

Washington Charleson, General

A 12 STEAM WESTIMMENTS OF MACHINOPOLOS

(Approved by AICTI New Delhi & Gort, of Maharashtra, Affiliated to University of Mumbai) (Religious Jain Minority)

Semester: 3

Subject: DSAA

Academic Year: 2017-2018

UNIT : 4 LINKED LISTS

A linked list is a collection of data element called modes in which the linear representation is given by links from one mode to the next node. We have studied that an array is a linear collection of data elements in which the element in which the are stored in consecutive memory locations. while declaring arrays, we have to specify the size of array, which will restrict the number of elements that the array can store. For example, if we declare an array as int mork [10], then the array can store a maximum of 10 data elements but not more than that. But what if we are not sure of the number of elements in advance? So, there must be a data structure that removes the restrictions on the maximum number of elements and the storage condition to write efficient programs.

free from the aforementioned restrictions. A linked list does not store its elements in consecutive memory locations and the user can add any number of elements to it. However unlike an array, a linked list does not allow random access of data. Elements in a linked list can be accessed only in a



warraymaki whaterida world

A IS STEAM MARINING ON MAGINADROICH

(Approved by AICTE New Delhi & Gort, of Maharashira, Affiliated to University of Mumbai) (Resignous Zriu Minority)

Semester: 3

Subject: DSA

Academic Year: 2017-2018

sequential manner But like an array, insertions and deletions can be done at any point in the list In a constant time. Basic Concept of Linked List

A linked list, in simple terms, is a linear collection of data elements. These data elements are called modes. Linked List is a data structure which in turn can be used to implement other data structures. Thus it acts as a building block to implement data structure such as maked the perceived as a train or a sequence of modes in which each node contains one as more data fields and a pointer to the next node.

START

WIT + 2 7 3 1 + 4 1 + 5 1 + 6 1 + 7 = 1

Simple linked list

we can see a simple linked list in which every node contains data two parts, an integer and a pointer to the next node. The left part of the node which contains data may include a simple data type ion array or a structure. The right part of the node contains a pointer to the next mode Con address of the next node in sequence). The last node will have no next mode connected to it so it will store a special value called NULL.

of the list.

SERVICE CONTRACTOR CONTRACTOR

A 12 STEATH WASHINGTON ON THE STEAM OF THE (Approved by AICTI New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)

(Religious Jam Minority)

Semester: 3

Subject: DSA

Academic Year: 2017-2018

Linked lists contain a pointer voriable START that stores the address of the first mode in the list. We can tranverse the entire list using START which contains the address of the first node; the next part of the first node in turn stores the address of the first node; the next port of the first node in turn stores the address of its succeeding mode Using this technique the individual node of the list will form a chain of nodes. If START = NULL , then the linked list is empty and contains no nodes. In C , we can implement a linked list using

the following code: struct node

int data; struct node + next;

Linked Lists verses Arrays Both arrays and linked lists are a linear collection of data elements. But unlike an array, a linked list deeb not store its nodes in consecutive memory location. Another point of difference between an array and a linked tink list is that a linked list does not allow random access of data. Modes in a linked list can be accessed only in a sequential manner. But like an array insertions and deletions can be done at any point in the list in a constant time.

PI

Contaminate Sensitable Gentle

AN IN STREAM IN STREET OF MANAGEMENT OF THEORY OF MANAGEMENT OF MANAGEME

(Approved by AICTI New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbal) (Religious Jain Minurity)

Semester: 3

Subject: DSA

Academic Year: 2017-2018

Another advantage of linked list over an array is that we can add any number of elements in the list. This is not possible in case of an array.

Thus linked lists provide an efficient way of Storing related data and performing basic Operations such as insertion, deletion and updation of information at the cost of extra space required for storing the address of next nodes.

Memory Allocation and De allocation for a Linked List

We have seen how a linked list is represented in the memory. If we want to add a node to an already existing linked list in the memory, we first find free space in the memory and then use it to store the information for example, consider the linked list shown in following figure. The linked list contains the roll number of students, marks obtained by them in Biology and finally a next field which stores the address of the next mode in sequence Now, if a new student joins the class and is asked to appear for the same test that the other students had taken, then the new stu--dent's marks should also be recorded in the linked list. For this purpose, we find a free space and store the information there. In tollowing fig. the grey shaded portion shows tree space, and thus we have 4 memory





Such course directed to the life

TO BELLY INSTITUTION OF THEIR MORNING

(Approved by AICII New Delhi & Gott, of Maharashira, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: 3

Subject: DSA

Academic Year: 2017-2018

them to store our data. This is illustrated in following fig. We can use any one of them to store out data.

Mow, the question is which part of the memory is available and which part is occupied? When we delete a node from a linked list, then who changes the status of the memory occupied by it from occupied to available. The ourswer is the operating system

	ROII MO	MONK	Next	建筑 种种 管	Roll No	Marks	Next
Las	501	38	2	BIOLOGY	Sel	78	2
T. Cal	502	84	3	2	502	84	3
3	503	45	5	3	503	us	6
14		- 20		L	512	75	-1
5	504	98	7		804	98	7
6	S	-	liber.	6	H. Tarre	1	
7	505	55	8	1	\$ 05	55	8
8	206	34	10	0	506	34	10
9				q		ASSESSED NO.	
10	507	90	11	16	507	90	11
1.5	808	87	12	13	202	8-7	12
12	509	86	13	12	509	86	13
13	510	67	15	13	810	67	15
14		- XX-1-XX-1		14		-	
15	113	56	-1	12,	5 11	56	4

(a) student's linked list and (b) linked list after the insertion of new students record.





