Date 07/3/22
Page

UNIT 1 :-

Introduction to data structure : Basic Terminology, classification of data structures, operations on data structures.
Linked Lists :- Singly linked list, Representation in memory, operations on a single linked list, circular linked lists, Doubly linked list.

UNIT 2 :-

Linear Data Structures - stacks :- Introduction to stacks, Array representation of stacks, operations on a stacks, Applications of stack Infix-to-Postfix transformation, evaluating Postfix Expressions Queues :- Introduction to Queues, Array representation of Queues, operations on a Queue, Type of Queues Dequeue circular Queues, Applications of Queues Round Robin Algorithm, Sparse matrix & its Representation

UNIT 3 :-

Searching and Sorting :- Introduction to different sorting techniques - Selection, insertion, bubble, quick and merge Introduction to different searching techniques - sequential & binary.

UNIT 4 :-

Non-Linear Data Structures : Trees :- Basic Terminologies, Definition & concepts of Binary trees, Representations of a Binary tree using Arrays & linked lists, operations on a Binary tree - Insertion, Deletion, Traversals, Types of Binary trees

Date / /
Page

Graphs :- Graph terminologies representation of Graph - set, Linked, Matrix, Graph traversals.

Unit 1

Date 14/3/22

* What is Data Structure?

Ans data structure دنیا میں کے Computer System کو data موجود اس میں کہ یہ ~ meaning کے store کر سکتے ہیں memory میں organized way ہے جسکے استعمال سے ہم اس data کو easy سے ضرورت پیش آنے پر اسکو use کر سکتے ہیں۔ اس طرح کو data organized کے طریقے سے store کر سکتے ہیں اس کے عمل کو ہم access کرنے کے [action] data structure کہتے ہے۔

* what is data and Information?

Ans data کو کوئی موجود میں store Computer کہتے ہے۔

For Example :- abeiaL si eman-ym

اس موجود data میں سے کام میں انے والے data کو ہم Information meaningful data طلب کہتے ہے۔

For Example :- My Name is Laieba.

