**DAY -16**

**DATA STRUCTURES**

* Data structures are the fundamental building blocks of computer programming. They define how data is organized, stored, and manipulated within a program**.**
* **Self referential Pointer :** when a member of structure is pointer which is referencing to itself type is known as self referential pointer.
* Only for navigation we use srp
* **Example :**

Struct home

{

Int value;

Struct home \*ptr; 2000 (Address)

}HOME;

\*

V

HOME h1;

HOME h2;

HOME h3;

\*

10

H1.value=10; 🡪

H2.value=20;

H3.value=30;

10 \* \*\*8888\*\*\*\*

H1.ptr=NULL; 🡪 NULL

H2.ptr=NULL;

H3.ptr=NULL;

Pf(“%d”, h1.value) 🡺10

Pf(“%d”, ptr ->value) 🡺10

H1.ptr1 =&h1 (holds the address)

H2.ptr1 =&h2 (holds the address) Making Relationship

H3.ptr1 =&h3 (holds the address) (Created Linked List)

Ptr1= &h1 (Base Address(starting element) of list);

Pf(“%d”,ptr1 -> value) ==🡺 contents of (2000)]

Ptr1 =&h2(changing the loc from h1 to h2)

Ptr1 = h1.ptr ;

Pf(“%d”, ptr ->value) =🡺 20

Ptr1 =&h3(changing the loc from h1 to h2)

Ptr1 = h2.ptr ;

Pf(“%d”, ptr ->value) =🡺 30

* We should not do data loss.

30 NULL

20 \*

10 \*



40 NULL

30 400

10 200

30 NULL

20 300

10 200

H1 h2 h3

h4

H1 h2 h3 h4

20 300

h5

50 NULL

ii

40

* If the list is present go to the end and append.
* If the list is not present it will become the first node of the list.
* For append we used to add the base address
* If fun is declared first then the structure then the reference is lost

**DELETING THE NODE:**

n1 n2 n3 n4

10 200

20 300

30 400

40 NULL

Key =30

Head Delete =n3

Temp=head(Later on free it)

Temp.v==30;

Temp.ptr =400;

----------------------------------------------------------------------------------------------------------

Base Location = First both the temporary and main pointer are located.

Always behind the pointer the temporary pointer will be moved….

i.e header and Null Pointer

**(Finding Priyanka):**

Example : Bhanu Sree, Havilah , Sowjanya, Priyanka, Mounika

Base Loc Pooja - - - -

NULL

Gunasri Pooja - - -

* Gunasri Pooja - -
* - Gunasri Pooja -

Guna sri,Pooja

Temp, header

* Key = Priyanka;
* If the key is found then flag is 0 so it will break…

Q : While Deleting the list or header node the beginning of the list and end of the list both should consider.

Created node using dynamic memory location (i.e it is not free)

* **Single Linked List:** Traverse Only in one direction.
* **Double Linked List:**

\* 30 NULL

NULL

\* 10 \*

\* 20 \*

1

Struct Node

{

Int val;

Struct Node \*next;

Struct Node \*prev;

}NODE;

NODE n1,n2,n3;

N1.val=10;

N1.prev =NULL;

N1.next = NULL;

N2.val=10;

N2.prev =NULL;

N2.next = NULL;

N2.val=10;

N2.prev =NULL;

N2.next = NULL;

// Making RelationShip:::

N1.next =&n2;

N2.next =&n3;

N2.next = &n1;

N3.prev=&n2;

Head =&n1;

Pf(“%d”,head->val);

Head = h->n

Pf(“%d”,head->val);

Head = h->prev

Pf(“%d”,head->val);

**CIRCULAR LINKED LIST:**

\* 30 \*

\* 20 \*

\* 10 \*

**100 200 300**

nn->ptr =h

nn->n->NULL

nn->n=temp

temp->p=nn

* In circular linked list the head will be rotated