



## QUESTION BANK

Name of the Department : Electronics and Communication Engineering

Subject Code & Name : EC8393 Fundamentals of Data Structures in C

Year & Semester : II & III

### UNIT I C PROGRAMMING BASICS

#### PART-A

#### 1. What is a program?

A program is a set of instructions written to carry out a particular task, so that computer can perform some specified task. Example program to find sum of 2 numbers

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

```
void main()
```

```
{ int a, b, sum;
```

```
printf("\nEnter two no: "); scanf("%d %d", &a, &b); sum = a + b;
```

```
printf("Sum : %d", sum); }
```

Output : Enter two no: 5 6

Sum : 11

#### 2. What are the main features and applications of C language? (JAN 2011)

- ✓ **Case sensitive:** C is case sensitive language.
- ✓ **Modularity:** we can split the program into number of modules. It allows reusability of modules.
- ✓ **Middle level language:** as a middle level language C combines both the advantages of low level and highlevel languages. (arrays, pointers etc). Efficient Use of Pointers
- ✓ **General purpose programming language:** C can be used to implement any kind of applications such as math's oriented, graphics, business- oriented applications.
- ✓ **Portability:** we can compile or execute C program in any operating system such as unix, dos, windows.
- ✓ **Powerful programming language:** C is very efficient and powerful programming language, it is best used for data structures and designing system software.

#### 3. List the primary data types in "C"? (NOV/DEC 2018)

There are four basic data types available in C.

- ✓ int  char
- ✓ float  double

#### 4. Define with example integer and floating type of data in C language. (JAN 2011)

- The **int** is used to define integer numbers. An integer occupies 2 bytes memory space and its value range limited to -32768 to +32767 (that is, -215 to +215-1) Example: int count=5
- The **float** is used to define floating point numbers. The float data type is used to store



fractional numbers (real numbers) with 6 digits of precision. Floating point numbers are denoted by the keyword float. Example: float miles = 4.5

## 5. Define Constants in C. Mention the types.

✓ The constants refer to fixed values that the program may not alter during its execution. These fixed values are also called literals. Constants can be of any of the basic data types like an integer constant, a floating constant, a character constant, or a string literal. There are also enumeration constants as well.

✓ The constants are treated just like regular variables except that their values cannot be modified after their definition.

**Syntax :** const data\_type constant\_name = value

## 6. What is a Variable? Illustrate it with an example. (NOV/DEC 2014)

✓ A variable is a data name used for storing a data value.

✓ Can be assigned different values at different times during program execution.

✓ Can be chosen by programmer in a meaningful way so as to reflect its function in the program. Example: int i, num; where i, num are variables that can store integer values

## 7. Difference between interpreter and compiler?

### Interpreter

### Compiler

Translates and executes the program line by line.

Scans the entire program before translating it into machine code.

The interpreter executes one line at a time, after checking and correcting its syntax errors and then converting it to machine code.

Converts the entire program to machine code and executes program only when all the syntax errors are removed.

Good for fast debugging.

Slow in debugging or removal of mistakes from a program.

Program execution time is more.

Program execution time is less.

Eg: Python, Ruby

Eg: C, C++

## 8. What is the difference between while loop and do...while loop? (or)

**Differentiate Entry and Exit controlled constructs.**

### While

### Do-while

It is an entry controlled loop

It is an exit controlled loop

In the while loop the condition is first executed. If the condition is true then it executes the body of the loop. When the condition is false it comes out of the loop

In the do...while loop first the statement is executed and then the condition is checked. The do...while loop will execute at least one time even though the condition is false at the very first



time

**Syntax:**

```
while(condition)
{ //body of the loop
}
```

**Syntax:**

```
do
{ //body of the loop
} while(condition);
```

**9. What are the I/O Functions in C? (MAY 2015, JAN 2016)**

- ✓ Formatted I/O Statements: scanf(),printf()
- ✓ Unformatted I/O Statements: getchar(), putchar(), gets(), puts()

**10. What are various types of C operators? (JAN 2014)**

- ✓ Arithmetic Operators
- ✓ Increment and Decrement Operators
- ✓ Assignment Operators
- ✓ Relational Operators
- ✓ Logical Operators
- ✓ Conditional Operators
- ✓ Bitwise Operators

**11. What is a Modulo Operator? What is the use of sizeof( ) operator? (NOV/DEC 2017)**

- ✓ Modulo Operator
- '%' is modulo operator. It gives the remainder of an integer division Example: a=17, b=6. Then c=%b gives 5.

- ✓ sizeof( ) operator

The sizeof ( ) operator gives the bytes occupied by a variable.

No of bytes occupied varies from variable to variable depending upon its data types.

Example: int x,y;

```
printf(“%d”,sizeof(x));
```

**Output: 2**

**12. What are the different versions of selection statements?**

- ✓ **One-way decision** statements do a particular thing or they do not.
- ✓ **Two-way decision** statements do one thing or do another.
- ✓ **Multi-way decision** statements can do one of many different things depending on the value of an expression.

**13. Write a for loop statement to print numbers from 10 to 1. (JAN 2014)**

```
for( i=10;i>=1;i--)
```

```
printf(“\n%d”,i);
```

**14. What is an array? Give an example.**

- ✓ Array is data structure
- ✓ It stores objects/variables of same data types in contiguous memory locations

Array elements can be accessed using index/subscript numbers . **Example:** int arr[10];



## 15. Define Compilation, linking process? (or) What is meant by Linking process?

✓ **Compilation** refers to the processing of source code files (.c, .cc, or .cpp) and the creation of an 'object' file. This step doesn't create anything the user can actually run. Instead, the compiler merely produces the machine language instructions that correspond to the source code file that was compiled.

✓ **Linking** refers to the creation of a single executable file from multiple object files. In this step, it is common that the linker will complain about undefined functions. During compilation, if the compiler could not find the definition for a particular function, it would just assume that the function was defined in another file. If this isn't the case, there's no way the compiler would know it doesn't look at the contents of more than one file at a time. The linker, on the other hand, may look at multiple files and try to find references for the functions that weren't mentioned.