* Classification of Data Structure

divede میں type دو کو Data Structure کیجے گا۔

1. Linear Data Structure [data والا ترتیب]
2. Non-Linear Data Structure.
[غیر ترتیب والا دینا۔]

1 Linear Data Structure

data ہے data structure اس قسم کے
- ہوتا ہے Sequence میں

Types

1. Array
2. Linked list
3. Stack
4. Queue

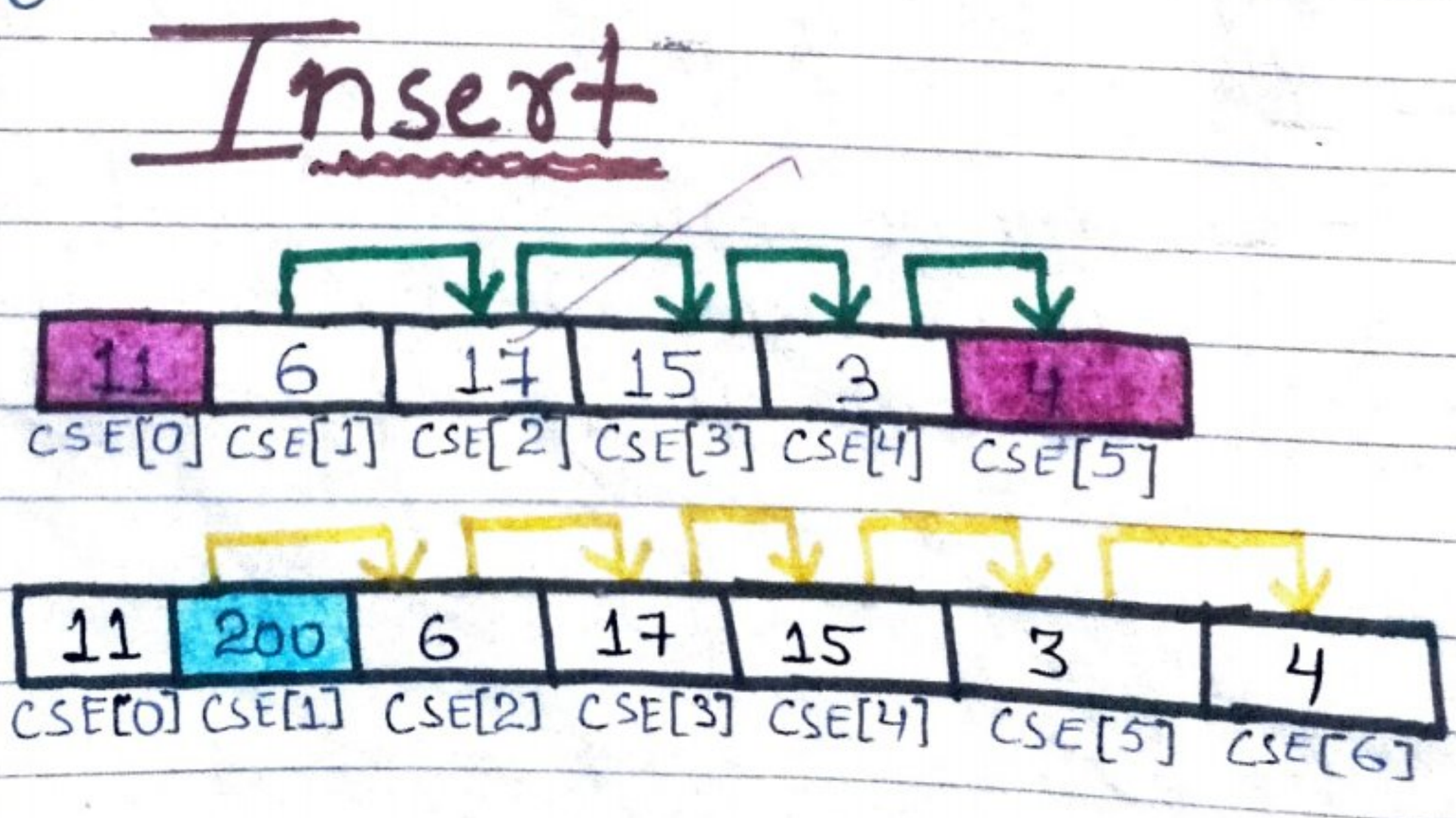
2. Non-Linear Data Structure

data ہے data structure اس قسم کے
- ہوتا ہے Sequence میں نہیں

Types

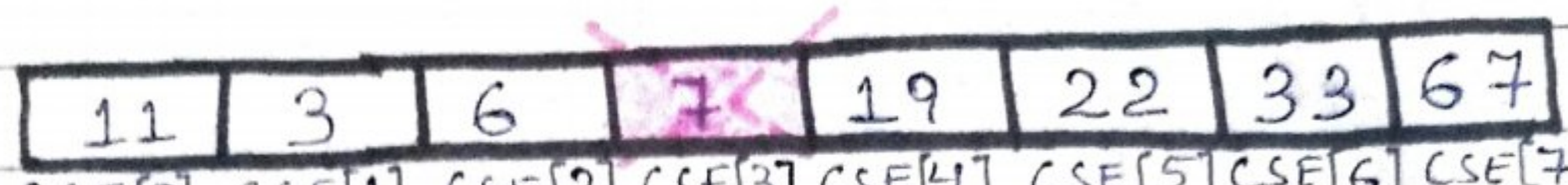
1. Tree
2. Tables
3. Sets
4. Graphs

* Array
Collection of similar Element is called Array.

