⇒ Long Answer Questions (LAQ's):-

1) Demonstrate program on Bubble Sort and factorial of a number using Recursion?

Ans:- * Bubble Sort:-

```
#include <stdio.h>
int main() {
int a[100];
int trip, N, i, temp;
printf(" Enter the size of array :\n");
scanf("%d", &N);
printf(" Enter %d elements:\n", N);
for(i=0 ; i<N; i++) {
scanf("%d", &a[i]);
}
for (trip=1 ; trip<N; trip++) {
for (i=0; i<N; i++) {
if (a[i] > a[i+1]) {
temp = a[i]; a[i] = a[i+1]; a[i+1] = temp;
} } }
printf(" \n the sorted array list is :\n");
for(i=0; i<N ; i++)
printf("%d\t", a[i]);
}
```

Factorial of a Number Using Recursion:

```
#include <stdio.h>
long int multiplyNumber(int n);
int main() {
  int n;
  printf("Enter a positive integer: ");
  scanf("%d", &n);
  printf("Factorial of %d = %ld", n, multiplyNumber(n));
  return 0;
}
long int multiplyNumber(int n) {
  if (n >= 1)
    return n * multiplyNumber(n-1);
  else
    return 1;
}
```

2) Classify and explain any six file handli functions with examples?

Ans:- **'C' - File Operations:-** C - File Operations refer to the different possible Operation that we can perform on a file in C, such as:

* **Create a File in C:-** The fopen() function can not only open a file but also can create a file if it does not exist alread For that, we have to use the modes that allow the creation of a file if not found such as 'w', w+, wb, wb+, a, a+, ab & ab+.

* **Syntax:-**

```
FILE *fptr;
fptr = fopen("filename.txt", "w");
```

* **Reading from a file:-** The file read operati in C can be performed using functions fscanf or fgets. Both the functions performed the same operations as that of scanf & gets but with an additional parameter, the file pointer

Function	Description
fscanf()	Use formed string and variable arguments list to take input from a file
fgets()	Input the whole line from the file
fgetc()	Reads a single character from the file
fgetw()	Reads a number from a file
fread()	Reads the specified bytes of data from a binary file.

★ Write to a file :- The file write operations can be performed by the functions fprintf() and fputs() with similarities to read operations. C-Programming also provides some other functions that can be used to write data to a file, such as:

Function	Description
fprintf()	Similar to printf(), this function use formatted string & varible arguments list to print output to the file.
fputs()	Prints the whole line in the file and a new line at the end.
fputc()	Prints a single character into the file.
fputw()	Prints a number to the file.
fwrite()	This functions write the specified amount of bytes to the binary file.

* Closing a File:- The fclose() functions is used to close the file. After successful file operations, you must always close a file to remove it from the Memory.

⇒ Syntax:- fclose (file_pointer);

where, the file_pointer is the pointer to the opened file.

5) Discover what is a pointer and how the assay of pointers is used in C.

Ans:- **Array of Pointers in C** :- In C, a pointer array is a homogeneous collection of indexed pointer variables that are references to a memory location. It is generally used in C Programming when we want to point at multiple memory locations of a similar data type in our C - Program. We can access the data by dereferencing the pointer pointing to it.

* Syntax :- pointer_type *array_name [array_size];

Here,

=> pointer_type : Type of data the pointer is pointing to.

=> array_name : Name of the array of pointers.

=> array_size : Size of the array of Pointers.

Note :- It is important to keep in mind the Operator precedence and associativity in the array of pointers declarations of diffuent

type as a single change will mean the whole different thing. For example enclosing

* array-name in the parenthesis will mean that array-name is a pointer to an array.

Pointers:- A pointer is defined as a derived datatype that can store the address of other 'C' variables or a memory location. We can access and manipulate the data stored in that memory location using pointers.

=> Syntax of C Pointers:- The syntax of pointers is similar to the variable declaration in C. but we use the (*) dereferencing operator in the pointer declaration.

data-type *ptr;

where, ptr is the name of the pointer datatype is the type of data it is pointing to.

4) Write a program for selection sort & Explain?

Ans:-

* Selection Sort :-

```
#include <stdio.h>
int main() {
int A[100];
int small, N, i, j, temp;
printf("Enter the size of array : \n");
scanf("%d", &N);
printf("\n Enter array elements : \n", N);
for (i=0; i<N; i++)
scanf("%d", &A[i]);
for (i=0; i<N; i++) {
small = i;
for (j=i+1; j<N; j++) {
if (A[j] < A[small]) {
small = j;
}}
temp = A[i];
A[i] = A[small];
A[small] = temp;
}
printf("\n The sorted array list is : \n");
for(i=0; i<N; i++)
printf("%d\t", A[i]);
}
```

* Output:-

```
Enter the size of array : 5
Enter array elements :
2
4
9
1
7
The sorted array list is :
1 2 4 7 9
```

* Selection Sort:- Selection Sort in C is a sorting algorithm that is used to sort a list of elements in either an ascending order or a descending order.

=> Algorithm:-

1. Start
2. Declare the variables & array.
3. Initialize the size of array
4. Provide the array elements
5. Compare the two values, store the swap result in temporary variable if still more swapping needed.
6. If compare & swap result in proper position then store the result.
7. Repeat the iteration upto given size.
8. Stop.

5) What is a pointer ? How it is used in self-referential structure.

Ans:- **Pointers:-** A Pointer is defined as a derived data type that can store the address of other 'C' variables or a memory location. We can access and manipulate the data stored in that memory location using pointers.

★ **Syntax of C Pointers:-** The syntax of pointers is similar to the variable declaration in C, but we use the (*) dereferencing operator in the pointer declaration.

```
datatype *ptr;
```

where,

⇒ ptr is the name of the pointer.

⇒ datatype is the type of data it is pointing to.

- The above syntax is used to define a pointer to a variable. We can also define pointers to functions, structures, etc.

* **Self - Referential Structures :-** Self - Referential Structures are those structures that have one or more pointers which point to the same type of structure as their member.

Self Referential Structures

```
struct node {
    int data1;
    char data2;
    struct node *link;
};
```

⇒ In other words, structures pointing to the same type of structures are self-referential in nature.

20/12/24