Parlayments Constitute General

A STEAM MASHINGING ON AUGUMATOR

(Approved by AICTI New Bellis & Govt. of Maharashtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: 3

Subject: DSA

Academic Year: 2017-2018

When we delete a particular mode from an existing linked list or delete the entire linked list, the space occupied by it must be given back to the free pool so that the memory can be reused by some other program that needs memory space.

The operating system does this task of adding freed memory to the free pool. The operating system will perform this operation whenever it finds the CPU idle or whenever the programs are falling short of memory space. The operating system scans through all the memory cells and marks those cells that are being used by Some program. These process is called goodage collection.

SINGLY LINKED LISTS

A singly linked list is the simplest type of linked list in which every node contains some data and a pointer to the next node of the same data type. By saying that the node contains a pointer to the next node, we mean that the node stores the address of the next node sequence. A singly linked list allows traversal of data only in one way ISTARTI

Singly Linked list





Proparation Charlette Confid

TO STRAIL MASIMUMOND OF MEDIMORDEN

(Approved by AICTI New Belli: & Govt. of Maharashtra, Affiliated to University of Mumba))
(Religious Jain Minority)

Semester: 3

Subject: DSA

Academic Year: 2017 - 2018

Traversing a Linked List

Traversing a linked list means accessing
the nodes of the list in order to perform
some processing on them. Remember a
linked list always contains a pointer variable
START which stores the address of the first
node of the list. End of the list is marked
by storeing NULL or -1 in the NEXT field of the
last node. For traversing the linked list, we
also make use of another pointer variable
PTR which points to the node that is
Currently being accessed. The algorithm to
traverse a linked list is shown in fig.

Step 1: [INITIALIZE] SET PTR = START Step 2: Repeal Steps 3 and 4 while PTR!

SHEP 3: SET PTR = PTR - NEXT

Step 4

Step 3: Apply Process to PTR -> DATA

Step 4: SET PTR = PTR - NEXT

[GOOT SO GNED

Step 5 : EXIT

searching for a Value in a Linked List searching a linked list means to find a porticular element in the linked list. As already discussed, a linked list contain consists of nodes which are divided into two pouts, the information port and the next pout. So searching means finding



A 15 3:17/1: INEQUALITY OF A HEROTON INVESTOR (Approved by AICTL New Delhi & Govt. of Maltarashtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

whether a given value is present in the information pant and the next port 80 searching means finding whether of the mode or not. If it is present, the algorithm returns the address of the node that contains the value The tollowing fig shows the algorithm to search a linked list

Step 1 : [INITIALIZE] SET PTR = START

Step 2 : Repeat Step 3 while PTR! = NULL

Step 3 : IF VAL = PTR -> DATA

SET POS = PTR

Go to step 5

ELSE

SET PTR = PTR -> NEXT

[END OF IF]

[END OF LOOP]

Step 4 : SET POS = MULL

Step 5 : EXIT

Inserting a New Mode 9n a Linked List In this section, we will see how a new node is added into an already existing linked list. We will take four cases and then see how insertion is done in each case.

Case 1: The new node is inserted at the beginning

Case 1: The new mode is inserted at the end

case 3: The new node is inserted after a given node

Case 4: The new node is inserted before a given node.





SALIMATER CHARLEST COLLEGE

The Birth Manning or hadingered.

(Approved by AICTI New Delhi & Govt. of Maharashtra, Affiliated in University of Mumbal) (Religious Jain Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

Inserting a Node at the Beginning of a linked List

Consider the linked list shown as below suppose we want to add a new node with data 9 and add it as the first node of the list. Then the following changes will be done in the linked list.

17 - 17 - 3 - 4 - 2 - 46 - 5=

START

Allocate memory for the new node and initialize its DATA port to 9

9

Add the new mode as the first mode of the list by making the next port of the new mode contain the address of START

9+1-47-3+41+27-461+51=

Now make START to point to the first mode of the list

9 + 11 + 13 + 4 + 2 + 16 + 15 =

at the beginning of a linked



PERSONAL CONTROLS CONTROL

A 15 STEVEL BY SHUNDHUNDS OF THE STEVENS

(Approved by AICT) New Delti & Govt. of Maharashtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

Step 1: IF AVAIL = MULL Write OVERFLOW Go to Step 7

[END OF IF]

Step 2 : SET MEW_MODE = AVAIL

Step 3: SET NEW-MODE = AVAIL - NEYT

Step 4: SET NEW_NODE - DATA = VAL

STEP 5 : SET NEW_MODE + NEXT = START

SHEP 6 : SET START = NEW_NODE

TIXE : Fasts

is available for the new node. If the free memory has exhausted, then an overflow message is printed.

