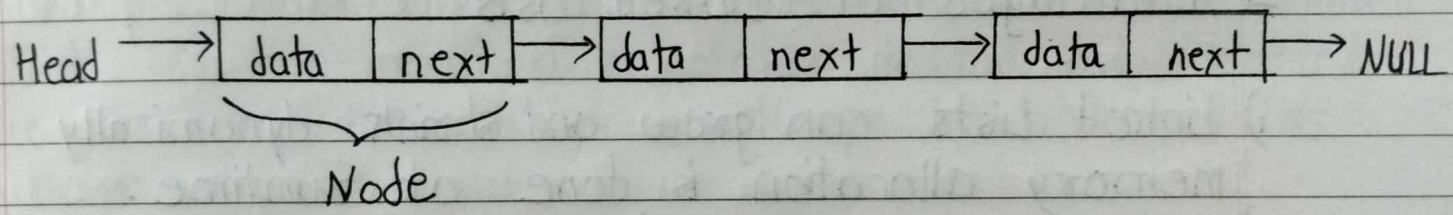
Page No.
Date

Assignment No: 3

Q.
1) What is linked list? State advantages & disadvantages of linked list?

→

Linked List is a linear data structure, in which elements are not stored at a contiguous location, rather they are linked using pointers. Linked List forms a series of connected nodes, where each node stores the data and the address of the next node.



① Node: A node in a linked list typically consists of two components.

② Data: It holds the actual value or data associated with the node.

③ Next (pointer): It stores the memory address (reference) of the next node in the sequence.

④ Head & Tail: The linked list is accessed through the head node which points to the first node. The last node (tail) in the list points to NULL.

Page No.
Date

Types of linked lists:
① Single - linked list
② Double - linked list
③ Circular linked list.

Operations on Linked Lists:
i) Insertion
ii) Deletion
iii) Searching.

Advantages of Linked Lists.

i) Linked Lists can grow and shrink dynamically, as memory allocation is done at runtime.
ii) Adding or removing elements from a linked list is efficient, especially for large lists.
iii) Can be easily reorganized and modified without requiring a contiguous block of memory.

Disadvantages

i) Unlike arrays, linked lists do not allow direct access to elements by index. Traversal is required to reach a specific node.
ii) Linked lists require additional memory for storing the pointers, compared to arrays.

2) Explain various applications of linked list.

→

Linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers.

Applications of Circular Linked List.

i) Useful for implementation of a queue. Unlike this implementation, we don't need to maintain two-pointers for the front and rear if we use a circular linked list. We can maintain a pointer to the last inserted node and the front can always be obtained as next of last.

ii) Circular linked lists can be used to implement circular queues, which are often used in operating systems for scheduling processes and managing memory allocation.

iii) Video games use circular linked lists to manage sprite animations. Each frame of the animation is represented as a node in the list and the last frame is connected to the first frame to create a loop.

iv) Traffic light control systems use circular linked lists to manage the traffic light cycles. Each phase of the traffic light cycle is represented as a node in the list and last node is connected to the first node to create a loop.

Application of Doubly Linked Lists:

i) Redo and undo functionality.
ii) Use of the Back and forward button in a browser.
iii) Used in networking.
iv) Used in Graph algorithms.

Applications of linked list in real world:

i) Image viewer: Previous and next images are linked and can be accessed by the next and previous buttons.

ii) Music Player: Songs in the music player are linked to the previous and next songs. So you can play songs either from starting or ending of the list.

iii) GPS navigation systems: Linked lists can be used to store and manage a list of locations and routes, allowing users to easily navigate to their desired destination.

iv) Image Processing: Linked lists can be used to represent images, where each pixel is represented as a node in the list.

3) What is garbage collection? Explain in detail? Who will run garbage collection program? When it will be run?

