Here, name is a pointer which points to a structure type and is also an element of the same structure.

Example

```
struct element
{
    char name{20};
   int   num;
    struct   element * value;
}
```

Element is of structure type variable. This structure contains three members

- a 20 elements character array called **name**
- An integer element called **num**
- a pointer to another structure which is same type called **value.** Hence it is self referential structure.

These structure are mainly used in applications where there is need to arrange data in ordered manner.

5.5 Unions

Introduction

A Union is a collection of heterogeneous elements. That is, it is a group of elements; each element is of different type. They are similar to structures. However, there is a difference in the way the structures members and union members are stored. Each member within a structure is assigned its own memory location. But the union members all share the common memory location. Thus, unions are used to save memory. Unions are chosen for applications involving multiple members, where values need to be assigned to all of the members at any one time.

5.6 Definition of Union

Union is a data type through which objects of different types and sizes can be stored at different times.

The general form of union type variable declaration is

```
        Union          name
```

```
{
data type  member-1;
        data type member-2;
data type  member-3;
        ………………….
        …………………
data type member-n;
}
```

The declaration includes a key word Union to declare the union data type. It is followed by user defined name, followed by curly braces which includes the members of the union

Example

```
union       value
{
        int no;
        float sal;
        char sex;
};
```

Characteristics of Union

1. Union stores values of different types in a single location in memory
2. A union may contain one of many different types of values but only one is stored at a time.
3. The union only holds a value for one data type. If a new assignment is made the previous value has no validity.
4. Any number of union members can be present. But, union type variable takes the largest memory occupied by its members.

5.7 Differences between Structure and Unions

Structure	Union

1. Struct StructureName
{
datatype
member-1;
datatype
member-2;
….
….
datatype
member-n;
};
	1. Union name {
	datatype
	member-1;
	datatype
	member-2;
	….
	….
	datatype
	member-n;
	};

2. Every structure member is allocated memory when a structure variable is defined	2. The memory equivalent to the largest item is allocated commonly for all members
3. All the members can be assigned values at a time	3. Values assigned to one ember may cause the change in value of other members.
4. All members of a structure can be initialized at the same time	4. Only one union member can be initialized at a time
5. value assigned to one member will not cause the change in other members	5. Value assigned to one member may cause the change in value of other members.
6. The usage of structure is efficient when all members are actively used in the program	6. The usage o union is efficient when members of it are not required to be accessed at the same time.

Model Questions

Short Answer Type Questions - 2 Marks

1. What is Structure? Write the syntax of "structure" (struct).
2. What is Pointer? Which variables are used to represent it?
3. What are the advantages of pointers?
4. What are the operators used with pointers.