Inserting a Mode at the End of Linked List

Step 1 : IF AVAIL & NULL

Write OVERFLOW

goto to step 10

[FI TO ONE]

Step 2 : SET NEW-MODE = AVAIL

Step 3 : SET AVAIL = AVAIL - NEXT

SHO 4 : SET NEW HODE - DATA = VAL

SHOS : SET NEW-HODE + NEXT = NULL

Step6: SET PTR = START

Step 7: Repeat Step & while PTR > NEXT! = NULL

SHEP 8 ? SET PTR = PTR - NEXT

[END OF LODD]

STEPP : SET PTR & HEXT = NEW_NODE

steplo: EXIT





Francousti Charleally Control

A 12 STENT BY CO CHUNHUMINES OF THE CHANGE OF

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

Inserting a Mode After a Given Mode in a Linked List

> Step 1 : IF AVAIL . NULL write OVERFLOW Go to Step 12 [END OF IF]

Step 2 : SET MEW_MODE = AVAIL

GIEPS : SET AVAIL : AVAL - NEXT

SHEP 4 " SET MEW_NODE -> DATA = VAL

SHEPS : SET PTR = START

STEP 6 & SET PREPTR - PTR

Step 7 : Repeat Step 8 and 9 while PREPIR-

Data -

MUM = 1

Step & ! SET PREPTR = PTR

Step 9 : SET PTR = PTR > NEXT

LEND OF LOOP]

STEP 10 : PREPTR - NEXT = NEW MODE

Step 11 : SET NEW_MODE > NEXT = PTR

Step 12 % EXIT





Sariasanaki Sinadalili Sanasi.

The attraction of about the property

(Approved by AICTI New Delhi & Govt. of Maharashtra, Affiliated in University of Mambal) (Religions Jain Ministry)

Semester: 3

Subject: DSA

Academic Year: 27-(8

Inserting a Mode Before a Given Mode in Linked List

SIEP 1: IF AVMIL = MULL Write OVERFLOW GOTO STEP 12 [END OF IF]

Step 2 : SET NEW_NODE = AVAIL

STEP 3 : SET AVAIL - AVAIL - NEXT

Step 4 : SET NEW_NODE - DATA = VAL

SKO 5 " SET PTR = START

SKP 6 : SET PREPTR = PTR

stept : Repeat step 8 and 9 while

PTR - DATA ! = MUM

Step8 \$ SET PREPTR = PTR

Step9: BET PTR = PTR > NEXT

LEND OF LOOP]

Step10: PREPTR - NEXT - NEW_MODE

STOPII: SET HEN-HODE - NEXT = PTR

Step 12: EXIT

[Algorithm to insert a new mode before a mode that has NUM.]





TO STREET BY SHUNDING ON THEOTHERY

(Approved by AICTI New Delhi & Govt. of Maharashura, Affiliated to University of Mumbai) (Religious Juio Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

Deleting a Mode from a Linked List

In this section, we will discuss how a mode is deleted from an already existing linked list. We will consider three cases and then see how deletion is done in each case

Case 1: The first mode is deleted Case 2: The last mode is deleted case 3: The node after a given mode is deleted

Deleting the first Mode from a linked List When we want to delete a node from the beginning of the list, then the following changes will be done in the linked list.

+> 17 +> 13 +> 4 +> 2 +> 6 + START

Make START to point to the next mode in sequence

71 - + 31 + 101 + 121 + 161 + 151x1 START

Deleting the first node of alinked list

Step 1 : IF START = NULL Write UNDERFLOW

Step : SET PTR = START

Step3 : SET START = START - NEXT

STEPY : FREETR

Steps: EXIT



PRINTING CHARLES COLD

A B SIEVI MENUMBER OF MEDITADROS

(Approved by AICTE New Delhi & Gost, of Maharashtra, Affiliated to University of Munibal) (Religious Jain Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

Deleting the last Mode from a Linked List Suppose we want to delete the last node from Linked list , then the following charges willbe done in the linked list . Take a pointer variables PTR & PREPTR which initially Point to START START PREPTE PTE) Move PTR and PREPTR such that NEXT Past Of PTR = MULL . PREPTR always points to the node just before the node pointed by PTR 117-171-START PREPTR Set the MEXT point of PREPTR node to MULL START Step1 : IF START = NULL Write UNDERFLOW [END OF IF] SHOP 2 : SET PTR = START Step3 : Repeat Step 4 & 5 while PTR + MEXT = NOW STEP 4: SET PREPTR = PTR SHEP S! SET PTR = PTR - NEXT EGOOD TO GUE Y SET PREPTR - NEXT - NULL 6 tep 6: SHEPT : FREE PTR Step 8 : EXIT





A IS SHEAR MARAMANANA OR MEDITALISMA

(Approved by AICIE New Delhi & Gevt. of Maharashtra, Affiliated to University of Muzabat) (Religious Jain Minority)

Semester: 3 Subject: DSA Academic Year: 17-18-

Deleting the Mode After a given Mode in a Linked List suppose we would to delete the rode that succeeds the node which contains data value 4 then the following changes will be done in the linked list. START Take pointer variable PTR and PREPTR which initially point to start 72 741-PREPTE Move PREFTR and PTR such that PREPTR points to the node containing VAL and PTR points to the succeeding START PREPTR PTR PTR PREPTR START PREPTR START SET THE NEXT port of PREPTR to the NEXT port of PTR 141 PTR START PREPTR START Deleting the node after a given in linked list node