→ Garbage collection is process of reclaiming the runtime unused memory automatically. In other words, it is a way to destroy the unused objects. To do so, we use free() function

i) Garbage collection's primary goal is to reduce memory leaks. Garbage collection frees the programmer from having to deallocate and return objects to the memory system manually.
ii) Garbage collection can account for a considerable amount of a program's total processing time and as a result, can have a significant impact on performance.

Following two are commonly implemented garbage collection technique.
1. Mark and Sweep
2. Reference Counting

1. Mark and Sweep.

The mark and sweep is as straight forward as its name suggests. It consists of two phases: a mark phase and a sweep phase. The collector crawls across all the roots (local-global variables, stack frames, registers, etc) and marks every item it meets by setting a bit anywhere around that object during the mark phase. It also walks across the heap during the sweep phase, reclaiming memory from all the unmarked items.

2. Reference Counting

The method of reference counting is really easy. It is based on counting how many pointer references each allocated object has. It's straight forward, inherent incremental solution because the program's memory management overhead is distributed. Aside from memory management, reference counting is widely used in operating systems as a resource management tool for managing system resources such as files, sockets

In terms of technical specifies, reference counting based collection is the simplest garbage collection approach. If the language runtime doesn't enable pointer manipulation and/or the programmers can't manipulate the object roots, the implementation is extremely simple.

4) Differentiate between sequential & linked organization

Linked Organization	Sequential Organization
i) Data is stored in nodes that are linked together	i) Data is stored in a linear sequence.
ii) Each node contains data and a pointer to next node.	ii) Each element is stored one after the other.
iii) Allows for fast insertions and deletion	iii) Allows for fast traversal and access.

iv) More complex to implement	iv) Simpler to implement
v) Can be used for implementing data structures like trees and linked lists	v) Can be used for implementing data structures like arrays and stacks.
vi) Requires more memory for pointer	vi) Requires less memory as no pointer.
vii) Flexible in terms of size and structure.	vii) Fixed size and structure.
viii) Random access is not possible	viii) Random access is possible.
ix) Can have a circular linked list	ix) Only a linear sequential list is possible.
x) Can have multiple head & tail pointers	x) Only one head and tail pointer is required.

5) Write short note on dynamic storage management. Explain how it is done in C.

→

When a C program is compiled, the compiler allocates memory to store different data elements as constants, variables (including pointer variable) arrays and structures. This is referred as static memory allocation. But, there are several limitations for it.

i) A static array has fixed size. We cannot increase its size to handle situations requiring more elements. As a result, we will tend to declare large arrays than required, leading to wastage of memory. Also we cannot reduce array size to save memory.

The C language provides a very simple solution to overcome these limitations : dynamic memory allocation in which the memory is allocated at run-time. Dynamic memory management involves the use of pointers and four standard library functions, namely, malloc(), calloc(), realloc(), free().

i) malloc()

This method is used to dynamically allocate a single large block of memory with the specified size.

Syntax :

```
ptr = (cast-type *) malloc (byte-size)
```

For example :

```
int * ptr = (int *) malloc (5 * sizeof (int));
```

(4 bytes)

ptr = [ ] → A large 20 bytes memory block is dynamically allocated to ptr.

(20 bytes of memory)

If space is insufficient, allocation fails then returns a NULL pointer.

ii) calloc ()

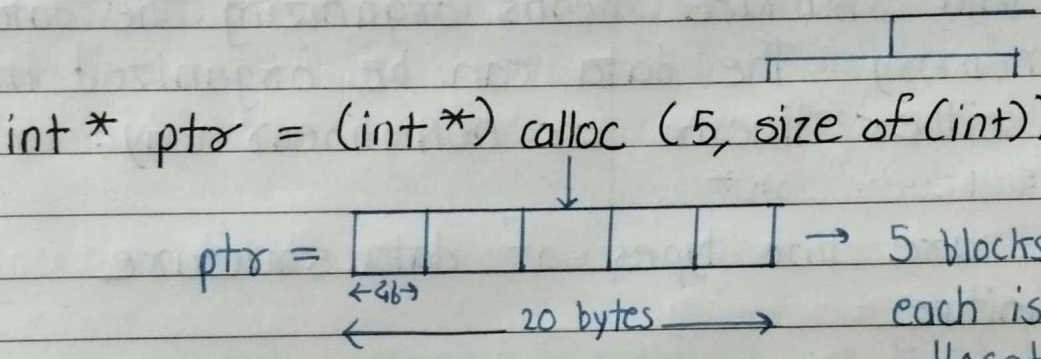
This method is used to dynamically allocate the specified no. of blocks of memory of specified type.