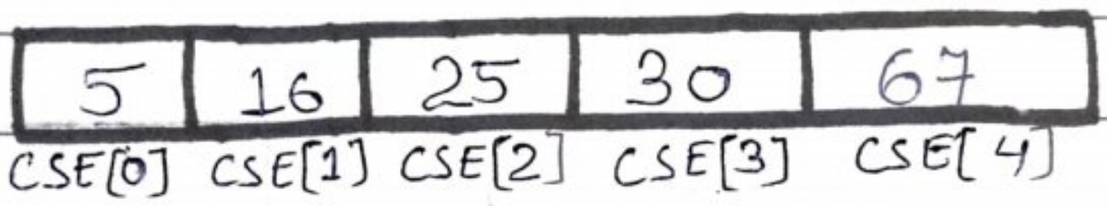


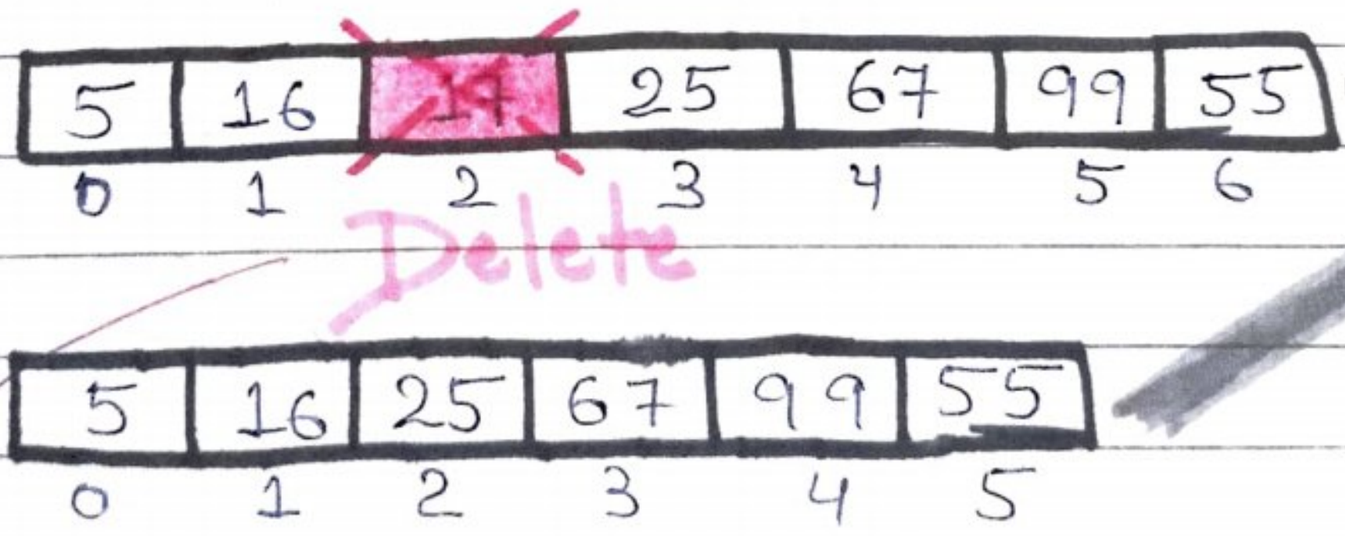
Date ___/___/___
Page ___

Deleting



Addressing





Array

Array جس میں ایک ہی قسم کے data کو
collect ہوتا ہے یا ایک sequential mapping
میں ہوتا ہے sequential mapping مطلب
یہ data کے elements کو ایک ہی جگہ fixed
ہوتی ہے اس وجہ سے Array میں خامیاں
[disadvantage.] موجود ہے -

Problem of Array

Array میں elements کو Insert کرنا اور
delete کرنا element کو proved costly
ہوتا ہے جس کے reason نیچے شکل میں
بتائے گئے ہے -

22	25	29	30	35	38	50	55	67
CSE[0]	CSE[1]	CSE[2]	CSE[3]	CSE[4]	CSE[5]	CSE[6]	CSE[7]	CSE[8]

اس خامی کی وجہ سے Array کی جگہ اسکا change
کیا جاتا ہیں link list کو use کیا جاتا ہے
جس سے Array کی خامی دور ہو سکتے ہے -