## 16. What are the main elements of an array declaration? (or) Give the general form of array declaration. (NOV/DEC 2006)

1. Data type of array object
2. Name of array object
3. Size of the array

**Syntax:** datatype arrayName [ arraySize ]; **Example:** int arr[10];

**17. Why is it necessary to give the size of an array in an array declaration?** When an array is declared, the compiler allocates a base address and reserves enough space in the memory for all the elements of the array at compile time. Therefore size must be specified in array declaration in order to allocate the required space.

## 18. What are the classifications of array?

✓ **ONE-Dimensional array:** Single or One Dimensional array is used to represent and store data in a linear form.

✓ **Syntax:** <data-type> <array\_name> [size];

(Example: int arr[10])

✓ **TWO- Dimensional array:** The elements of a 2D array are arranged in rows and columns

**Syntax:** <data-type> <array\_name> [row\_subscript][column- subscript];

(Example: int arr[10][20])

✓ **Multi-Dimensional array:** Array having more than one subscript variable is called Multi-Dimensional array.

**Syntax:** <data-type> <array\_name> [row\_subscript][column-subscript]; (Example: int arr[10][20][15])

## 18. Write the features of array. (NOV/DEC 2017)

- ✓ Array contains data elements of same data type.
- ✓ Array stores fixed number of data elements.
- ✓ The elements in the array can be accessed by the index of the array.
- ✓ The elements are stored in contiguous location in an array.

## 19. Define String.

A string in C is merely an array of characters. The length of a string is determined by a terminating null character: '\0'. So, a string with the contents, say, "abc" has four characters: 'a', 'b', 'c', and the terminating null character.



Example: char name [] = {'a','b','c','\0'};

## 20. How is a Character String declared? (MAY 2015)

Syntax: char array\_name[size] = "String";

**Example:** char mes[] = {'w','e','l','c','o','m','e','\0'}; Or char mes[] = "Welcome"

## 21. Write example code to declare two dimensional array?

**Syntax:** <datatype> Arrayname [row size][column size];

**Example:** #include<stdio.h>

```
void main() {
```

```
✓ int a[3][3]; //2-d Array with 3 row and 3 column }
```

## 22. What is meant by Sorting? What are its types?

Sorting refers to ordering data in an increasing or decreasing fashion according to some linear relationship among the data items. Sorting can be done on names, numbers and records.

Types of sorting available in C:

- ✓ Insertion sort
- ✓ Merge Sort
- ✓ Quick Sort
- Radix Sort
- Bubble sort
- Heap Sort
- Selection s

## 23. Define Searching.

Searching for data is one of the fundamental fields of computing. Search is an operation in which a given list is searched for a particular value. The location of the searched element is informed.

Various types of searching techniques in C are

- ✓ Linear search
- Binary search

## 24. Write a C program to get a paragraph of text as input. (Nov/Dec 2019)

```
✓ #include<stdio.h>
```

```
int main()
```

```
{
```

```
char para[150];
```

```
printf("Enter Your Paragraph:"); scanf("%[^\t]s", para);
```

```
printf("Your Entered Paragraph is: %s", para); return 0;
```

```
}
```

### Output:

Enter Paragraph : Fundamental of Data structures in C is an easy paper

✓ Accepted Paragraph : Fundamental of Data structures in C is an easy paper

✓

## PART B

1. Explain the Structure of C Program in detail.
2. Explain the different types of operators available in C and the associativity with example(NOV/DEC 2017) (APRIL/MAY 2017) (NOV/DEC 2019)
3. Explain various input and output functions of C language in detail.
4. (i)With an example program explain the various decision making statements available in



C.

(ii) Explain switch case statement and 'for' loop statement with suitable example.

5. Explain Entry and Exit Controlled Constructs in detail with suitable illustrations.

6.i) What is an array? Write a C program to arrange the given 10 numbers in ascending order using one dimensional array.

ii) Write a C program to multiply two 3\*3 matrices. (NOV/DEC 2017)

7. Explain the Characteristics and Classification of an array with examples.

8. a) Write a C program to find the largest and smallest element in an array.

b) Write a C Program to insert an element in a specified position in a given array.

9. a) Write a C program to reverse a String. (JAN 2014)

b) Write a C program to check whether the given string is palindrome or not.

10. Explain various String operations. Write a C Program to find length of a String without using Library functions. (or) Explain the concepts of strings in detail.

## UNIT II FUNCTIONS, POINTERS, STRUCTURES AND UNIONS

### PART A

#### 1. What is a Pointer? How a variable is declared to the pointer?

Pointer is a variable which holds the address of another variable.

**Pointer Declaration:** datatype \*variable-name;

**Example:**

```
int *x, c=5;
```

```
x=&a;
```

#### 2. What is the purpose of Unions in C? (MAY-2014)

✓ The purpose of union is to save memory by using the same memory region for storing different objects at different times.

✓ Unions provide an efficient way of using the same memory location for multi-purpose.

#### 3. Write the syntax for including functions in a program? (or) What are the components of a function? (APRIL/MAY 2017)

The syntax for including functions in program is

```
return_type function_name(datatype var1, datatype var2,...); //FUNCTION  
DECLARATION
```

```
int main()
```

```
{ .....
```

```
variable_name = function_name(var1, var2, ...); //FUNCTION CALL
```

```
..... return 0; }
```

```
return_type function_name(datatype var1, datatype var2,...) //FUNCTION DEFINITION
```

```
{ .....
```

```
statements return(variable); }
```

#### 4. What are the uses of Pointers? (or) What are the advantages of using pointers in a



## program? (NOV/DEC 2017)

- ✓ Pointers are used to return more than one value to the function.
- ✓ Pointers are more efficient in handling the data in arrays
- ✓ Pointers reduce the length and complexity of the program
- ✓ They increase the execution speed
- ✓ The pointers saves data storage space in memory

## 5. What is dangling pointer?

In C, a pointer may be used to hold the address of dynamically allocated memory after this memory is freed with the free() function, the pointer itself will still contain the address of the released block. This is referred to as a dangling pointer. Using the pointer in this state is a serious programming error. Pointer should be assigned NULL after freeing memory to avoid this bug.

## 6. What is Pointer Arithmetic?

A pointer is an address, which is a numeric value. Therefore, you can perform arithmetic operations on a pointer just as you can on a numeric value. There are four arithmetic operators that can be used on pointers: ++, --, +, and -.