- It initializes each block with a default value '0'.
- It has two parameters or arguments as compare to malloc().

Syntax:

ptr = (cast-type *) calloc (n, element-size);

For example:

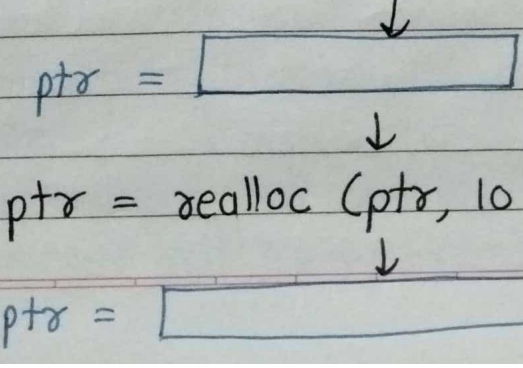


iii) Realloc ()

This method is used to dynamically change the memory allocation of a previously allocated memory.

Syntax: ptr = realloc (ptr, newSize);

For example: int* ptr = (int *) malloc (5 * sizeof (int));



iv) free()

This method in C is used to dynamically deallocate the memory.

Syntax :

```
free (ptr);
```

6) What is primitive data structure? Enlist difference between primitive data structure & non-primitive d.s.

→

Data structure means organizing the data in the memory. The data can be organized in two ways either linear or non-linear way.

There are two types of data structure.

① Primitive

② Non - Primitive.

① Primitive data structure is a fundamental type of data structure that stores the data of only one type whereas the non-primitive data structure is a type of data structure which is a user-defined that stores the data of different types in a single entity.

Primitive Data Structure	Non-primitive data structure.
① Primitive data structure is a kind of data structure that stores the data of only one type.	① Non-primitive data structure is a type of data structure that can store the data of more than one type.
② Example : int, char, float	② Example : Array, linked list
③ Contains some value, it cannot be NULL	③ Can consist of a NULL value.
④ The size depends on data type	④ The size is not fixed.
⑤ It starts with a lowercase character	⑤ It starts with uppercase character.
⑥ Can be used to call the methods.	⑥ It cannot be used to call the methods.

7) What is data structure ? Why to study data structure ? Enlist 5 areas of computer science in which data structure is used.

→

A data structure 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. It is also used for processing retrieving and storing data.

- Why ? :

Let's take a example

If you need to search your roll no in 20000 pages of PDF document (arranged in increasing order) how would you do it?

i) If you will try to search it randomly or in a sequential manner it will take too much time.

ii) Another solution for it:

① Go to page no. 10000

② If your roll no. is not there, but all other roll no. in that page are lesser than your roll no.

③ Go to page no. 15000

④ Still if your roll no. is not there. but this time all other roll no. is greater than your.

⑤ Go to page no. 12500

⑥ Continue the same process and within 30-40 seconds you will find your roll number.

iii) The above process is nothing but a binary search algorithm.

iv) Data structure helps us to save time and organize the data efficienty.

1) Web pages can be accessed using the previous and next URL links which are linked using a linked list.

2) The music players also use the same technique to switch between music.

3) Ms-Paint drawings & shapes are connected via a linked list on canvas.

4) Train coaches are connected to one another in a doubly-linked list fashion.

