



**VETRI VINAYAHA COLLEGE OF ENGINEERING AND TECHNOLOGY**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**B.E SECOND SEMESTER**  
**(REGULATION 2013)**



**CS 6202 PROGRAMMING AND DATA STRUCTURES I**

**UNIT I**

**C PROGRAMMING FUNDAMENTALS- A REVIEW**

Conditional statements – Control statements – Functions – Arrays – Preprocessor - Pointers -  
Variation in pointer declarations – Function Pointers – Function with Variable number of arguments

**PART A**

**1. Define global declaration?**

The variables that are used in more than one function throughout the program are called global variables and declared in outside of all the function that is before the main ( ) function.

**2. Define data types?**

Data type is the type of data, which are going to access within the program.

C supports different data types:

- Primary – char, int, float, double
- User defined- typedef
- Derived – array, pointers, structure, union
- Empty - void

**3. Define variable with example?**

A variable is an identifier and it may take different values at different times of during the execution .

A variable name can be any combinations of 1 to 8 alphabets, digits or underscore.

Example: int a,b; here a,b are variables

**4. What is decision making statement?**

Decision making statement is used to break the normal flow of the program and execute part of the statement based on some condition.

**5. What are the various decision making statements available in C ?**

□

If statement □, if...else statement □, nested if statement □,if...else ladder statement ,switch statement

**6. What is an array?**

An array is a collection of data of same data type. The elements of the array are stored in continuous memory location and array elements are processing using its index.

Example: int a[10];

Here a is an array name.

## 7. What is two dimensional array?

Two dimensional is an array of one dimensional array. The elements in the array are referenced with help of its row and column index.

Example: `int a[2][2];`

## 8. Define is function?

Functions are group of statements that can be perform a task. Function reduces the amount of coding and the function can be called from another program.

Example:

```
main()  
{  
-----  
fun ();  
---  
}
```

## 9. What are the various looping statements available in 'C'?

- While statement
- Do...while statement
- For statement

## 10. What are the uses of Pointers?

- 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 save data storage space in memory

## 11. What is a Pointer? How a variable is declared to the pointer? (MAY 2009)

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

Pointer Declaration: `datatype *variable-name;`

Example: `int *x, c=5; x=&a;`

## 12. Define pre-processor in C.

Preprocessor are used to link the library files in to source program, that are placed before the main() function and it have three preprocessor directives that are Macro inclusion, Conditinal inclusion, File inclusion .

## 13. Define recursive function?

A function is a set of instructions used to perform a specified task which repeatedly occurs in the main program. If a function calls itself again and again, hten that function is called recursive function.

## 14. What is the difference between while and do....while statement?

The while is an entry controlled statement. The statement inside the while may not be executed at all when the condition becomes false at the first attempt itself. The do ...while is an exit controlled statement. The statements in the block are executed at least once.

## 15. Define Operator with example?

An operator is a symbol that specifies an operation to be performed on operands . some operators require two operands called binary operators, while other acts upon only one operand called unary operator. Example:  $a+b$  here a,b are operands and + is operator.

## 16. Define conditional operator or ternary operator?

Conditional operator itself checks the condition and execute the statement depending on the condition.  $(a>b)?a:b$  if a is greater than b means the a value will be return otherwise b value will be return.

Example:  $big=a>b?a:b;$

## 17. Give the syntax for the 'for' loop statement

for (Initialize counter; Test condition; Increment / Decrement)

```
{
statements;
}
```

- Initialization counter sets the loop to an initial value. This statement is executed only once.
- The test condition is a relational expression that determines the number of iterations desired or it determines when to exit from the loop.
- The increment / decrement parameter decides how to make changes in the loop.

## 18. List the header files in 'C' language.

<stdio.h> contains standard I/O functions  
<ctype.h> contains character handling functions  
<stdlib.h> contains general utility functions  
<string.h> contains string manipulation functions  
<math.h> contains mathematical functions  
<time.h> contains time manipulation functions

### 19. What are steps involved in looping statements?

- Initialization of a condition variable. □
- Test the control statement. □
- Executing the body of the loop depending on the condition. □
- Updating the condition variable. □

### 20. Define break statement?

The break statements is used terminate the loop. When the break statement is encountered inside a loop, the loop is immediately exited and the program continues with the statement immediately following the loop.