### Example:

```
int i = 12, *ip = &i; ip = ip + 1
```

## 7. What is meant by Recursive function?

If a function calls itself again and again, then that function is called recursive function.

## 8. What is a void pointer?

A void pointer is a C convention for "a raw address." The compiler has no idea what type of object a void pointer "really points to." `int *ip; ip-> points to an int. void *p; p-> doesn't point to a void!`

## 9. Define Static Storage Class (Dec 2014)

Storage class specifier static specifies that the declared object can be used both in local scope and global scope. Static variable/object has a value throughout the execution of the program and the object will be saved in main memory Example: `static int a;`

## 10. What is built-in functions?

The standard library functions are built-in functions to handle tasks such as mathematical computations, I/O processing, string handling etc. Built-in function is a set of code that takes a finite number of input and optionally returns a value. These functions are defined in the header file. Functions that operate on string expression are classified as string functions, they include functions for finding the length of a given text, remove certain words from a text etc.

## 11. What is a structure? Or Define Structure. (MAY 2015) (NOV/DEC 2017)

- ✓ It's a User defined datatype
- ✓ Can hold many data objects of different data types (heterogeneous) may contain the integer elements, float elements and character elements. etc.
- ✓ Collection of variables under single name
- ✓ Can conveniently used to represent a record



## 12. Define structure declaration.

Variables/constants for structure types can be declared at definition or after definition.

8

[storage class specifier] struct named\_Structure-type identifier name [=initialization list]; struct key word mandatory A structure must end with a semicolon

**Example:** Declare variables Book1, Book2 of type Books struct Books Book1; struct Books Book2;

## 13. Give syntax for structure definition (NOV/DEC 2017)

Syntax:

```
[storage class specifier][data type] struct [structure name]
{ Data _type member_name[, member name 2,..]; Data _type member_name[, member
name 2,..];
```

}[variable name]; Example:

```
struct Books
```

```
{ char title[50]; char author[50]; char subject[100]; int book_id;
} book;
```

## 14. Write the rules for declaring a structure?

- ✓ A structure must end with a semicolon
- ✓ Struct key word mandatory
- ✓ Each structure member must be terminated.
- ✓ The structure variable must be accessed with dot(.) operator.
- ✓ Structure declaration list (structure members):
  1. Can have char, float, double, int, array[], pointer\* other structure types
  2. Cannot have void, function type, same structure instance
  3. Can have pointer to an instance of itself which is called as self referential structures.

## 15. How structure elements can be accessed?

Structure members can be accessed using

1. Direct member access operator/dot operator
  - ✓ Represented as (.)
  - ✓ It's a Binary Operator
  - ✓ **Syntax:** Structure\_name.structure\_member\_name
2. Indirect member access operator/arrow operator
  - ✓ Represented as ->
  - ✓ It is to access structure members by the pointer to the structure
  - ✓ **Syntax:** Pointer\_to\_Structurestructure\_member\_name

## 16. Define nested structure.

- ✓ Structure within another structure is called as nested structure
- ✓ Used to create complex datatypes
- ✓ Nested Structures contain declaration of members of other structure types
- ✓ Structures can also be defined within declaration list of another structure
- ✓ Double braces used to initialize nested structure objects
- ✓ Example: emp={name,salary,{date of birth}};



**Accredited by NAAC**

- ✓ It is used to increase the readability of the program by reducing the complexity.

### 17. Give the rules for defining preprocessor.

- ✓ # pound symbol used before preprocessor directive
- ✓ # must be first character in source file or first non white space character in a line
- ✓ New line character ends preprocessor directive
- ✓ Only single space/tab space allowed between preprocessing tokens
- ✓ Can appear anywhere in program but generally placed in beginning of program
- ✓ The preprocessor cannot have termination with semicolon.

### 18. Differentiate between structure and union. (NOV/DEC-2018)

Structures	Union
1. The keyword struct is used to define a structure	1. The keyword union is used to define a union.
2. Each member within a structure is assigned separate storage area of location.	2. Memory allocated is shared by individual members of union.
3. The address of each member will be different	3. The address is same for all the members
4. Altering the value of a member will not affect other members of the structure.	4. Altering the value of any of the member will alter other member values.
5. Individual member can be accessed at a time	5. Only one member can be accessed at a time.
6. Several members of a structure can initialize at once.	6. Only the first member of a union can be initialized.

### 19. Distinguish between call by value and call by reference?(NOV/DEC 2017)

Call By Value	Call By Reference
In call by value, a copy of actual arguments is passed to formal arguments of the called function and any change made to the formal arguments in the called function have no effect on the values of actual arguments in the calling function.	In call by reference, the location (address) of actual arguments is passed to formal arguments of the called function. This means by accessing the addresses of actual arguments we can alter them within from the called function
In call by value, actual arguments will remain safe, they cannot be modified accidentally.	In call by reference, alteration to actual arguments is possible within from called function; therefore the code must handle arguments carefully else you get unexpected results.

### 20. Define Preprocessor Directives and list out few examples.



## Accredited by NAAC

Preprocessor is controlled by directives (commands) known as Preprocessor Directives

✓ Preprocessor directives are not part of C language It consists of various preprocessing tokens

10

✓ Begins with pound symbol(#) Examples:

```
#include          #if          #pragma          #endif
#define          #else          #ifdef #error #undef #elif #ifndef
```

### 21. What is the use of #define processor? (DEC 2014)

#define directive is used to define Macros –which are tokens that can be replaced for user defined sequence of characters

#### Syntax:

```
#define macro-name replacement-list
```

#### Example:

```
#define PI 3.14
```

### 22. What does #include<header\_name> do? How is it possible to tell the pre-processor where to look for header files? (Nov/Dec 2019)

Header files serve two purposes.

- System header files declare the interfaces to parts of the operating system. You include them in your program to supply the definitions and declarations you need to invoke system calls and libraries.
- Your own header files contain declarations for interfaces between the source files of your program. Each time you have a group of related declarations and macro definitions all or most of which are needed in several different source files, it is a good idea to create a header file for them.

