1. What is pointer and explain its applications.

A pointer is a variable that contains a memory address of variable. Since these memory addresses are the locations in the computer memory where program instructions and data are stored, pointers can be used to access and manipulate data stored in the memory. Pointers are undoubtedly one of the most distinct features of C languages. It has added power and flexibility to the language tool and handy to use once they are mastered.

The applications of pointer are as follows:

1. Pointer is widely used in Dynamic Memory Allocation.
2. Pointer can be used to pass information back and forth between a function and its reference point.
3. Pointers provide an alternative way to access individual array elements.
4. Pointers increase the execution speed as they refer address.
5. Pointers are used to create complex data structures such as linked list, trees, graphs and so on.
6. Pointer reduces length and complexity of program.
7. The use of pointer arrays to character strings results in saving of data storage space in memory.
8. Pointers are more efficient in handling arrays and data tables.

3. **Justify that pointer is jewel in C language. Write a function that is passed an array of n pointers to floats and returns a newly created array that contains those n float values in reverse order. Assume any necessary data.**

->

Pointer is a variable whose value is the address of another variable i.e. it directs the address of the memory location. Pointer can also be known as jewel in C language. Some of its reasons are as follows: -

a. Pointer allows passing array and strings easily from one function to another function.
b. Pointer makes convenient to manipulate array.
c. Pointer allows creating complex data structure.
d. Pointer allows returning more than one value a called function to calling function by reference.
e. Pointer allows creating data structure such as linked list.
f. Pointer allows conveniently obtaining memory from the system.

6. Explain the pointer to structure with example.

Pointers can be accessed along with structures. A pointer variable of structure can be created as below:

```
struct name {
        member1;
        member2;
        .
        .
        };
        ……inside function……..
        struct name *ptr;
```

Here, the pointer variable of type struct name is created.

Structure's member through pointer can be used in two ways:

a) Referencing pointer to another address to access memory
b) Using dynamic memory allocation

7.Explain the pointer arithmetic with example.

Various arithmetic operators used in pointers can be said as pointer arithmetic. Some of them are as follow: -

a. Increment (++): Increase value of pointer by 1.
b. Decrement (--): Decrease value of pointer by 1.
c. Comparison (==, <,>): Compares pointer values.

It can be further understood clearly by the help of example in next text file: -

10. What is dynamic memory allocation (DMA)? How can you use it in C?

The process of allocating and freeing memory at run time is known as dynamic memory allocation. This reserves the memory required by the program and returns valuable resources to the system once the use of reserved space is utilized. There are four library routines known as “memory management function” that can be used for allocating and freeing memory during program execution. These functions help us build complex application programs that use the available memory intelligently. These four library functions comes under <stdlib.h> header file in C. They are as follows:

a) malloc( ): It helps to allocate a block of memory
b) calloc( ): It helps in allocating multiple of block of memory.
c) free( ): It is used in releasing the space used.
d) realloc( ): It helps in altering the size of a block.

Regarding the use of the DMA in C it makes the program more convenient. Malloc helps us to allocate the size of memory that is required for user so that is helps in minimization of wastage of unused memory and saves the memory of the computer i.e. only used memory can be taken by the computer.

Syntax:

```
ptr=(data_type*) malloc(size_of_block);
```

Calloc is another memory allocation function that is normally used for requesting memory space at run time for storing derived data types such as arrays and structures. While malloc allocates single block of storage, calloc allocates multiple blocks of storage, each of the same size, and then sets all bytes to zero.

Syntax:

```
ptr=(data_type*)calloc(no_of_blocks,size_of_each_block);
```

Free is another memory allocation function that helps to release the unused space of the program. When we no longer need the data we stored in a block of memory, and we do not intend to use that block for storing any other information, we may release the block of memory for future use, using the free function

Syntax:

```
free(ptr);
```

Realloc is also another memory allocation function which is used to modify the size of previously allocated space. Sometime while coding the problem may arise that the space we allocate at before may not be sufficient and we

may require more space in this case we use realloc for reallocating the size of a block for addition of more information.

Syntax:

ptr= realloc(ptr,newsize);

11. What are the advantages of dynamic memory allocation over static memory allocation?

When we have gone through the static memory allocation concept, it is clear that the static memory allocation major disadvantages. To overcome those disadvantages dynamic memory allocation is used. That is the reason why dynamic memory allocation has many advantages over static memory allocation.

When main memory is allocated statically it cannot be altered during the execution of program. When main memory is allocated dynamically it can be altered during the execution of program as per the user wish. The length of dynamically allocated memory either can be decreased or increased. This is the major advantage of dynamic memory allocation over static memory allocation.

Suppose user wishes to enter elements of an array or a list one by one just after allocating memory and he may stop at any point then definitely it is not possible by means of static memory allocation. So the dynamic memory allocation certainly has an advantage in this case over static memory allocation. Such dynamically created list is called as linked list.

In case of dynamically created lists insertions and deletions can be done very easily just by the manipulation of addresses whereas in case of statically allocated memory insertions and deletions lead to more number of movements and wastage of memory.

In case of statically allocated memory there is every chance of "overflow" during insertions in the lists, whereas in case of dynamically allocated memory it does not come into picture unless otherwise unavailability of main memory.

For some liner data structures like STACK and QUEUE, dynamic memory allocation proves very much appropriate because the length of such data structure is not fixed. They may increase or decrease dynamically that is during the execution of program. Similarly non-linear data structures need recursive definitions of "struct" data type, so the need of dynamic memory allocation.

12. How is malloc() function different from calloc() function?

Malloc and Calloc both are memory allocation function used in C for dynamic memory allocation. These two have different function and declaration type.

Firstly, malloc is memory allocation function used in allocating the block of memory. It reserves a block of memory of specified size and returns a pointer of type void. This means that we can assign it to any type of the pointer. Its declaration syntax is:

ptr=(data_type*) malloc(size_of_block);

But Calloc is another memory allocation function that is normally used for requesting memory space at run time for storing derived data types such as arrays and structures. While malloc allocates single block of storage, calloc allocates multiple blocks of storage, each of the same size, and then sets all bytes to zero.Its declaration syntax is:

ptr=(data_type*)calloc(no_of_blocks,size_of_each_block);