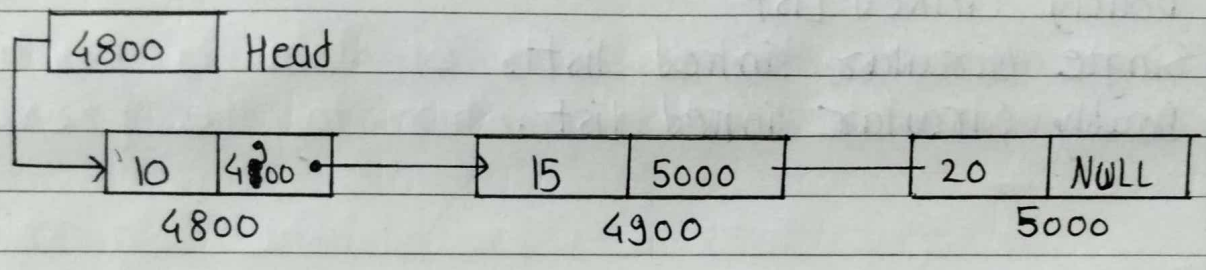
5) History of visited pages.

8) What is singly linked list ? Write algorithm to find the number of times a given ITEM occurs in the singly linked list.

→

A singly linked list is a special type of linked list in which each node has only one link that points to the next node in the linked list.

Example :



- Each node holds a single value and a reference to the next node in the list.

Algorithm :

Step 1 : Start
Step 2 : Create a function of a linked list, Pass A number as arguments and provide the count of the number to the function.
Step 3 : Initialize count = 0
Step 4 : Traverse in linked list until equal number found
Step 5 : If found then update count by 1.
Step 6 : After reaching end of linked list return count.
Step 7 : Call the function
Step 8 : Print the number of ITEM occurrences
Step 9 : Stop.

9) What are the different types of linked list? Give advantages and disadvantages of each linked list.

→

A Linked list is a linear data structure, in which the elements are not stored at contiguous memory locations.

Types of Linked List.

① Singly Linked List
② Doubly Linked List
③ Single Circular linked list.
④ Doubly Circular linked list.

① Singly Linked List

Advantages	Disadvantages.
1) Efficient insertion & deletion at begnning	1) Inefficient Random Access.
2) Simple implementation	2) Limited backward traversa
3) Space efficiency	3) Extra memory overhead.
4) Flexibility in Traversal	4) Costly deletion operations.
5) Dynamic size	5) Difficulty in finding the last element.

Page No.
Date

② Doubly Linked List

Advantages	Disadvantages.
i) Bi-directional traversal	i) Increased Memory usage.
ii) Efficient insertion and deletion at Any position	ii) Complexity in implementation
iii) Enhanced flexibility in implementations	iii) Slower Traversal
iv) Improved Error handling	iv) Increased overhead in insertion and deletion.
v) Simplified reversal of elements.	v) Reduced space efficiency.

③ Circular Linked Lists

Advantages	Disadvantages
i) Efficient operations.	i) Complexity in implementation
ii) Space efficiency.	ii) Memory leaks
iii) Flexibility.	iii) Traversal can be more difficult
iv) Dynamic size	iv) lack of a natural end.
v) Ease of implementation.	

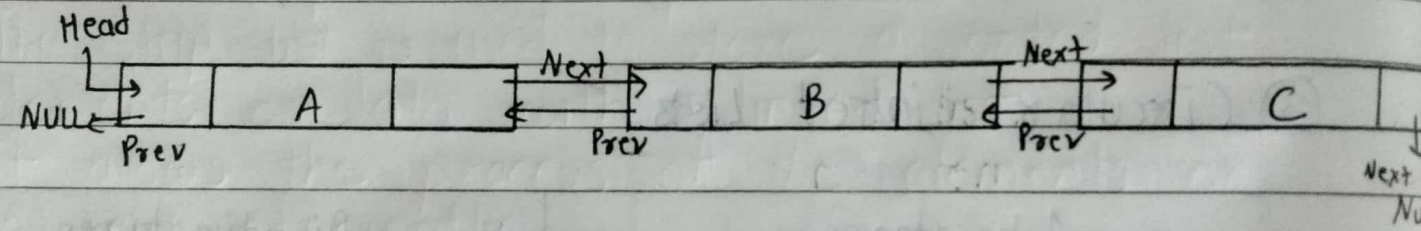
10) Explain circular and double linked list in detail.