## PART B

1. Write a C program to get two numbers and exchange these numbers using pass by value and pass reference (Nov/Dec 2018) (April/May 2017)

2. i) Write a c program to generate Fibonacci series using function

ii) Write a C Program to find factorial of a given number using recursive function.

3. i) Write a C program to swap the content of two variables using pointers.

ii) Write a C program to read integers into an array and reversing them using pointers.

4. What is Pointer? How to pass pointer as an argument in function?

5. Explain in detail the concept of pointer in C language. (April/May 2017)

6. Explain in detail the concept and importance of structures with example program in C language. (April/May 2017)

7. i) Write a C program to get 10 student details using structures from the user and display these details on the screen. (Nov/Dec 2018)

ii) Write a C program to get a 4 digit number and display the reverse of the same number.

8. Explain a C program that demonstrates array of pointers.

9. i) Write short notes on nested structure / Explain the concept of structure within structure



with suitable program.

ii). Define and declare a structure to store date, which including day, month and year.

11

10. Define a structure called book with book name, author name and price. Write a C program to read the details of book name, author name and price of 200 books in a library and display the total cost of the books and book details whose price is above Rs.500. (May 2015)

## UNIT III LINEAR DATA STRUCTURES

### PART A

#### 1. Define data structure. (Nov/Dec 2012) (Nov/dec-2016)

The organization, representation and storage of data is called the data structure. Since all programs operate on data, a data structure plays an important role in deciding the final solution for the problem.

#### 2. Write any two data structures used in Operating System?

Linear list

1. Tree data structure.

#### 3. Define ADT (Abstract Data Type)? What are operations of ADT? What are all not concerned in an ADT? (Nov/Dec 2012) (Nov/Dec 2017)

ADT may be defined as a "class of objects whose logical behaviour is defined by a set of values and a set of operations. Abstract data types or ADTs are a mathematical specification of a set of data and the set of operations that can be performed on the data.

#### Operations of ADT:

- Union, Intersection, size, complement and find are the various operations of ADT.
- The definition of an ADT is not concerned with the implementation details at all.
- It may not even be possible to implement a particular ADT on a particular piece of hardware or using a particular software system.

#### 4. Define linear data structure?

Linear data structures are data structures having a linear relationship between its adjacent elements. A linear data structure traverses the data elements sequentially, in which only one data element can directly be reached. **Example:** Arrays, Linked Lists

#### 5. What is meant by list ADT?

List or sequence is an abstract data type that represents a sequence of values, where the same value may occur more than once. List ADT is a sequential storage structure. General list of the form  $a_1, a_2, a_3 \dots a_n$  and the size of the list is 'n'. Any element in the list at the position  $i$  is

defined to be  $a_i$ ,  $a_{i+1}$  the successor of  $a_i$  and  $a_{i-1}$  is the predecessor of  $a_i$ .

#### 6. Define non-linear data structure.

Data structure which is capable of expressing more complex relationship than that of physical adjacency is called non-linear data structure. The elements of data structure do not form a



sequence or a linear list. **Example:** Trees, BST (Binary Search Trees) etc.

### 7. What is a DEQUEUE? (May/June 2013)

DeQueue or double ended Queue is a data structure in which elements may be added to or deleted from the front or the rear. Like an ordinary queue, a double-ended queue is a data structure it supports the following operations: enq\_front, enq\_back, deq\_front, deq\_back, and empty. Dequeue can be behave like a queue by using only enq\_ front and deq\_ front, and behaves like a stack by using only enq\_ front and deq\_ rear.

### 8. What is the use of header pointer and Null pointer in a linked list?

A linked list contains a pointer, referred as the head pointer, which points to the first node in the list that stores the address of the first node of the list. The final node in the linked list does not point to a next node. If link does not point to a node, its value is set to NULL. NULL pointer is often written 0 it is present in the address field of last node in the list

### 9. What are the advantages and disadvantages of a singly linked list?

Advantage	Disadvantage
<input type="checkbox"/> Insertions and Deletions can be done easily.	<input type="checkbox"/> It requires more space as pointers are also stored with information.
<input type="checkbox"/> It does not need movement of elements for insertion and deletion.	<input type="checkbox"/> Different amount of time is required to access each element.
<input type="checkbox"/> Its space is not wasted as we can get space according to our requirements.	<input type="checkbox"/> If we have to go to a particular element then we have to go through all those elements that come before that element.
<input type="checkbox"/> Its size is not fixed.	<input type="checkbox"/> We cannot traverse it from last & only from the beginning.
<input type="checkbox"/> It can be extended or reduced according to requirements.	<input type="checkbox"/> It is not easy to sort the elements stored in the linear linked list.
<input type="checkbox"/> Elements may or may not be stored in consecutive memory available, even then we can store the data in computer.	
<input type="checkbox"/> It is less expensive.	

### 10. Write a procedure to insert an element in the beginning of a list ADT implemented using a singly linked list. (April / May 2017)

Insertion is a three step process –

- ✓ Create a new Link with provided data.
- ✓ Point New Link to old First Link.
- ✓ Point First Link to this New Link.

//insert link at the first location

```
void insertFirst(int key, int data){ //create a link
struct node *link = (struct node*) malloc(sizeof(struct node)); link->key = key;
link->data = data;
//point it to old first node link->next = head;
//point first to new first node
head = link; }
```



## Accredited by NAAC

### 11. List the advantages in the linked list implementation of stack. (April / May 2017)

- ✓ Stack using linked list is a dynamic data structure so it can grow and shrink at runtime by allocating and de-allocating memory.
- ✓ Insertion and deletion of nodes in stack are really easier.
- ✓ As size of linked list can increase or decrease at run time so there is no memory wastage.

Data structures such as stack and queues can be easily implemented using linked list. Implementing a stack using a linked list is particularly easy because all accesses to a stack are at the top.

### 12. Define Stack. What are the operations of a stack (Nov/Dec 2012) (Nov/Dec 2018)

Stack is a LIFO (Last In First Out) data structure. A stack is an ordered collection of elements in which insertions and deletions are restricted to one end. The end from which elements are added and/or removed is referred to as top of the stack.