STREET, STREET, STREET, STREET,

7 13 23:17:11 11/12 HARDING OF THE CHANGE OF THE COLORS

(Approved by AICTI New Delhi & Govt. of Malarashtra, Affiliated to University of Mumbai) (Religious Janu Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

Step 1: IF START = NULL
WRITE UNDERROW
Go to Step 10
[END OF IF]

Step 2 : SET PTR - START

Step 3 : SET PRPTR = PTR

Step 4: Repeat Step 5 and 6 while PREPTR -> DATA! = NUM

Step 5 : SET PREATR = PTR

STEP 6 1 SET PTR = PTR -> NEXT

[FND OF LOOP]

Step 7 : SET TEMP = PTR

SIEP 8 : SET PREPTR - NEXT - PTR - NEXT

step 9 : FREE TEMP

Step 10 : EXIT

Alogorithm to delete the node after a given node



(Approved by AICTE New Delhi & Govt. of Maharushtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: Subject: Academic Year: ____

//Creating a Singly Linked List inserting a node at the beginning, At end and at specific position. Also deleting a node from beginning, from end and from specific position.

```
#include<stdio.h>
#include<stdlib.h>
// A linked list (LL) node to store a data element
struct node
{
int data;
struct node* next;
}*start;
//Insert data at the beginning
void insert at beg(int x)
 /*Allocate Node or allocate memory for new node*/
 struct node* newnode=(struct node*)malloc(sizeof(struct node));
 printf("\nEnter the data :");
 scanf("%d",&x);
 if(start==NULL)//if Linked List is empty
 /*Put in the data in the node and set next to NULL*/
  newnode->data=x:
  newnode->next=NULL;
 /*Move start to point to the newnode*/
  start=newnode;
 else
 /*Put in the data in the node*/
 newnode->data=x;
 /*Make next of new node as start*/
 newnode->next=start;
 /*Move the start to point to the new node */
 start=newnode;
//Insert data at the end
void insert at end(int x)
 /*create a pointer to traverse till end of linked list*/
 struct node* ptr;
 /*Allocate Node or allocate memory for new node*/
 struct node* newnode=(struct node*)malloc(sizeof(struct node));
 printf("\nEnter the data :");
 scanf("%d",&x);
 /*Put in the data in the node and set next to NULL*/
 newnode->data=x:
 newnode->next=NULL;
 if(start==NULL)//if Linked List is empty Make next of new node as start
```



(Approved by AICTE New Delhi & Govt. of Maharushtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: Subject: Academic Year:

```
{ //newnode->data=x;
   //newnode->next=NULL;
   /*Move the start to point to the new node*/
   start=newnode;
 else //if not empty then
 ptr=start;
 while(ptr->next!=NULL)//traverse through Linked List till last node
   ptr=ptr->next;
 ptr->next=newnode;//linked last node of next to newnode
//Insert data at specific position
void insert_at_specific_position(int x,int pos)
 struct node* ptr;
 ptr=start;
 int i=0;
 struct node*newnode=(struct node*)malloc(sizeof(struct node));
 printf("\nEnter the data :");
 scanf("%d",&x);
 printf("Enter the position :");
 scanf("%d",&pos);
 if(pos==1)
   printf("Please use insert_at_begin() ");
   //break;
 else //if posotion is not 1 then
    newnode->data=x;
    if(start==NULL)//if LL is empty
     start=newnode;
     newnode->next=NULL;
    else
      while(i<(pos-2))//if pos=5 while loop will continue from 0 till i<3
      ptr=ptr->next;//in last iteration ptr=3->next i.e 4
      i++;
      newnode->next=ptr->next;//ptr->next is 4->next i.e 5 that is 5th node shifted to
                                                                                          6th position
      ptr->next=newnode;//4->next is set to newnode i.e newnode is added at 5th position
 }
//Delete data from beginning
void delete_from_beg()
{
```



(Approved by AICTE New Delhi & Covt. of Maharushtra, Affiliated to University of Mumbol) (Religious Jain Minority)

Semester: Subject: Academic Year:

```
struct node* ptr;
 if(start==NULL)
 printf("\nUNDERFLOW!!!");
 //break;
 else
 ptr=start;
 start=start->next;
 printf("\n%d is deleted...",ptr->data);
 free(ptr);
//Delete data from end
void delete_from_end()
{
 struct node* ptr,*preptr;
 if(start==NULL)
 printf("\nUNDERFLOW!!!");
 //break:
 else
 ptr=start;
 if(ptr->next==NULL)//if there is only one element in the LL
  printf("\n%d is deleted...",ptr->data);
  start=NULL;
  free(ptr);
  }
 else
  while(ptr->next!=NULL)
   preptr=ptr;//preptr=1
  ptr=ptr->next;//ptr=ptr->next=1->next=2
  preptr->next=NULL;
  printf("\n%d is deleted...",ptr->data);
  free(ptr);
  }
 }
//Delete data from specific position
void delete_from_specific_position(int pos)
 struct node* ptr,* temp;
 int i=0;
 if(start==NULL)
 printf("\nUNDERFLOW!!!");
```



(Approved by AICTE New Delhi & Govt. of Maharushtra. Affiliated to University of Mumbol) (Religious Jain Minority)

