%d\n”, SQUARE(a+1));
    return 0;
}
```

1. 16
2. 12
3. 10
4. Complier error

58) What is the difference between memcpy and memmove?

1. memmove is faster than memcpy
2. memcpy handles overlapping memory regions safely
3. memmove handles overlapping memory regions safely
4. memcpy is thread-safe, while memmove is not