#### Stack operations:

- ✓ Push()                       Top()
- ✓ Pop()                         IsEmpty()

### 13. What is a double linked list?

Doubly linked list is an advanced form of a singly linked list, in which traversal can be done both forward and backward direction. In this each node contains three fields namely,

1. Previous address field.
2. Data field.
3. Next address field.

The previous address field of a node contains address of its previous node. The data field stores the information part of the node. The next address field contains the address of the next node in the list.

### 14. Give some applications of stack. (Nov/Dec 2011)

1. Conversion of infix to postfix
2. Backtracking problem
3. Expression evaluation
4. Towers of Hanoi

### 15. What are the advantages and disadvantages of doubly linked list? Advantage:

- ✓ We can traverse in both direction i.e from starting to end as well as from end to starting.
- ✓ It is easy to reverse the linked list.
- ✓ If we are at a node, then we can go at any node. But in linked list, it is not possible to reach the previous node.

#### Disadvantage:

- ✓ It requires more space per node because extra field is required for pointer to previous node.
- ✓ Insertion and Deletion take more time than linear linked list because more pointer operations are required than linear linked list.

### 16. What is a circular queue?

The queue, which wraps around upon reaching the end of the array is called as circular queue. In a standard queue data structure re-buffering problem occurs for each dequeue operation. To solve this problem by joining the front and rear ends of a queue to make the queue as a



circular queue

Circular queue is a linear data structure. It follows FIFO principle.

- ✓ In circular queue the last node is connected back to the first node to make a circle.
- ✓ Circular linked list follow the First In First Out principle
- ✓ Elements are added at the rear end and the elements are deleted at front end of the queue
- ✓ Both the front and the rear pointers points to the beginning of the array.

14

17. Evaluate the value of the expression  $ab+c*d$ - using stack. (Nov/Dec 2014)

- |               |                   |                             |
|---------------|-------------------|-----------------------------|
| 1. Stack: a   | 3. Stack:         | 5. Stack:                   |
| Output:       | Output: a+b       | Output: (a+b)*c             |
| 2. Stack: a b | 4. Stack: (a+b) c | 6. Stack: ((a+b)*c) Output: |
| Output:       | d                 |                             |

Output:

7. Stack:

Output:  $((a+b)*c) - d$

18. What are the postfix and prefix forms of the expression  $A+B*(C- D)/(P-R)$ ? (Nov/Dec 2011)

Postfix form:  $ABCD-*PR-/+$

Prefix form:  $+A/*B- CD-PR$

19. Give some applications of queue. (Nov/Dec 2011)

Queues have many applications in computer systems. Most computers have only a singly processor, so only one user may be served at a time. Entries from other users are placed in a queue. Each entry gradually advances to the front of the queue as users receive their service. Queues are also used to support print spooling.

20. Define Queue. How the data are stored in queue. (Nov/Dec 2014) Queue is a FIFO (First In First Out) data structure. A Queue is an ordered collection of elements in which insertions are made at one end and deletions are made at the other end. The end at which insertions are made is referred to as the rear end, and the end from which deletions are made is referred to as the front end. In queue, the datas are inserted only at the front end of the queue and it is pointed using rear pointer. The data's can be accessed sequentially. The elements are to be deleted from the back end of the queue.

21. List the applications of linked list. (Nov/dec-2016)

- ✓ Linked lists are used to implement stacks, queues, graphs, etc.
- ✓ Linked lists let you insert elements at the beginning and end of the list.

In Linked Lists we don't need to know the size in advance

22. What are the advantages of using doubly linked list over singly linked list?

The advantage of using doubly linked list is, it uses the double set of pointers. One pointing to the next item and other pointing to the proceeding item. This allows us to traverse the list in either direction.

23. Define a heap. How can it be used to represent a priority queue? (Nov/Dec 2017)

Heap data structure is a specialized binary tree based data structure. Here, the nodes are



## Accredited by NAAC

arranged based on their value. A heap data structure, sometimes called as Binary Heap. Heap is generally preferred for priority queue implementation because heaps provide better performance compared to arrays or linked list. When a priority queue is implemented using a heap, the worst-case times for both insert and removeMax are logarithmic in the number of values in the priority queue.

15

## PART B

1. Explain the applications of stack with an example.
- 2.i) Describe Hash function and its various techniques. (May/June 2014)  
ii) Explain heap data structure along with its properties. (Nov/Dec 2013)
3. Implement insertion, deletion and search operations in singly linked list.
4. Explain the operation performed on QUEUE in detail. Write C program to implement these queue operations. (Nov/Dec 2016, May/June 2013 )
5. Design an algorithm to reverse the linked list. Trace it with an example. And Define an efficient representation of two stacks in a given area of memory with n words and explain.
6. Explain linear linked implementation of stack and queue.
- 7.(i) Write a procedure to perform polynomial addition using linked lists.  
(ii) Write an algorithm for inserting and deleting an element from Doubly Linked List.
8. Develop procedures to perform push and pop operation on an array implementation of stack and use the same to evaluate a postfix expression  $1\ 2\ 3\ * -\ 4\ +$ .
9. Design an algorithm to convert an infix expression to postfix expression using stacks and apply to the expression  $(a + b - d * e + (f * g + h) * i)$ . (April / May 2017)
10. Write a C program to perform all the operations of the stack and Explain the evaluation of the expression using the stack in Infix notation.

## UNIT IV NON-LINEAR DATA STRUCTURES

### PART A

#### 1. Define tree.

A tree is an Abstract Data Type (ADT) or data structure which represents hierarchical



relationship between individual data items, with a root value and sub trees of children, represented as a set of linked nodes. Tree is a non-linear data structure.

## 2. Define height or depth of a tree.

The depth of a node is the number of edges from the node to the tree's root node. A root node will have a depth of 0. The height of a node is the number of edges on the longest path from the node to a leaf. A leaf node will have a height of 0.

## 3. Define Binary tree. What are the two ways of representing binary tree?(Nov/Dec-2016)

Binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child. A binary tree is a finite set of nodes which is either empty or consists of a root and two disjoint binary trees called the left sub tree and the right sub tree.