Problem :-

1. Address Issue
2. Deleting Issue
3. Shifting of data Issue
4. Size Issue

Date ___/___/___
Page _____

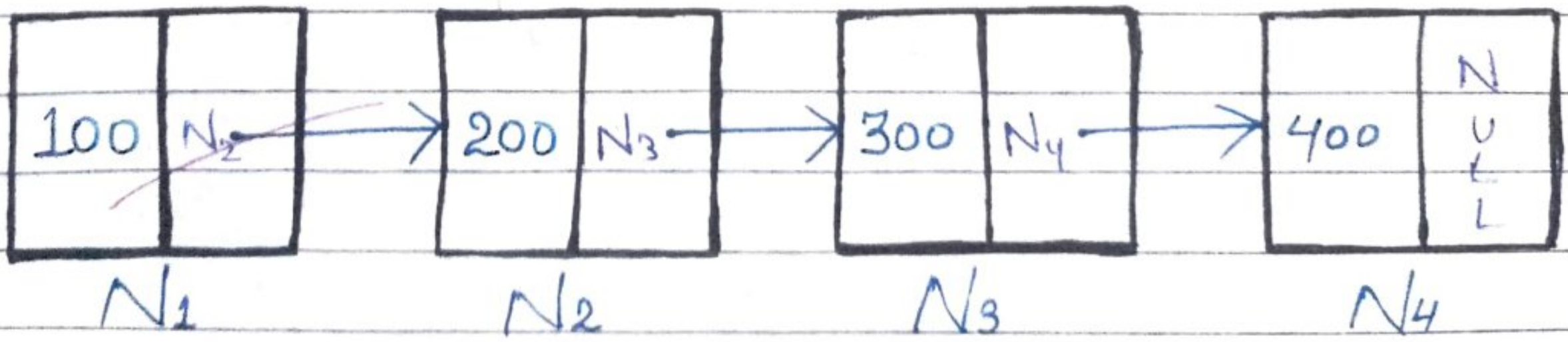
Linked List

اس میں جگہ fixed نہیں ہوتی ہے - بلکہ اس میں موجود دو حصے ہوتی ہیں ایک حصے میں data اور دوسرے حصے میں link موجود ہوتا ہے جسکو ہم node کے name سے جانتے ہے -

Link والے حصے کی help سے ہم element کو memory کے کسی بھی حصے میں store کرسکتے ہے -

Link یا connection بنانے کے لیے ایک node میں دوسری node کا address لکھنا ہوتا ہے - اس طرح آپس میں node link کے ذریعے ایک دوسرے سے connect ہوتا ہے جسکو Linked List کہا جاتا ہے -

اس کو figure میں بتایا گیا ہے -



Linked List

جیسا کی figure میں بنایا گیا ہیں node 1 سے
node 4 جسکو ہم ایک element بھی کہتے ہے -
ہر ایک node میں دو حصے موجود ہوتے ہیں
linked list کے end [اختام] بتانے کے لیے
آخر والے node کے حصے میں
capital letter سے NULL لکھنا ہوتا ہے - جس سے
computer کو information ملتے ہیں کی یا list
[End of list] ہے -

Date / /
Page

Data Structure

Linear Data Structure.

- Array
- Linked List
- Stack
- Queue

Non-Linear Data Structure.

- Tree
- Graphs
- Tables
- Sets

Date ___/___/___
Page ______

* Basic operation on Linked List

1. Create a list
2. Traversing a list [shift ادھر سے اٹھا کے کسی اور]
3. Counting items in the list.
4. Printing the list (Sub list)
5. Looking up an item for editing.
6. Insert an item.
7. Delete an item.
8. Concatinating an item.
[دو چیز کو ملانا]

* **Advantage of Linked List**
1. Linked List ایک dynamic data structure ہے جو program کے execution کے دوران grow بھی ہو سکتا ہے اور shrink [کم] بھی ہوتا ہے۔

2. memory کی waste نہیں کرتے ہیں جس وقت memory کی ضرورت پیش ہو تبھی یہ memory کو use کرتے ہے۔

3. Dynamic memory [DMA] کی وجہ سے memory کو بہترین انداز میں use کر سکتے ہے۔

4. Data elements کو insert اور delete کرنے میں بہت easy ہوتا ہے۔

5. Insert اور delete والا عمل بہت کم وقت سے انجام دے سکتے ہے۔
[result]

CamScanner

Date 1/1/
Page

Type of Linked List are :-

1. Single Linked List.
2. Circular Single Linked List.
3. Double Linked List.
4. Circular Double Linked List.

1. Single Linked List

یہ ایک Linear Data Structure ہے جس دو part ہوتے ہیں. ایک part ہے data part کے لیے ہوتا ہے اور دوسرا part ہوتا ہے Link کے لیے ہوتا ہے۔ اس میں Single linked ہے اس میں Backward پر نہیں جاسکتے ہے۔ اس کے last والے node میں NULL ہوتا ہے۔

