SNS COLLEGE OF TECHNOLOGY
(An Autonomous Institution)

COIMBATORE – 35
DEPARTMENT OF AIML

First Year, 2nd Semester

16 Marks Question and Answer

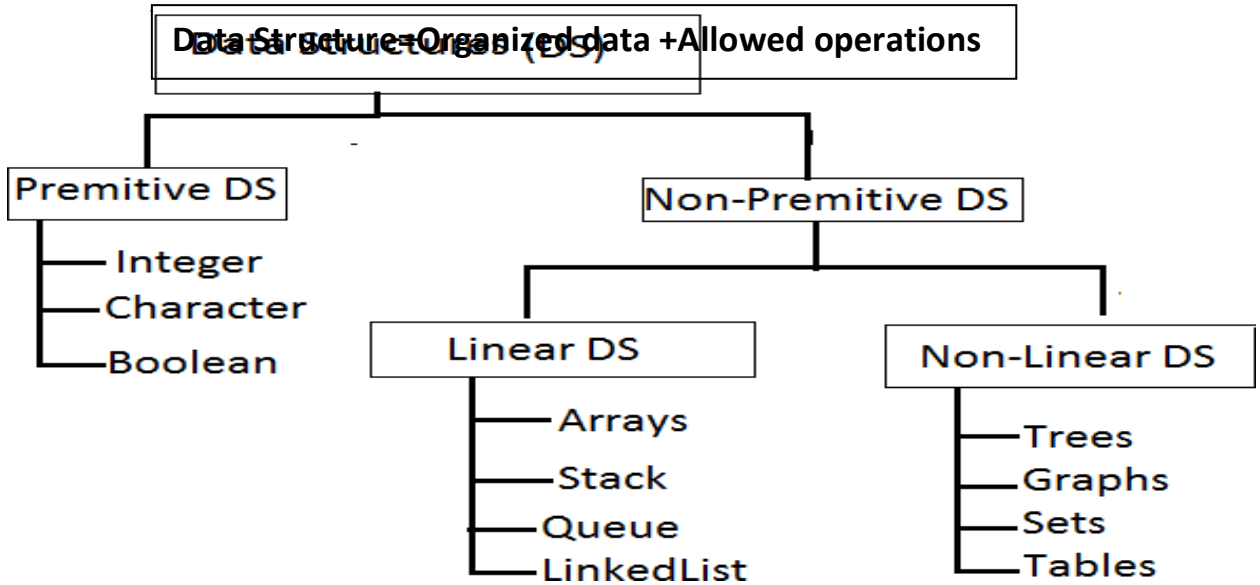
UNIT 3

1.What is data structure? Explain the classification of data structure?

A data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.

A data structure is not only used for organizing the data. It is also used for processing, retrieving, and storing data. Different basic and advanced types of data structures are used in almost every program or software system that has been developed. So we must have good knowledge of data structures.

The organized collection of data is called a 'Data Structure'.



Data Structure involves two complementary goals. The first goal is to identify and develop useful, mathematical entities and operations and to determine what class of problems can be solved by using these entities and operations. The second goal is to determine representation for those abstract entities to implement abstract operations on this concrete representation

Primitive Data structures are directly supported by the language ie; any operation is directlyperformed in these data items.
Ex: integer, Character, Real numbers etc.
Non-primitive data types are not defined by the programming language, but are insteadcreated by the programmer.

Linear data structures organize their data elements in a linear fashion, where data elements are attached one after the other. Linear data structures are very easy to implement, since the memory of the computer is also organized in a linear fashion. Some commonly used linear data structures are arrays, linked lists, stacks and queues.

In nonlinear data structures, data elements are not organized in a sequential fashion. Data structures like multidimensional arrays, trees, graphs, tables and sets are some examples of widely used nonlinear data structures.

2. What is an array? Explain it with an example?

Array is the collection of similar data types or collection of similar entity stored incontiguous memory location. Array of character is a string. Each data item of an array is called an element. And each element is unique and located in separated memory location. Each of elements of an array share a variable but each element having different index no. known as subscript.

An array can be a single dimensional or multi-dimensional and number of subscripts determines its dimension. And number of subscript is always starts withzero. One dimensional array is known as vector and two dimensional arrays are known as matrix.

ADVANTAGES: array variable can store more than one value at a time whereother variable can store one value at a time.