The two representations are,

- ✓ Sequential representation
- ✓ Linked representation

## 4. Define a full binary tree. State some properties of a binary tree.

A full binary tree is a binary tree in which all the leaves are on the same level and every non-leaf node has exactly two children. A full binary tree of a given height  $h$  has  $2^h - 1$  nodes. The properties of a binary tree are as follows

- ✓ A binary tree of  $n$  elements has  $n-1$  edges
- ✓ Binary tree of height  $h$  has at least  $h$  and at most  $2^h - 1$  elements.
- ✓ The maximum number of nodes on level  $n$  of a binary tree is  $2^{n-1}$ , where  $n \geq 1$ .

## 5. Define Traversal. Give the types of traversal.

One of the most important operations performed on a binary tree is its traversal. Traversing a binary tree means moving through all the nodes in the binary tree, visiting each node in the tree exactly once. There are three different traversal of binary tree.

- ✓ In order traversal
- ✓ Post order traversal
- ✓ Pre order traversal

## 6. How to perform union operation? (Nov/Dec 2014)

If  $a$  &  $b$  are in two different equivalence sets, then

- Union ( $a$   $b$ ) should merge those two sets into one Otherwise, no change.
- Union by height: Making the shallow tree a sub tree of the deeper tree.

Union by size: Making the smaller tree a sub tree of the larger.

## 7. Give the code for pre order traversal.(Nov/Dec 2011)

```
void preorder(struct tree* t)
{ if (t==NULL)
return; else{
cout<<"\t"<<t->data;
preorder(t->left); preorder(t->right);} }
```



Accredited by NAAC

### 8. What are the applications of binary tree? (Nov/Dec 2012)

Binary tree is used in data processing.

- a. File index schemes
- b. Hierarchical database management system

### 9. Define equivalence relation. (Nov/Dec 2012)

Equivalence relation is the relation that holds between two elements if and only if they are members of the same cell within a set that has been partitioned into cells such that every element of the set is a member of one and only one cell of the partition. The intersection of any two different cells is empty; the union of all the cells equals the original set. These cells are formally called equivalence classes.

Equivalence relation R is a relation that satisfies the following properties

**Reflexive-** a R a for all a belongs to S

**Symmetric-** a R b implies b R a

**Transitive-** a R b & b R c implies a R c

### 10. Write short notes on connected components? (Nov/Dec 2014)

A connected component of an undirected graph is a maximal connected sub graph of the graph. Every vertex of the graph lies in a connected component that consists of all the vertices that can be reached from that vertex, together with all the edges that join those vertices. If an undirected graph is connected, there is only one connected component.

### 11. Define a weighted graph.

A graph is said to be weighted graph if every edge in the graph is assigned some weight or value. The weight of an edge is a positive value that may be representing the distance between the vertices or the weights of the edges along the path.

### 12. What is a strongly connected graph and weakly connected graph?

A directed graph is said to be a strongly connected graph if for every pair of distinct vertices there exists a directed path from every vertex to every other vertex. It is also called as Complete Graph. A directed graph is said to be a weakly connected graph if any vertex doesn't have a directed path to any other vertices. The nodes in a weakly connected digraph therefore must all have either out degree or in degree of at least 1.

### 13. Give the code for post order traversal.

```
void postorder(struct tree* t)
{
if(t==NULL)
return; else{
postorder(t->left); postorder(t->right); cout<<"\t"<<t->data;
}
}
```

### 14. Differentiate between trees and graphs? (Nov/Dec 2014)

Features	Trees	Graph
----------	-------	-------



Accredited by NAAC

<b>Path</b>	Tree is special form of graph i.e. minimally connected graph and having on y one path between any two vertices.	In graph there can be more than one path i.e. graph can have <u>18</u> uni-directional or bi- directional (edges) between nodes.
<b>Model</b>	Tree is a hierarchical model.	Graph is a network model.
<b>Applications</b>	Tree applications: sorting and searching like Tree Traversal & Binary Search.	Graph applications : Coloring of maps, in OR (PERT & CPM), algorithms, Graph coloring, job scheduling, etc.

**15. What is a graph? List the two ways to represent a graph. What are the different ways to represent graph**

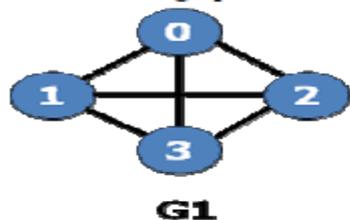
Graph is a non-linear data structure that represents less relationship between its adjacent elements. There is no hierarchical relationship between the adjacent elements in case of graphs.

- A Graph  $G=\{V, E\}$  consists of two sets  $V$  &  $E$ , where  $V$  is a set of vertices and  $E$  is a set of Edges. The set  $E$  is set of pair of vertices. So each is a pair  $(v,w)$  where  $v,w$  is an element of  $V$ .

**Two ways of representing a graph are:**

- Adjacency matrix
- Adjacency list

Consider the graph below:



Adjacency matrix representation is

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Adjacency list representation is

**16. Find the maximum number of nodes in complete binary tree if d is the depth? How many trees are possible with 3 nodes? (Nov/Dec 2014)**

The maximum number of nodes in a binary tree of depth  $k$  is  $2^{k+1}-1, k \geq 1$ . If there are  $n$  nodes, there exist  $2n-n$  different trees.  $(2*3)-3=3$  trees

**17. Define path in a graph?**

The path in a graph is the route taken to reach terminal node from a starting node. A path in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. A path in a diagram in which the edges are distinct is called a simple path.

**18. What is an acyclic graph?**

A path which originates and ends in the same node is called a cycle or circuit. A simple

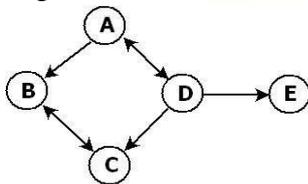
diagram which does not have any cycles is called an acyclic graph. An acyclic graph does not contain any cycles. Trees are connected acyclic undirected graphs. Directed acyclic graph are called DAG.

**19. List out the applications of depth—first search(APRIL/MAY 2016) (Nov/Dec 2017)**

- ✓ Detecting cycle in a graph  Topological Sorting
- ✓ Path Finding  To test if a graph is bipartite

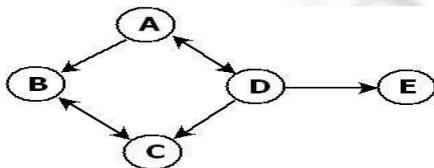
**20. Define out degree and outdegree of a Graph.( Nov/Dec 2012)**

In a directed graph, for any node v, the number of outgoing edges from v, are called out degree of a node .