### 21. Define null pointer?

A pointer is said to be a null pointer when its right value is 0, a null pointer can never point to a valid data. For checking a pointer, if it is assigned to 0, then it is a null pointer and is not valid

Example:

```
int *a; int *b; a=b=0;
```

## PART B

1. Explain about the various decision making statements in C language.
2. Explain the control statements in c.
3. What are functions? Explain the types of functions in detail with an example program for each type.
4. Define arrays. Explain the array types with an example program for each type
5. What are pointers? When and why they are used? Explain in detail with sample programs.
6. Describe in detail about the Preprocessors in C.
7. Explain function pointers with example.
8. Detailly explain about function with variable number of arguments?

## UNIT II

### C PROGRAMMING ADVANCED FEATURES

Structures and Unions - File handling concepts – File read – write – binary and Stdio – File Manipulations

## PART A

### 1. Define Structure in C.

A structure contains one or more data items of different data type in which the individual elements can differ in type. A simple structure may contain the integer elements, float elements and character elements etc. and the individual elements are called members.

Example:

```
struct result  
{  
int marks;  
float avg;
```

```
char grade;  
}std;
```

## 2. Rules for declaring a structure?

- A structure must end with a semicolon.
- Usually a structure appears at the top of the source program.
- Each structure element must be terminated.
- The structure must be accessed with structure variable with dot (.) operator.

## 3. Define structure pointers.

Pointer is a variable, it contains address of another variable and the structure pointers are declared by placing \* in front of a structure variable's name.

Example:

```
struct result  
{  
int marks;  
float avg; char grade;  
};  
struct result *sam;
```

## 4. Define union.

A union is a collection of variables of different types, just like structure. Union is a derived data type and the difference between union and structure is in terms of storage. In structure each member has its own storage location, whereas all the members of union use the same memory location.

Example:

```
union result  
{  
int marks;  
float avg;  
char grade;  
}std;
```

## 5. Define file.

A file is a collection of bytes stored on a secondary storage device, which is generally a disk of some kind. The collection of bytes may be interrupted, for example, as characters, words, lines, paragraph and pages from a textual document.

Example:

```
FILE *infile;  
FILE *outfile;
```

## 6. Define binary files.

Binary files can be processed sequentially or, depending on the needs of the application, they can be processed using random access techniques. In C, processing a file using random access techniques involves moving the current file position to an appropriate place in the file before reading or writing data.

## 7. Define opening a file.

A file requires to be opened first with the file pointer positioned on the first character. No input-output functions on a stream can be performed unless it is opened. When a stream is opened, it is connected to named DOS device or file. C provides a various functions to open a file as a stream. Syntax: FILE \*fopen(char \* filename, char \*mode);

## 8. What is meant by fseek()?

fseek() will position the file pointer to a particular byte within the file. The file pointer is a pointer is a parameter maintained by the operating system and determines where the next read will comes from , or to where the next write will go.

## 9. List out the file handling functions

- fopen()-create a new file or open a existing file fclose-close a file
- getc()-reads a character from a file putc()-writes a character to file fscanf()-reads a set of data from a file

## 10. What is meant by structure template?

A structure template contains definition of a structure name and a list of members along with data types.

## 11. What are nested structures?

A structure built within another structure is called nested structures or structure within a structure.

Example:

```
struct dob
{
int day;
int month;
int year;
};
struct student
{
int rno;
char name[20];
struct dob date;
};
```

## 12. What are the advantages of union over structure?

- Union save memory space compared to structures since the size of union is equal to its largest sized member.
- Unions are useful in situations where there is a need to use only one of its member elements at a time

## 13. What are file modes?

Modes tell about the type of operation to be performed on a file that is being opened.

There are three basic file modes:

- a. Read
- b. Write
- c. Append

#### 14. What are the steps for using a file?

The basic steps for using a file in C are as follows:

- a. Create a variable of type "FILE \*";
- b. Open the file using "fopen" function and assign the file to the variable.
- c. Check to make sure that the file was successfully opened by checking to see if the variable is equal to NULL.
- d. Perform file operations like read,write.

#### 15. What are the file pointers of stdlib.h?

There are three special file pointers.

1. stdin (Standard input)
2. stdout (Standard output)
3. stderr (Standard error)

#### 16. What are the features of binary files?

- Any structure can be used in the file.
- Contents can be changed anywhere in the file.

#### 17. What are the functions to access files randomly?

There are three functions to perform random access.

ftell() – tells the current position of a file

fseek() – moves to a desired location from the current position

rewind() – resets the file position to the beginning of a file.