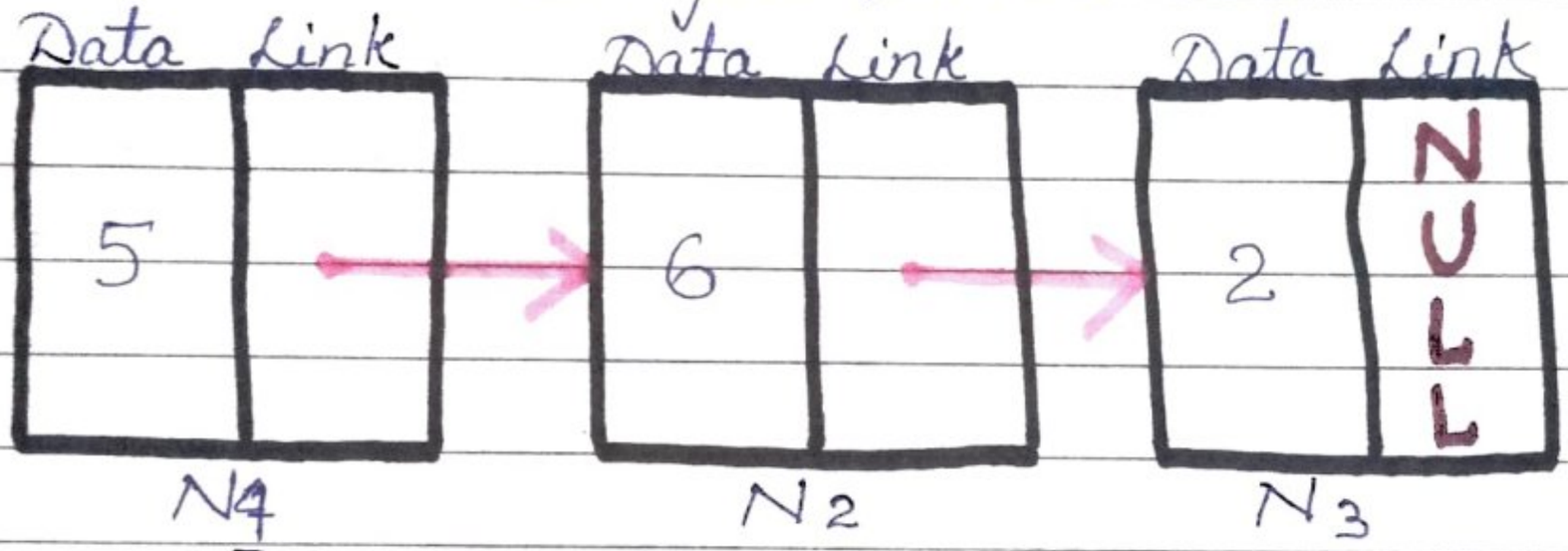
Date / /
Page

Insertation of Elements

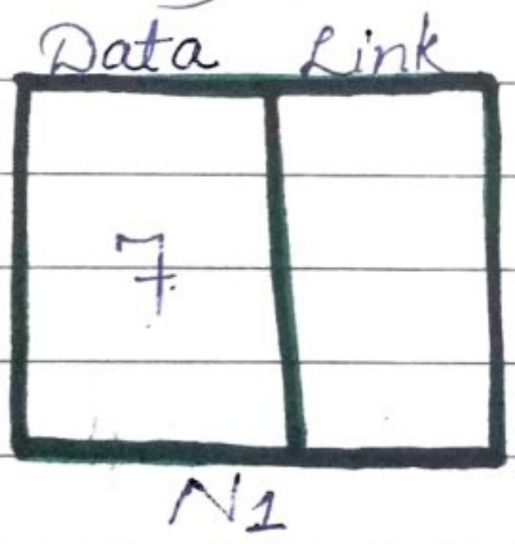
1. Insert at first
2. Insert at Middle
3. Insert at Last.