Semester: Subject: Academic Year: ____

```
//break;
 }
 else
 printf("\nEnter the position :");
 scanf("%d",&pos);
 ptr=start;
 if(ptr->next==NULL)//if there is only one node in the LL
  printf("\n%d is deleted...",ptr->data);
  start=NULL;
  free(ptr);
 else
  while(i<(pos-1))//if pos=4 then while loop will continue from 0 to 3
  temp=ptr;//in last iteration temp=4
  ptr=ptr->next;//ptr=4->next=5
  i++;
  temp->next=ptr->next;//4->next=5->next i.e. 4->next=6
  printf("\n%d is deleted...",ptr->data);
  free(ptr);
 }
void display()
 struct node *ptr=start;
 if(ptr==NULL)
 printf("\nSORRY! NO ELEMENT!!!");
 else
 while(ptr!=NULL)
  printf("%d -> ",ptr->data);
  ptr=ptr->next;
 }
void main()
start=NULL;
//head;
int ch,x,pos;
int z=1;
while(z) //While loop to keep program in loop
printf("\n\n----\n");
printf("1. Insert at the begining\n");
```



(Approved by AICTE New Delhi & Govt. of Maharushtra. Affiliated to University of Mumbol) (Religious Jain Minority)

Semester: Subject: Academic Year: ____

```
printf("2. Insert at the end\n");
printf("3. Insert at a specific position\n");
printf("4. Delete from begining\n");
printf("5. Delete from the end\n");
printf("6. Delete from a specific position\n");
printf("7. Display\n");
printf("8. Exit");
printf("\n----\n");
printf("ENTER A CHOICE :");
scanf("%d",&ch);
switch(ch)
 case 1: insert_at_beg(x);
         break:
 case 2: insert_at_end(x);
         break;
 case 3: insert_at_specific_position(x,pos);
         break;
 case 4: delete_from_beg();
         break;
 case 5: delete_from_end();
         break;
 case 6: delete_from_specific_position(pos);
         break;
 case 7: display();
         break;
 case 8: exit(0);
         break;
 default : printf("\nOOPS ! WRONG CHOICE !");
          break;
//goto head;
printf(" \n Do you want to cotin....? press 1 or 0 ");
scanf("%d",&x);
}//end of while loop
}//end of main()
```



(Approved by AICTE New Delhi & Govt. of Maharushtra. Affiliated to University of Mumbai) (Religious Jain Minority)

Semester: Subject: Academic Year: ____

//Perform Operations on Singly Linked List like copy,concat,count no of nodes, split and reverse.

```
#include<stdio.h>
#include<stdlib.h>
struct node
int data;
struct node *next;
}*start;
void copy()
struct node *ptr=start,*t=NULL;
if(start->next==NULL)//only one element in LL
 struct node* temp=(struct node*)malloc(sizeof(struct node));
 temp->data=start->data;
 temp->next=NULL;
 t=temp;
}
else//if LL has more elements
 while(ptr!=NULL)
 struct node* temp=(struct node*)malloc(sizeof(struct node));
 temp->data=ptr->data;
 temp->next=NULL;
 if(t==NULL)//assign t to start of copied LL
  t=temp;//if copied LL is empty then assign t to temp
 else// if t is not empty then traverse till the end
  struct node *add=t;//pointer to link list "t"
  while(add->next!=NULL)
   add=add->next;
  add->next=temp;
 ptr=ptr->next;//to copy next node from original LL to copied LL
printf("\n->Copied list:\n");
struct node *k=t;//pointer to link list "t"
while(k!=NULL)//display copied LL using while loop
  printf("%d ",k->data);
  k=k->next;
void count()
struct node *ptr=start;
int k=0;
```



(Approved by AICTE New Delhi & Govt. of Maharushtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: Subject: Academic Year:

```
while(ptr->next!=NULL)
 k++;
 ptr=ptr->next;
k++;//as intially k=0 so incermented by 1 to count the last node
printf("\n->Length of the list : %d",k);
void concatenate()
struct node *head=NULL;
int m,j,x;
printf("Create list2:-\n");
printf("\nEnter the no. of nodes : ");
scanf("%d",&m);
for(j=0;j < m;j++)
{
 struct node* temp=(struct node*)malloc(sizeof(struct node));
 printf("\nEnter the data %d : ",j+1);
 scanf("%d",&x);
 temp->data=x;
 temp->next=NULL;
 if(head==NULL)//for first node of empty LL
 head=temp;
 else //for next nodes
 struct node *add=head;
 while(add->next!=NULL)//traverse till the last node
   add=add->next;
 add->next=temp;
}//linked list2 created
struct node *ptr1=start,*ptr2=head;
while(ptr1->next!=NULL)//traverse till last node of LL1
 ptr1=ptr1->next;
}
ptr1->next=ptr2;//set last node ptr of LL1 to first node of LL2
printf("\n->Concatenated List :-\n");
struct node *k=start;
while(k!=NULL)
  printf("%d ",k->data);
  k=k->next:
void split()
```