Out degree of A → 2 Out degree of B → 1 Out degree of C → 1 Out degree of D → 3 Out degree of E → 0

In a directed graph, for any node v, the number of incoming edges to v, are called in degree of a node v.



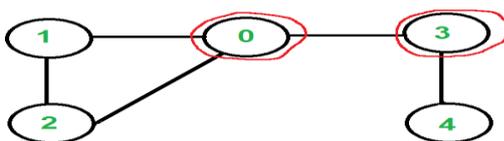
In degree of A → 2 In degree of B → 1 In degree of C → 1 In degree of D → 2 In degree of E → 1

**21. What is a spanning tree? Name two algorithms to find minimum spanning Tree**

A spanning tree of a graph is just a sub graph that contains all the vertices and is a tree. A graph may have many spanning trees. Spanning tree of the graph is a connected sub graph in which there are no cycles. The two algorithms are Kruskal’s algorithm, Prim’s algorithm

**22. Define articulation point in a graph. (Nov/Dec 2012)**

The articulation point is the point at which the removal of the vertex will split the graph. Articulation points represent vulnerabilities in a network – single points whose failure would split the network into 2 or more disconnected components.

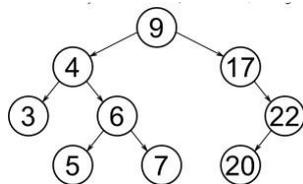


Articulation points are 0 and 3

**23. Define Binary searchtree (Nov/Dec 2018)**

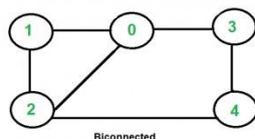
A binary search tree (BST), is a node-based data structure in which each node has no more than two child nodes. The left sub-tree contains only nodes with keys less than the parent node. The right sub-tree contains only nodes with keys greater than the parent node.

Example:

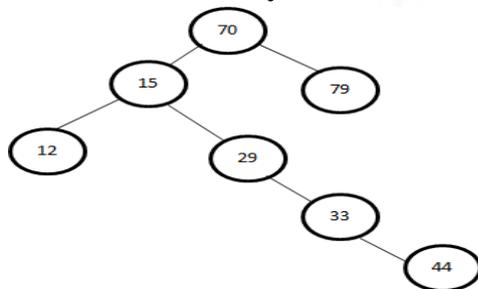


**24. When a graph is said to be bi connected? (Nov/Dec 2017)**

An undirected graph is called Biconnected if there are two vertex-disjoint paths between any two vertices. In a Biconnected Graph, there is a simple cycle through any two vertices.



**25. Draw the binary search tree for the following inputs: 70,15,29,33,44,12,79**



## PART B

1. Explain the various tree traversal methods with algorithm and predict a binary tree with Preorder: ABCDEFGHI and Inorder: BCAEDGHF (**April/May**)
2. How can binary trees be represented using arrays? Write algorithm for finding  $k^{\text{th}}$  element, inserting and deleting an element in binary tree with example program. (**Nov/Dec 2017, April / May 2017, Nov/Dec 2011**)
3. Write routines to convert a set as a tree and perform find operation on the elements. Illustrate with examples. (**April / May 2017**)
4. With necessary algorithms, explain the two graph traversal methods. Demonstrate with examples.
5. Write procedures to count the number of connected components and the number of vertices in each component of a graph. Illustrate with examples. (**April / May 2017**)
6. Explain in detail about the smart union algorithm.
7. State and explain the dynamic equivalence problem.
  - (i) Explain the path compression algorithm and analyze the Union/Find algorithm used.
8. Explain in detail the implementation of Binary Search Tree and perform its operations. Draw a binary search tree for the input 60,25,75,15,50,66,33,44. Trace the algorithm to delete nodes 25,75,44 from the tree. (**May/June 2012**)
9. How can you construct an expression tree? Describe your answer with an example.
  - (i) Explain set representations and Union-Find operations with suitable examples.



10. Explain in detail about Biconnected Graph.

(i) Explain the different ways for representing the graph with examples

## UNIT V SEARCHING AND SORTING ALGORITHMS

### PART A

#### 1. What is meant by sorting and what are its classifications?(April/May 2015)

Ordering the data in an increasing or decreasing order according to some relationship among the data item is called sorting.

1. Internal sorting
2. External sorting

#### 2. What is meant by external sorting?

External sorting is a process of sorting in which large blocks of data stored devices are moved to the main memory and then sorted. Example: Merge sort.

#### 3. What is meant by internal sorting?

This method uses only primary memory during sorting process if all data to be sorted can be accommodated at the time in memory is called as internal sorting. Example: Heap sort, shell sort, quick sort.

#### 4. What are the various factors to be considered in deciding a sorting algorithm?

- ✓ Programming time
- ✓ Execution time of the program
- ✓ Memory needed for program environment

#### 5. What is the main idea in Bubble sort?

The basic idea underlying the bubble sort is to pass through the file sequentially several times. Each pass consists of comparing each element in the file with its successor ( $x[i]$  and  $x[i+1]$ ) and interchanging the two elements if they are not in proper order.

#### 6. What is the basic idea of quick sort?

Pick one element in the array, which will be the pivot. Make one pass through the array, called a partition step, re-arranging the entries so that: The pivot is in its proper place. Entries smaller than the pivot are to the left of the pivot. Entries larger than the pivot are to its right. Recursively apply quick sort to the part of the array that is to the left of the pivot, and to the right part of the array.

#### 7. What is the advantage of quick sort?

Quick sort reduces unnecessary swaps and moves an item to a greater distance, in one move. One of the fastest algorithms on average. Does not need additional memory

#### 8. What is complexity analysis?



It is the analysis of the amount of memory and time an algorithm requires to completion.

There are two types of Complexity

✓ **Space Complexity:** Space complexity of an algorithm is the amount of memory it needs to run to completion.

✓ **Time Complexity:** Time complexity is the amount of computer time an algorithm requires to run to completion.

## 9. What does asymptotic notation mean?

Asymptotic notations are terminology that is introduced to enable us to make meaningful statements about the time and. The different notations are

- 1) Big – Oh notation                      2) Omega notation                      3) Theta notation.