Example:

```
int arr[100];
int mark[100];
```

DECLARATION OF AN ARRAY **:**

Its syntax is :

Data type array name [size];

int arr[100];

int mark[100];

int a[5]={10,20,30,100,5}

The declaration of an array tells the compiler that, the data type, name of the array,size of the array and for each element it occupies memory space. Like for int data type, it occupies 2 bytes for each element and for float it occupies 4 byte for each element etc. The size of the array operates the number of elements that can be stored in an array and it may be a int constant or constant int expression.

We can represent individual array as : int

ar[5];

ar[0], ar[1], ar[2], ar[3], ar[4];

Symbolic constant can also be used to specify the size of the

array as:#define SIZE 10;

INITIALIZATION OF ANARRAY:

After declaration element of local array has garbage value. If it is global or static array then it will be automatically initialize with zero. An explicitly it can be initialize that

Data type array name [size] = {value1, value2, value3…}

Example:

in ar[5]={20,60,90, 100,120}

Array subscript always start from zero which is known as lower bound and upper value is known as upper bound and the last subscript value is one less than the sizeof array. Subscript can be an expression i.e. integer value. It can be any integer, integer constant, integer variable, integer expression or return value from functional call that yield integer value.

So if i & j are not variable then the valid subscript are ar [i*7],ar[i*i],ar[i++],ar[3];

The array elements are standing in continuous memory locations and the amount of storage required for hold the element depend in its size & type.

Total size in byte for 1D array is:

Total bytes=size of (data type) * size of array.

Example : if an array declared is: int [20];

Total byte= 2 * 20 =40 byte.

int mark[100];

DECLARATION OF AN ARRAY :

Its syntax is :

Data type array name [size];

int arr[100];

int mark[100];

int a[5]={10,20,30,100,5}

The declaration of an array tells the compiler that, the data type, name of the array, size of the array and for each element it occupies memory space. Like for int data type, it occupies 2 bytes for each element and for float it occupies 4 byte for each element etc. The size of the array operates the number of elements that can be stored in an array and it may be a int constant or constant int expression.

We can represent individual array as :int ar[5];

ar[0], ar[1], ar[2], ar[3], ar[4];

Symbolic constant can also be used to specify the size of the

array as:#define SIZE 10;

INITIALIZATION OF AN ARRAY:

After declaration element of local array has garbage value. If it is global or static array then it will be automatically initialize with zero. An explicitly it can be initialize that

Data type array name [size] = {value1, value2, value3…}

Example:

Int ar[5]={20,60,90, 100,120}

Array subscript always start from zero which is known as lower bound and upper value is known as upper bound and the last subscript value is one less than the sizeof array. Subscript can be an expression i.e. integer value. It can be any integer, integer constant, integer variable, integer expression or return value from functional call that yield integer value.

So if i & j are not variable then the valid subscript are ar [i*7],ar[i*i],ar[i++],ar[3];

The array elements are standing in continuous memory locations and the amount of storage required for hold the element depend in its size & type.

Total size in byte for 1D array is:

Total bytes=size of (data type) * size of array.

Example : if an array declared

is:int [20];

Total byte= 2 * 20 =40 byte.

3. What are the Operations on the Data Structure?

Following operations can be performed on the data structures:

1. Traversing
2. Searching
3. Inserting
4. Deleting
5. Sorting
6. Merging

1. *Traversing-* It is used to access each data item exactly once so that it can be processed.
2. *Searching-* It is used to find out the location of the data item if it exists in the given collection of data items.
3. *Inserting-* It is used to add a new data item in the given collection of data items.
4. *Deleting-* It is used to delete an existing data item from the given collection of data items. *5. Sorting-* It is used to arrange the data items in some order i.e. in ascending or descending order in case of numerical data and in dictionary order in case of alphanumeric data.
5. *Merging-* It is used to combine the data items of two sorted files into single file in the sorted form.

4. Write a program to input values into an array and display them?

```
#include<stdio.h>

int main()

{

int arr[5],i; for(i=0;i<5;i++)

{

printf("enter a value for

arr[%d]                \n",i);

scanf("%d",&arr[i]);


}

printf("the        array
```

```
elements  are:  \n");
for (i=0;i<5;i++)
{
printf("%d\t",arr[i]);
}
return 0;
}
```

Output

```
Enter a value for arr[0] = 12
Enter a value for arr[1] =45
 Enter a value for arr[2] =59
Enter a value for arr[3] =98
Enter a value for arr[4]  =21
The array elements are 12  45  59  98  21
```