(Approved by AICTE New Delhi & Govt. of Maharushtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: Subject: Academic Year: ____

```
struct node *st,*ptr1,*ptr=start,*head=NULL,*k=start;
printf("\nEnter the position of split : ");
scanf("%d",&pos);
while(ptr->next!=NULL)
 ptr=ptr->next;
 p++;
 if(p==pos)//if pos=3 when we reach to third node we store its next ptr in "st"
          // and set 3->next to NULL
 st=ptr->next;
 ptr->next=NULL;
while(st!=NULL)
 if(head==NULL)//assign head as starting point of LL2
 struct node* temp=(struct node*)malloc(sizeof(struct node));
 temp->data=st->data;
 temp->next=NULL;
 head=temp;
 st=st->next;
 else //if head is not null then
 struct node* temp=(struct node*)malloc(sizeof(struct node));
 temp->data=st->data;
 temp->next=NULL;
 ptr1=head;
 while(ptr1->next!=NULL)//travese till the end of new LL
  ptr1=ptr1->next;
 ptr1->next=temp;
 st=st->next;
 }
printf("\nList 1:-\n");
while(k!=NULL)
 printf("%d ",k->data);
 k=k->next;
struct node *j=head;
printf("\nList 2:-\n");
while(j!=NULL)
 printf("%d ",j->data);
 j=j->next;
void reverse()
```



(Approved by AICTE New Delhi & Govt. of Maharushtra, Affiliated to University of Mumbul) (Religious Jain Minority)

Semester: Subject: Academic Year: ____

```
struct node *head=NULL,*ptr=start;
struct node* temp=(struct node*)malloc(sizeof(struct node));
while(ptr!=NULL)
 if(head==NULL)//if LL is empty
 temp->data=ptr->data;
 temp->next=NULL;
 head=temp;
 else //if LL has more nodes
 struct node* temp=(struct node*)malloc(sizeof(struct node));
 temp->data=ptr->data;
 temp->next=head;
 head=temp;
 ptr=ptr->next;
struct node *k=head;
while(k!=NULL)
 printf("%d ",k->data);
 k=k->next;
}
void main()
start=NULL;
int ch,n,x,i;
printf("Create list1:-\n");
printf("\nEnter the no. of nodes : ");
scanf("%d",&n);
for(i=0;i < n;i++)
 struct node* newnode=(struct node*)malloc(sizeof(struct node));
 printf("\nEnter the data %d: ",i+1);
 scanf("%d",&x);
 newnode->data=x;
 newnode->next=NULL;
 if(start==NULL)//if the LL is empty
 start=newnode;
 }
 else
 struct node *add=start;
 while(add->next!=NULL)//traverse till the end of LL
  add=add->next;
 add->next=newnode;
```



Semester:

(Paratazanesta Girantelata Generala STINGTON OF THE SHAHAMANA OF THE STINGTON OF THE AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai) (Religious Jain Minority)

Academic Year: _

Subject: _ printf("\nLIST CREATED SUCCESFULLY !\n"); struct node *k=start; while(k!=NULL)//display elements from LL printf("%d ",k->data); k=k->next; } do printf("\n\n-----\n"); printf("1. COPY\n2. CONCATENATE\n3. SPLIT\n4. COUNT\n5. REVERSE\n6. EXIT"); printf("\n----\n"); printf("ENTER YOUR CHOICE : "); scanf("%d",&ch); switch (ch) { case 1: copy(); break; case 2: concatenate(); break; case 3: split(); break; case 4: count(); break; case 5: reverse(); break; case 6: exit(0); break; default : printf("OOPS ! WRONG CHOICE !"); break; }while(ch!=6);





Suchaman Charlette College

A STEAT OR CHURCHEN STATE OF THEORY

(Approved by AICTI New Bellii & Govt, of Maliarashtra, Affiliated to University of Mambai) (Religious Jaio Minority)

Semester: 3

Subject: PSA

Academic Year: 17-18

Circular Linked Lists

In a circular linked list, the last mode contains a pointer to the first node of the list. We can have a circular singly linked list as well as a circular doubly linked list. While traversing a circular linked list, we can begin at any mode and traverse the list in any direction forward or backward untill we reach the the same node where we started. Thus a circular linked list has no beginning and no ending. Fig shown below

>[1] + 2] - + 3] - + 5] - 6[- >[-] - 6[- >[-] -]

Inserting a New Mode in a Circular Linked list In this section, we will see how a new node is added into an already existing linked list. We will take two cases and then see how insertion is done In each case.

Case 1: The new mode is inserted at the beginning of the circular linked list case 2: The new mode is inserted at the end of the circular linked list.



Parlixandi Bireklik Birikli



17 15 2:177:1 1772 miniming of maginal morotole,

(Approved by AICTI New Delhi & Govt. of Maharashtra, Affiliated to University of Murchal) (Religious Jain Minority)

Academic Year: 17-18 Subject: DSA Semester: 3 Inserting a Node at the Beginning of a Circular Linked List. Consider the linked list shown in following tig. Suppose we want to add a new mode with data 9 as the first mode of list. Then the following changing will be in the linked list. 11 +> 17 +> 3 +> 4 +> 12 STARTE Allocate memory for the new node and initialize its DATA part to 9 0 Take a pointer variable PTR that points to the START mode of the list. START A PTR Move PTR so that it now points to the last node of the list. START Add the new mode in between PTR & START 9 +>11->7+ +33 ->41 ->21 ->61 ->45 BTART Make START point to the new mode START Inserting a new node at the beginning

of circular linked list.