#### 18. How do you reference a structure member?

The structure members cannot be accessed directly. They can be accessed by using the name of the structure variable followed by a dot(.) and then the name of the member variable.

```
struct student
{
int rno;
int age;
char name[20];
}s1;
s1.rno=223;
s1.age=17;
s1.name="john";
```

#### 19. What is structure assignment?

The value of one structure variable is assigned to another variable of same type using assignment statement. If the e1 and e2 are structure variables of type employee then the structure assignment can be performed as e1=e2; this assigns value of structure variable e2 to e1.

### PART B

1. Explain the different modes in which a file can be opened.
2. Explain the functions fscanf, fprintf, fgets, fputs, getw, putw, getc, putc, fread, fwrite, feof, ferror, fseek, ftell, rewind with example.
3. Explain structures with an example
4. Explain union with an example
5. Discuss about the file handling concepts in C
6. Explain different ways to read data from a file.
7. Explain different ways to write data to a file
8. Design a c program to create the employee database and to perform the manipulations such as adding a record, deleting a record, updating a record.
9. Explain about structure within structure with example.

## UNIT III

### LINEAR DATA STRUCTURES – LIST

Abstract Data Types (ADTs) – List ADT – array-based implementation – linked list implementation — singly linked lists- circularly linked lists- doubly-linked lists – applications of lists –Polynomial Manipulation – All operation (Insertion, Deletion, Merge, Traversal)

#### PART - A

##### 1. What is meant by an abstract data type?

An ADT is an object with a generic description independent of implementation details. This description includes a specification of an components from which the object is made and also behavioral details of objects.

##### 2. Advantages and Disadvantages of array?

Advantages:

- Data accessing is faster
- Array is simple-in terms of understanding point and in terms of programming

Disadvantages:

- Array size is fixed
- Array elements stored simultaneously
- Insertion and deletion of elements in an array is difficult.

##### 3. What is an array?

Array may be defined abstractly as a finite ordered set of homogenous elements. Finite means there is a specific number of elements in the array.

##### 4. What is a linked list?

Linked list is a kind of series of data structures, which are not necessarily adjacent in memory. Each structure contains the element and a pointer to a record containing its successor.

##### 5. What is singly linked list?

A singly linked list is a linked list, there exists only one link field in each and every node and all nodes are linked together in some sequential manner and this type of linked list is called singly linked list.

##### 6. What is a doubly linked list?

In a simple linked list, there will be one pointer named as NEXT POINTER to point the next element, where as in a doubly linked list, there will be two pointers one to point the next element and the other to point the previous element location.

##### 7. Define double circularly linked list?



In a doubly linked list, if the last node or pointer of the list, point to the first element of the list, then it is a circularly linked list.

### 8. What is the need for the header?

Header of the linked list is the first element in the list and it stores the number of elements in the list. It points to the first data element of the list.

### 9. Define Polynomial ADT

A polynomial object is a homogeneous ordered list of pairs  $\langle \textit{exponent}, \textit{coefficient} \rangle$ , where each coefficient is unique. Operations include returning the degree, extracting the coefficient for a given exponent, addition, multiplication, evaluation for a given input.  $10x^4+5x^2+1$

### 10. How to search an element in list.

Searching can be initiated from first node and it is compared with given element one after the other until the specified key is found or until the end of the list is encountered.

### 11. Define Dequeue?

Dequeue is also data structure where elements can be inserted from both ends and deleted from both ends. To implement a dequeue operations using singly linked list operations performed insert\_front, delete\_front, insert\_rear, delete\_rear and display functions.

### 12. How to implement stack using singly linked list?

Stack is an Last In First Out (LIFO) data structure. Here, elements are inserted from one end called push operation and the same elements are deleted from the same end called pop operation. So, using singly linked list stack operations are performed in the front or other way we can perform rear end also.