## 10. What are the techniques used to choose the pivot element for quick sort?

The various techniques are

1. First element    2. Random pick    3. Median of three partitioning

## 11. What is binary search?

Binary search is also a method used to locate a specified item in a sorted list. This method starts by comparing the searched element to the elements in the middle of the list. If the comparison determines that the two elements are equal the method stops and returns the position of the element. If the searched element is greater than the middle element, it starts the method again using only the bottom half of the sorted list. If the searched element is less than the middle element, it starts the method again using only the top half of the sorted list.

## 12. What is linear search?

Linear search is the simplest searching method, which checks each element in a list sequentially until it finds a specified element. The input to the linear search method is a sequence and the item that needs to be searched. The output is true if the specified item is within the provided sequence or false if it is not in the sequence. The complexity of linear search is  $O(n)$ .

## 13. What is divide and conquer technique?

Divide and Conquer algorithm is based on dividing the problem to be solved into several, smaller sub instances, solving them independently and then combining the sub instances solutions so as to yield a solution for the original instance.

## 14. What is dynamic programming?

Dynamic programming algorithm is a general class of algorithms which solve problems by solving smaller versions of the problem, saving the solutions to the small problems and then combining them to solve the larger problems.

## 15. What is the time complexity of Quick sort and Binary search?

Time complexity	Quick sort	Binary Search
Best case	$O(n \log n)$	$O(\log n)$
Average case	$O(n \log n)$	$O(\log n)$
Worst case	$O(n^2)$	$O(\log n)$

## 16. Define merge sort. How an array elements are sorted using merge sort?(Nov/Dec



2015)

Merge sort is a divide and conquer algorithm. Merge sort is the technique which has sorted sub arrays which are merged to form a single sorted array.

A merge sort works as follows:

- ✓ Divide the unsorted list into  $n$  sub lists, each containing 1 element
- ✓ Repeatedly merge sub lists to produce new sorted sub lists until there is only 1 sub list remaining. This will be the sorted list.

**17. What are the advantages of merge sort?**

- ✓ It can be applied to files of any size.
- ✓ Reading of the input during the run-creation step is sequential  Not much seeking.
- ✓ Reading through each run during merging and writing the sorted record is also sequential.

**18. State why quick sort is more efficient than merge sort. (Nov/Dec 2013)**

Theoretically, both quick sort and merger sort take  $O(n \log n)$  time and hence time taken to sort the elements remains same. However, quick sort is superior to merge sort in terms of space. Quick sort is in-place sorting algorithm whereas merge sort is not in-place. In-place sorting means, it does not use additional storage space to perform sorting. In merge sort, to merge the sorted arrays it requires a temporary array and hence it is not in-place. However time efficiency of the quick sort depends on the choice of the pivot element.

**19. What is time complexity?(April/May 2015)**

Time complexity of an algorithm computes the amount of time taken by an algorithm to run to its completion. Commonly it is represented using big  $O$  notation. Generally it is categorized as best case time complexity, worst case time complexity and average case time complexity. It is commonly estimated by counting the number of elementary operations performed by the algorithm.

**20. How to take care of an overflow problem in hashing.(Nov/Dec 2018)**

An overflow occurs when the home bucket for a new pair (key, element) is full.

We may handle overflows by:  $f$

- ✓ Search the hash table in some systematic fashion for a bucket that is not full.
  1. Linear probing(linear open addressing).
  2. Quadratic probing.
  3. Random probing
- ✓ Eliminate overflows by permitting each bucket to keep a list of all pairs for which it is the home bucket.
  1. Array linear list
  2. Chain

**21. What is insertion sort?**

Insertion sort iterates, consuming one input element each repetition, and growing a sorted output list. Each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remain.



## 22. What is performance analysis of an algorithm?

The analysis of the performance of an algorithm based on specification is called performance analysis. It is loosely divided into Priori estimates and posterior estimates

## 23. Give the algorithm to search an element in an array using linear search.

Linear search is implemented using following steps...

**Step 1:** Read the search element from the user

**Step 2:** Compare, the search element with the first element in the list. **Step 3:** If both are matching, then display “Given element found!!!” and terminate the function

**Step 4:** If both are not matching, then compare search element with the next element in the list.

**Step 5:** Repeat steps 3 and 4 until the search element is compared with the last element in the list.

**Step 6:** If the last element in the list is also doesn't match, then display “Element not found!!!” and terminate the function.

## 24. Differentiate linear search and binary search.

Linear search	Binary search
Linear search is easy but takes more time to search an element as it compare all element sequentially	Binary search it start searching from middle, if the searching element is not found in middle then it goes to left or right and vice versa. and hence take less time than linear search
Linear search requires $O(n)$ times	Binary search is very best in time and efficiency. It requires $O(\log n)$ times

## 25. List some algorithms uses Dynamic Programming.

- ✓ Matrix-chain multiplication
- ✓ Optimal Binary Search Tree
- 0-1knapsack problem
- All-pairs Shortest Path

## PART B

1. Define Heap Sort. Write the algorithm for heap sort procedures.
2. Write a C program to implement binary search and explain it with an example. Compute its complexity.
3. Write an algorithm to sort 'n' numbers using merge sort. Trace the algorithm for the numbers 38, 47, 23, 3, 9, 82, 10 and show the order in which they are processed. (Nov/Dec 2019)
4. Write a C program to sort the following set of numbers using a quick sort algorithm. Find the time complexity of the same. Give the trace of the algorithm for the



## Accredited by NAAC

following set of numbers. 44,33,11,55,77,90,40,60,99,22,88,66 (Nov/Dec 2018, Nov/Dec 2019)

25

5. List the various sorting techniques and explain the worst and best case time complexity of various sorting techniques. (Nov/Dec 2017)
6. Write a C code and explain the implementation of linear search algorithm with an example (Nov/Dec 2014, April/May 2015, Nov/Dec 2015)
7. Explain the insertion sort with its time complexity
8. What is Asymptotic Notation? Explain Asymptotic complexity classes.
9. Explain Shell Sort Algorithm with examples.
10. Explain in detail about various algorithm design techniques.