A 13 STEAT WASHINGING OF ASSISTANCE A

(Approved by AICTI New Delhi & Gort, of Maharashtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: ろ

Subject: DSA

Academic Year: 17-18

Step 1: IF AVAIL = NULL Write OVERFLOW Go to Step 11

[FND OF IE]

Step 2: SET NEW_NODE = AVAIL

Step 3: SET AVAIL = AVAIL - NEXT

Step 4 : SET NEW_NODE - DATA = YAL

Step 5 " SET PTR = START

Step 6: Repeat Step 7 while PTR >

NEXT ! = NOM START

SKP7: PTR = PTR -> NEXT

[END OF LOOP]

Step & SET NEW NODE -> NEXT = START

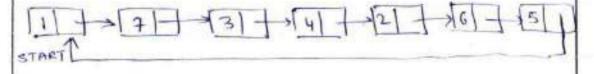
Step 9 : SET PTR - NEXT = NEXT_MODE

Step 10: SET START = NEW_NODE

Step 11 : EXIT

Algorithm to insert a new mode at the beginning

Inserting a Mode at the END of a orrewor Linked List suppose we want to add a new mode with data 9 as the last node of the list. Then the following changes will be done in the linked list





STREET, STREET, STREET, STREET,

A 13 STEATH WEARINGHING OF HERITAGEN

(Approved by AICTI New Delhi & Govt. of Maharashtra, Affiliated to University of Mumba) (Religious Join Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

Allocate memory for the new mode and initialize its DATA part to 9 . Take a pointer variable PTR which will initially point to START 11-+>17/+>13/-START PTR Move PTR so that it now points to the least node of the list. 11 +> 13 +> 131 +> 141 +> 12' START Add the new node after the node pointed by PTR. START IF AVAIL = MULL Step 1 Write OVERFLOW GO to Step 10 [END OF IF] Step 2 : SET MEW_MODE = AVAIL STOB : SET AYAIL = AYAIL - NEXT Step 4: SET NEW_MODE > DATA = VAL STEP 5 & GET NEW_HODE-> NEXT = START STEP 6 & SET PTR = START Step 7: Repeat Stop & While PTR > NEX)= STEP & SET PTR = PTR + NEXT CEND OF LOOP] Step 9 : SET PTR + NEXT = NEW_MODE Step 10 : EXIT



war paymath ethadis 321s thought

A STATE OF THE STA (Approxed by AICTI New Dolhi & Govt. of Maharashtra, Affiliated to University of Mumbai)

(Religious Jam Minority)

Semester: 3

Subject: DSA

Academic Year: 17-18

Deleting a Mode from a Circular Linked List

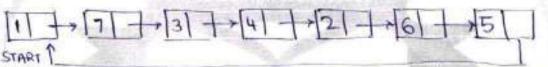
In this section, we will discuss how a node is deleted from an already existing circular linked list. We will take two cases and then see how deletion is done in each case Rest of the cases of deletion are some as that given for singly linked list.

case 1: The first node is deleted

Case 2: The last node 9s deleted

Deleting the First Mode from a Circular Linked List

Consider the circular linked list shown in following tig. When we want to delete a node from the beginning of the list, then the following changes will be done in the linked list



Take a variable PTR and make it point to START

node of the list 11-17-13 START PER

Move PTR further so that it now points to the last

START

The NEXT part of PTR is made to point to the second node of the list and the memory of



Parliamento Cinvisado Veneta

A 15 STEAT INSTRUMENTS OF THEORY

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbal) (Religious Jain Minority)

Semester: 3

Subject:

AZG

Academic Year: 17-18

the first node is freed. The second node becomes the first mode of the list.

START 1

[3]

[4]

2

[6]

5 PTR

Step 1: IF START = NULL

Write UNDERFLOW

Go to Step 8

TEND OF IF]

SKP 2 8 SET PTR = START

Step 3: Repeat Step 4 while PTR > NEXT

Step 4 : SET PTR = PTR - NEXT

Step 5 : GET PTR - NEXT - START - NEXT

SHOP & FREE START

Step 7 : SET START = PTR - NEXT

Step 8 " EXIT

Algorithm to delete the first mode

Deleting the Last Mode from a Circular Linked List

Consider the circular linked list shown in following fig. Suppose we want to delete the last node from the linked list, then the following changes will be done in the linked list.

11 +> + +> (3) - + > (-1 - +) -))))))))))))))))))))))

Serriazantele Christeleldio Centico

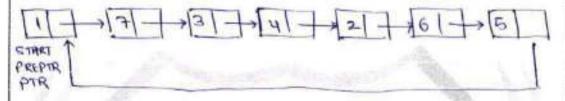
A B SHEAN MARANDAMAN ON MESCHAOLORORS

(Approved by AICTI New Delhi & Govt, of Maharashura, Affiliated to University of Mumbai) (Religious Jain Minority)

Semester: 3

Subject: DSA Academic Year: 17-18

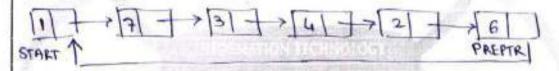
Take two pointer PREPTR and PTR which will initially point to START



Move PTR so that it points to the last node of the list PREPTR will always point to the node preceding PTR.