### 13. What are the types of Linear Linked List?

Circular Singly linked List

Circular Doubly linked list

### 14. What are advantages of Linked lists?

- Linked lists are dynamic data structures
- The size is not fixed
- Data can store non-continuous memory
- Insertion and deletions nodes are easier

### 15. How data structures are classified?

Data structures are classified into two categories based on how the data items are operated:

- i. Primitive data structure
- ii. Non-Primitive data structure
  - a. Linear data structure
  - b. Non-linear data structure

### 16. Mention the features of ADT.

- a. Modularity
  - i. Divide program into small functions
  - ii. Easy to debug and maintain

- iii. Easy to modify
- b. Reuse

- i. Define some operations only once and reuse them in future
- c. Easy to change the implementation

### **17. What are the ways of implementing linked list?**

The list can be implemented in the following ways:

- i. Array implementation
- ii. Linked-list implementation
- iii. Cursor implementation

### **18. Where cursor implementation can be used?**

The cursor implementation of lists is used by many languages such as BASIC and FORTRAN that do not support pointers. The two important features of the cursor implementation of linked are as follows:

- The data are stored in a collection of structures. Each structure contains data and a index to the next structure.
- A new structure can be obtained from the system's global memory by a call to cursorSpace array.

### **19. List down the applications of List.**

- a. Representation of polynomial ADT
- b. Used in radix and bubble sorting
- c. In a FAT file system, the metadata of a large file is organized as a linked list of FAT entries.
- d. Simple memory allocators use a free list of unused memory regions, basically a linked list with the list pointer inside the free memory itself.

### **20. What are the advantages of linked list?**

- a. Save memory space and easy to maintain
- b. It is possible to retrieve the element at a particular index
- c. It is possible to traverse the list in the order of increasing index.
- d. It is possible to change the element at a particular index to a different value, without affecting any other elements.

### **21. Mention the demerits of linked list**

- a. It is not possible to go backwards through the list
- b. Unable to jump to the beginning of list from the end.

### **22. What are the operations performed in list?**

The following operations can be performed on a list

- i. Insertion
  - a. Insert at beginning
  - b. Insert at end
  - c. Insert after specific node
  - d. Insert before specific node
- ii. Deletion
  - a. Delete at beginning
  - b. Delete at end
  - c. Delete after specific node
  - d. Delete before specific node
- iii. Merging
- iv. Traversal

## **PART B**

1. Explain the various operations of the list ADT with examples
2. Write the program for array implementation of lists
3. Write a C program for linked list implementation of list.
4. Explain the operations of singly linked lists
5. Explain the operations of doubly linked lists
6. Explain the operations of circularly linked lists
7. How polynomial manipulations are performed with lists? Explain the operations
8. Explain the steps involved in insertion and deletion into an singly and doubly linked list.

## **UNIT IV**

### **LINEAR DATA STRUCTURES – STACKS, QUEUES**

Stack ADT – Evaluating arithmetic expressions- other applications- Queue ADT – circular queue implementation – Double ended Queues – applications of queues

## **PART A**

### **1. What is a Stack ?**

A stack is a non-primitive linear data structure and is an ordered collection of homogeneous data elements. The other name of stack is Last-in -First-out list. One of the most useful concepts and frequently used data structure of variable size for problem solving is the stack.

### **2. What are the operations of Stack?**

- a. CreateStack/ InitStack(Stack) – creates an empty stack
- b. Push(Item) – pushes an item on the top of the stack
- c. Pop(Item) – removes the top most element from the stack
- d. Top(Stack) – returns the first element from the stack
- e. IsEmpty(Stack) – returns true if the stack is empty

### **3. What is a Queue ?**

A Queue is an ordered collection of items from which items may be deleted at one end called the front of the queue and into which items may be inserted at the other end called rear of the queue. Queue is called as First –in-First-Out (FIFO).

### **4. What is a Priority Queue?**

Priority queue is a data structure in which the intrinsic ordering of the elements does determine the results of its basic operations. Ascending and descending priority queue are the two types of Priority queue.

### **5. What are the postfix and prefix forms of the expression?**

$A+B*(C-D)/(P-R)$

□ Postfix form: ABCD-\*PR-/+□□

Prefix form: +A/\*B-CD-PR □

□

## 6. Explain the usage of stack in recursive algorithm implementation?

In recursive algorithms, stack data structures is used to store the return address when a recursive call is encountered and also to store the values of all the parameters essential to the current state of the procedure.

## 7. Write down the operations that can be done with queue data structure?

Queue is a first - in -first out list. The operations that can be done with queue are insert and remove.

## 8. What is a circular queue?

The queue, which wraps around upon reaching the end of the array is called as circular queue.

## 9. How do you test for an empty queue?

To test for an empty queue, we have to check whether  $READ=HEAD$  where REAR is a pointer pointing to the last node in a queue and HEAD is a pointer that pointer to the dummy header.