→

① Doubly Linked List :

A doubly linked list or a two-way linked list is a more complex type of linked list that contains a pointer to the next as well as the previous node in sequence.

∴ it contains three parts of data, a pointer to the next node and a pointer to the previous node. ~~it~~ :



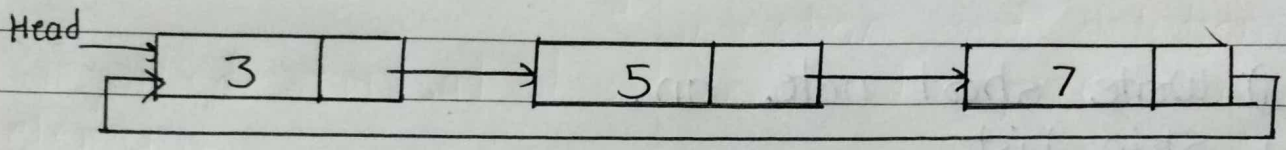
This would enable us to traverse the list in the backward direction as well.

Syntax :

```
Struct Node {
    int data;
    struct node * previous;
    struct node * next;
};
```

② Circular Linked List

While traversing a circular linked list, we can begin at any node and traverse the list in any direction forward and backward until we reach the same node we started. Thus, a circular linked list has no beginning and no end.

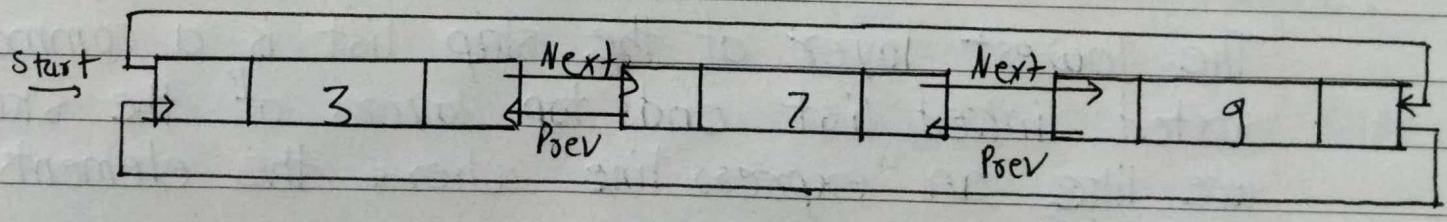


Syntax :

```
struct Node {
        int data;
        struct node * next;
};
```

- Doubly Circular linked list

The circular doubly linked list does not contain NULL in the previous field of the first node.



Page No.
Date

Syntax:

```
Struct Node *{
        int data;
        struct Node* prev;
        struct Node* next;
};
```

II) Write short note on.

1. Skip List:

A skip list is a probabilistic data structure. The skip list is used to store a sorted list of elements or data with a linked list.
It allows the process of the elements or data to view efficiently.
In one single step, it skips several elements of the entire list, which is why it is known as a skip list.

It is built in two layers: The lowest layer and Top layer.

The lowest layer of the skip list is a common sorted linked list and top layers of the skip list are like an "express line" where the elements are skipped.

i) Search complexity : O(log n)
ii) Delete complexity : O(log n)
iii) Insert complexity : O(log n)
iv) Space complexity : O(n log n) - worst case.

Advantages:
i) Simple to implement
ii) Searching and insertion is fast

Disadvantages:
i) It requires more memory
ii) Reverse searching is not allowed.

2. Header Linked list:

A header linked list START will not point to the first node of the list but START will contain the address of the header node.

It is a special type of linked list that contains a header node at the beginning of the list.

There are two types:
1) Grounded Linked List: The last node contains NULL.

2) Circular: Last node contains address of first Node.