1. Insert at first

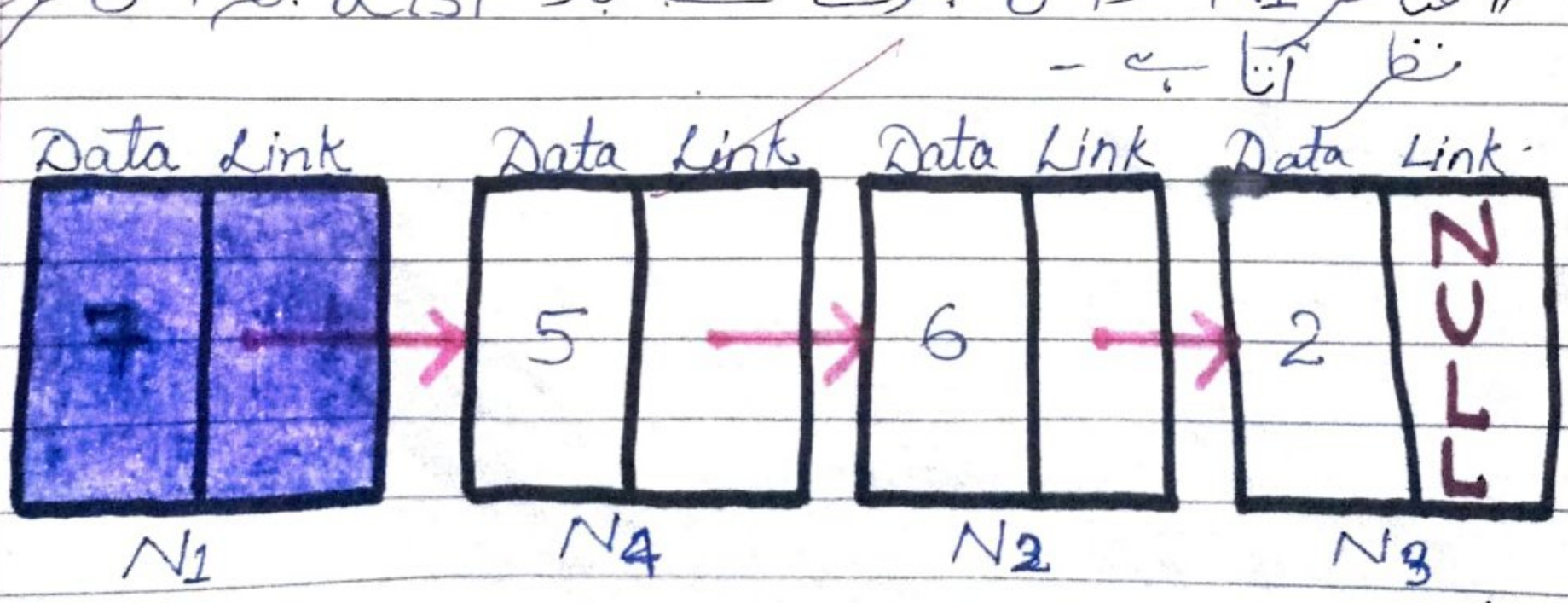
Single Linked List کا موجود



N1 [Element] # داخل ہونے والا عناصر



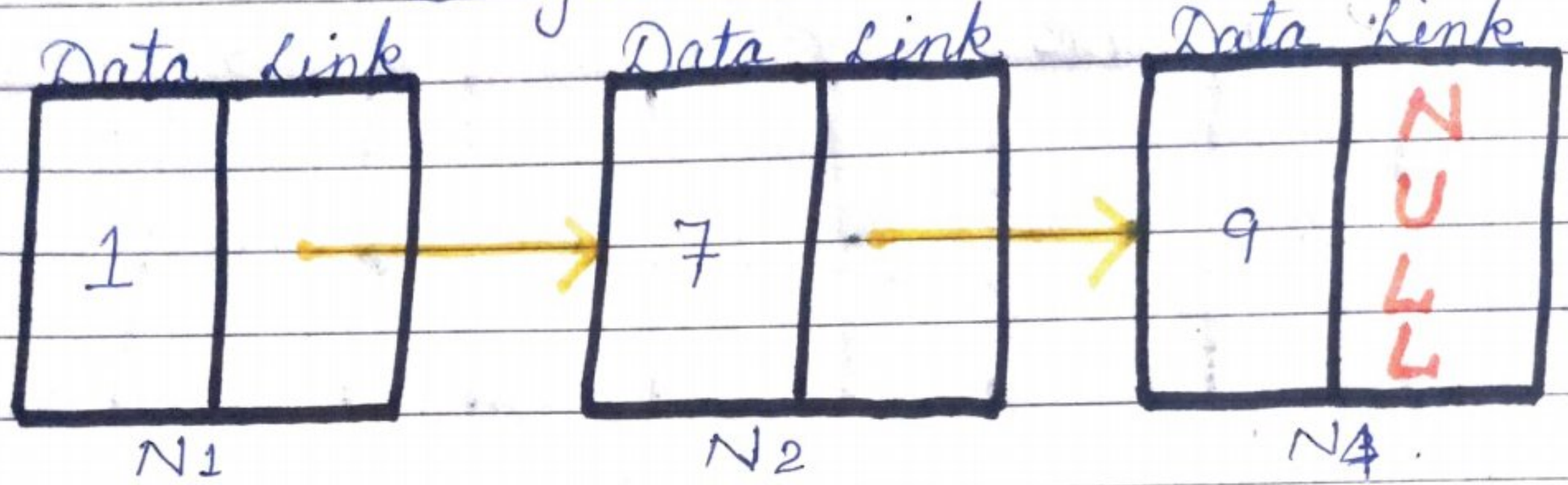
عناصر N1 داخل ہونے کے بعد List کچھ اس طرح نظر آتا ہے -