In the case of array implementation of queue, the condition to be checked for an empty queue is  $READ<FRONT$ .

## 10. Define recursion?

It is a technique and it can be defined as any function that calls itself is called recursion. There are some applications which are suitable for only recursion such as, tower of Hanoi, binary tree traversals etc, can be implementing very easily and efficiently.

## 11. Write the routine to push a element into a stack.

```
Push(Element X, Stack S)
{
if(IsFull(S)
{
Error("Full Stack");
}
else
S→Array[++S→TopOfStack]=X;
}
```

## 12. What are the applications of stack?

The following are the applications of stacks

- Evaluating arithmetic expressions
- Balancing the parenthesis
- Towers of Hanoi
- Function calls
- Tree traversal

## 13. How the stack is implemented by linked list?

It involves dynamically allocating memory space at run time while performing stack operations.

Since it consumes only that much amount of space is required for holding its data elements, it prevents wastage of memory space.

```

struct stack
{

int element;
struct stack *next;
}*top;

```

**14. Write the routine to pop a element from a stack.**

```

int pop()
{
if(top==NULL)
{
printf("\n Stack is empty.\n");
getch();
exit(1);
}
else
{
int temp;
temp=top->element; /* retrieving the top element*/
top=top->next; /* Updating the stack pointer */
return temp; /* returning the popped value */
}
}

```

**15. What are the operations of a queue?**

The operations of a queue are

```

isEmpty()
isFull()
insert()
delete()
display()

```

**16. How the queue is implemented by linked list?**

- It is based on the dynamic memory management techniques which allow allocation and de-allocation of memory space at runtime.

**Insert operation**

It involves the following subtasks:

- Reserving memory space of the size of a queue element in memory
- Storing the added value at the new location
- Linking the new element with existing queue
- Updating the *rear* pointer

**Delete operation**

It involves the following subtasks:

1. Checking whether queue is empty
2. Retrieving the front most element of the queue
3. Updating the front pointer
4. Returning the retrieved value

**17. What are the applications of queue?**

The following are the areas in which queues are applicable

- a. Simulation
- b. Batch processing in an operating systems
- c. Multiprogramming platform systems
- d. Queuing theory
- e. Printer server routines
- f. Scheduling algorithms like disk scheduling , CPU scheduling
- g. I/O buffer requests

### **18. What are push and pop operations?**

- Push – adding an element to the top of stack
- Pop – removing or deleting an element from the top of stack

### **19. What are enqueue and dequeue operations?**

- Enqueue - adding an element to the queue at the rear end
- Dequeue – removing or deleting an element from the queue at the front end

### **20. What are the types of queue?**

The following are the types of queue:

Double ended queue

Circular queue

Priority queue

## **PART B**

1. Explain Stack ADT and its operations
2. Explain array based implementation of stacks
3. Explain linked list implementation of stacks
4. Explain the applications of Stacks
5. Explain how to evaluate arithmetic expressions using stacks
6. Explain queue ADT
7. Explain array based implementation of queues
8. Explain linked list implementation of queues
9. Explain the applications of queues
10. Explain circular queue and its implementation
11. Explain double ended queue and its operations

## **UNIT V**

### **SORTING, SEARCHING AND HASH TECHNIQUES**

Sorting algorithms: Insertion sort - Selection sort - Shell sort - Bubble sort - Quick sort - Merge sort - Radix sort – Searching: Linear search – Binary Search Hashing: Hash Functions – Separate Chaining – Open Addressing – Rehashing – Extendible Hashing

## **PART - A**

### **1. What is meant by Sorting and searching?**

Sorting and searching are fundamentals operations in computer science. Sorting refers to the operation of arranging data in some given order. Searching refers to the operation of searching the particular record from the existing information

### **2. Define Bubble sort.**

Bubble sort is the one of the easiest sorting method. In this method each data item is compared with its neighbor and if it is an descending sorting, then the bigger number is moved to the top of all the smaller numbers are slowly moved to the bottom position; hence it is also called as the exchange sort.

### **3. Mention the various types of searching techniques in C**

- Linear Search
- Binary Search

#### **4. What is linear search?**

In Linear Search the list is searched sequentially and the position is returned if the key element to be searched is available in the list, otherwise -1 is returned. The search in Linear Search starts at the beginning of an array and move to the end, testing for a match at each item.