PREPTR START 1

Make the PREPTR's next part store START mode's address and free the space allocated for PTR Now PREPTR is last node of the list



exce 1 : IF START = NULL

Write UNDERFLOW

Go to Step 8

[END OF IF]

Step 2 : CET PTR = START

step 3 : Repeat Step 4 and 5 while

PTR - NEXT ! = START

Step 4 : SET PREPER = PTR

SHOP 5 : SET PTR - PTR - NEXT

LEND OF LOOP]

Step 6 & GET PREPTR - NEXT = START

Step 7 : FREE PTA

Step & & EXIT



(Approved by AICTE New Delhi & Govt. of Maharushtra. Affiliated to University of Mumbai) (Religious Jain Minority)

Semester: Subject: Academic Year: ____

```
* C Program to Demonstrate Circular Single Linked List
1.
2.
3. #include <stdio.h>
4. #include <stdlib.h>
5.
6. struct node
7. {
       int data;
8.
9.
       struct node *link;
10. };
11.
12. struct node *head = NULL, *x, *y, *z;
14. void create();
15. void ins_at_beg();
16. void ins_at_pos();
17. void del_at_beg();
18. void del_at_pos();
19. void traverse();
20. void search();
21. void sort();
22. void update();
23. void rev_traverse(struct node *p);
25. void main()
26. {
27.
        int ch;
28.
        printf("\n 1.Creation \n 2.Insertion at beginning \n 3.Insertion at
29.
remaining");
30.
        printf("\n4.Deletion at beginning \n5.Deletion at remaining
\n6.traverse");
        printf("\n7.Search\n8.sort\n9.update\n10.Exit\n");
31.
32.
        while (1)
33.
        {
34.
            printf("\n Enter your choice:");
            scanf("%d", &ch);
35.
36.
             switch(ch)
37.
             {
38.
            case 1:
39.
                 create();
40.
                 break;
41.
            case 2:
42.
                 ins_at_beg();
43.
                 break;
44.
            case 3:
                 ins_at_pos();
45.
46.
                break;
47.
            case 4:
48.
                 del_at_beg();
                 break;
49.
50.
            case 5:
```



(Approved by AICTE New Delhi & Govt. of Maharushtra. Affiliated to University of Mumbol) (Religious Jain Minority)

Subject: _ Academic Year: _ Semester: 51. del_at_pos(); 52. break; 53. case 6: 54. traverse(); 55. break; 56. case 7: search(); 57. 58. break; 59. case 8: 60. sort(); 61. break; case 9: 62. 63. update(); 64. break; 65. case 10: rev_traverse(head); 66. 67. break; 68. default: 69. exit(0); 70. } 71. } 72. } 73. 74. /*Function to create a new circular linked list*/ 75. void create() 76. { 77. int c; 78. 79. x = (struct node*)malloc(sizeof(struct node)); 80. printf("\n Enter the data:"); scanf("%d", &x->data); 81. $x - \sinh x = x;$ 82. head = x;83. printf(" $\ \ \$ n If you wish to continue press 1 otherwise 0:"); 84. 85. scanf("%d", &c); while (c != 0)86. 87. y = (struct node*)malloc(sizeof(struct node)); 88. printf("\n Enter the data:"); 89. scanf("%d", &y->data); 90. $x - \sinh = y;$ 91. 92. v->link = head; x = y;93. printf("\n If you wish to continue press 1 otherwise 0:"); 94. scanf("%d", &c); 95. 96. } 97. } 98. 99. /*Function to insert an element at the begining of the list*/ 100. 101. void ins_at_beg() 102. { 103. x = head;y = (struct node*)malloc(sizeof(struct node)); 104. 105. printf("\n Enter the data:"); scanf("%d", &y->data); 106.



(Approved by AICTE New Delhi & Govt. of Maharushtra. Affiliated to University of Mumbol) (Religious Jain Minority)

Academic Year: _ Semester: Subject: while (x->link != head) 107. 108. 109. x = x->link;110. } $x - \sinh = y;$ 111. y->link = head; 112. 113. head = y; 114. } 115. 116. /*Function to insert an element at any position the list*/ 117. 118. void ins_at_pos() 119. { 120. struct node *ptr; 121. int c = 1, pos, count = 1; 122. 123. y = (struct node*)malloc(sizeof(struct node)); 124. if (head == NULL) 125. { printf("cannot enter an element at this place"); 126. 127. } printf("\n Enter the data:"); 128. scanf("%d", &y->data); 129. printf("\n Enter the position to be inserted:"); 130. scanf("%d", &pos); 131. 132. x = head;133. ptr = head;134. while (ptr->link != head) 135. { 136. count++; 137. ptr = ptr->link; 138. } 139. count++; 140. if (pos > count) 141. 142. printf("OUT OF BOUND"); 143. return; 144. 145. while (c < pos)146. 147. z = x;148. $x = x - \sinh;$ 149. C++; 150. 151. y->link = x; 152. z->link = y; **153.** } 154. 155. /*Function to delete an element at any begining of the list*/ 156. 157. void del_at_beg() **158.** { 159. if (head == NULL) printf("\n List is empty"); 160. 161. else 162. {