#### **5. What is binary search?**

Binary search is simpler and faster than linear search. Binary search the array to be searched is divided into two parts, one of which is ignored as it will not contain the required element One essential condition for the binary search is that the array which is to be searched, should be arranged in order.

#### **6. Define merge sort.**

Merge sort is based on divide and conquer method. It takes the list to be stored and divide it in half to create two unsorted lists. The two unsorted lists are then sorted and merge to get a sorted list.

#### **7. Define insertion sort.**

Successive element in the array to be sorted and inserted into its proper place with respect to the other already sorted element. We start with second element and put it in its correct place, so that the first and second elements of the array are in order.

#### **8. Define selection sort.**

It basically determines the minimum or maximum of the lists and swaps it with the element at the index where it's supposed to be. The process is repeated such that the  $n^{\text{th}}$  minimum or maximum element is swapped with the element at the  $n-1^{\text{th}}$  index of the list.

#### **9. What is the basic idea of shell sort?**

Shell sort works by comparing elements that are distant rather than adjacent elements in an array or list where adjacent elements are compared. Shell sort uses an increment sequence. The increment size is reduced after each pass until increment size is 1.

#### **10. What is the purpose of quick sort and advantage?**

The purpose of the quick sort is to move a data item in the correct direction; just enough for to reach its final, place in the array. □ Quick sort reduces unnecessary swaps and moves an item to a greater distance, in one move. □ □

#### **11. Define quick sort.**

The quicksort algorithm is fastest when the median of the array is chosen as the pivot value. That is because the resulting partitions are of very similar size. Each partition splits itself in two and thus the base case is reached very quickly and it follows the divide and conquers strategy.

#### **12. Advantage of quick sort?**

Quick sort reduces unnecessary swaps and moves an item to a greater distance, in one move.

### 13. Define radix sort

Radix sort the elements by processing its individual digits. Radix sort processing the digits either by least significant digit(LSD) method or by most significant digit(MSD) method.Radix sort is a clever and intuitive little sorting algorithm, radix sort puts the elements in order by comparing the digits of the numbers.

### 14. Define hash table

All the large collection of data are stored in a hash table. The size of the hash table is usually fixed and it is bigger than the number of elements we are going to store.The load factor defines the ration of the number of data to be stored to the size of the hash table.

### 15. What are the types of hashing?

□ □ Static hashing-In static hashing the process is carried out without the usage of an index structure. □  
Dynamic hashing- It allows dynamic allocation of buckets, i.e □.according to the demand of database the buckets can be allocated making this approach more efficient.

### 16. Define Rehashing.

Rehashing is technique also called as double hashing used in hash tables to resolve hash collisions, cases when two different values to be searched for produce the same hash key.It is a popular collision-resolution technique in open-addressed hash tables.

### 17. How the insertion sort is done with the array?

It sorts a list of elements by inserting each successive element in the previously sorted sublist.

Consider an array to be sorted  $A[1], A[2], \dots, A[n]$

- a. Pass 1 :  $A[2]$  is compared with  $A[1]$  and placed them in sorted order.
- b. Pass 2 :  $A[3]$  is compared with both  $A[1]$  and  $A[2]$  and inserted at an appropriate place. This makes  $A[1], A[2], A[3]$  as a sorted sub array.
- c. Pass  $n-1$  :  $A[n]$  is compared with each element in the sub array  $A[1], A[2], \dots, A[n-1]$  and inserted at an appropriate position.

### 18. What is open addressing?

Open addressing is also called closed hashing, which is an alternative to resolve the collisions with linked lists. In this hashing system, if a collision occurs, alternative cells are tried until an empty cell is found. There are three strategies in open addressing:

- Linear probing
- Quadratic probing
- Double hashing

### 19. What are the collision resolution methods?

The following are the collision resolution methods

- Separate chaining
- Open addressing
- Multiple hashing



## **20. Define separate chaining**

It is an open hashing technique. A pointer field is added to each record location, when an overflow occurs, this pointer is set to point to overflow blocks making a linked list. In this method, the table can never overflow, since the linked lists are only extended upon the arrival of new keys.

### **PART B**

1. Explain the sorting algorithms
2. Explain the searching algorithms
3. Explain hashing
4. Explain open addressing
5. Write a C program to sort the elements using bubble sort.
6. Write a C program to perform searching operations using linear and binary search.
7. Explain in detail about separate chaining.