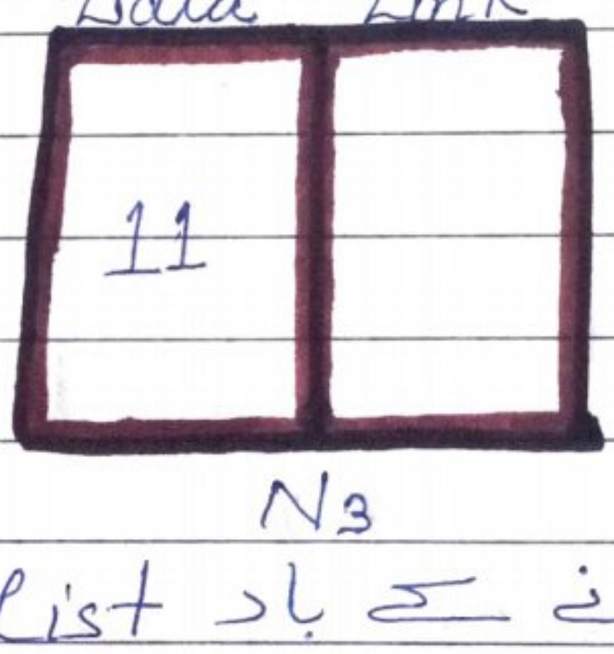
Date / /
Page

2. Insert at Middle

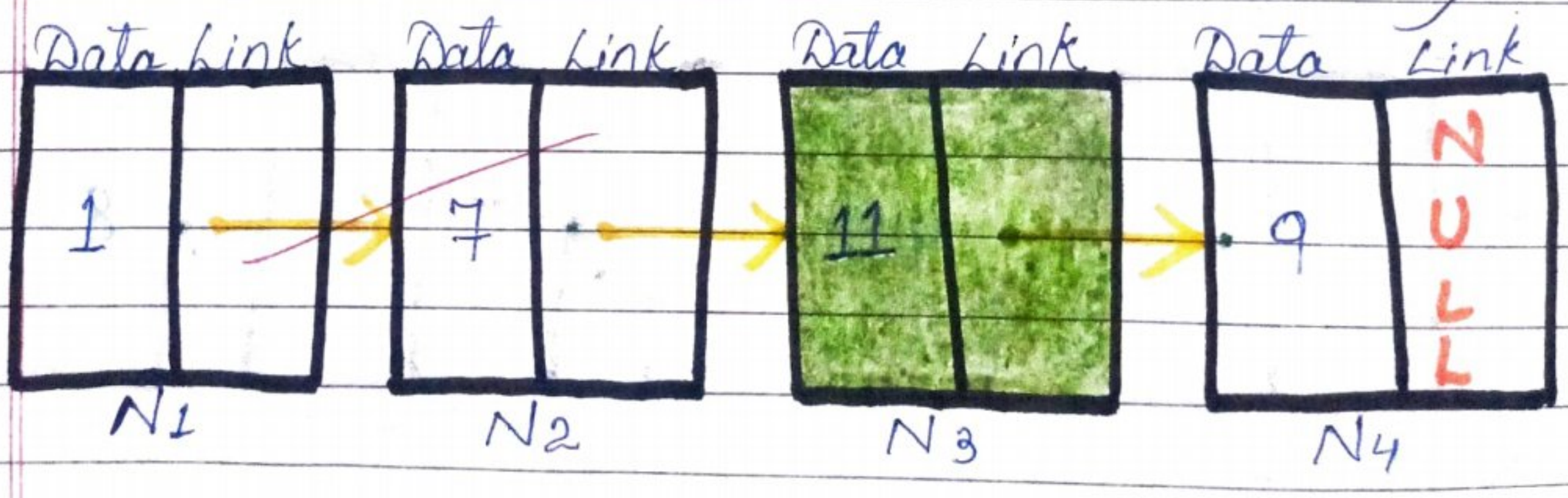
* Single Linked List کا وجود



* داخل ہونے والا عناصر [Element] N3

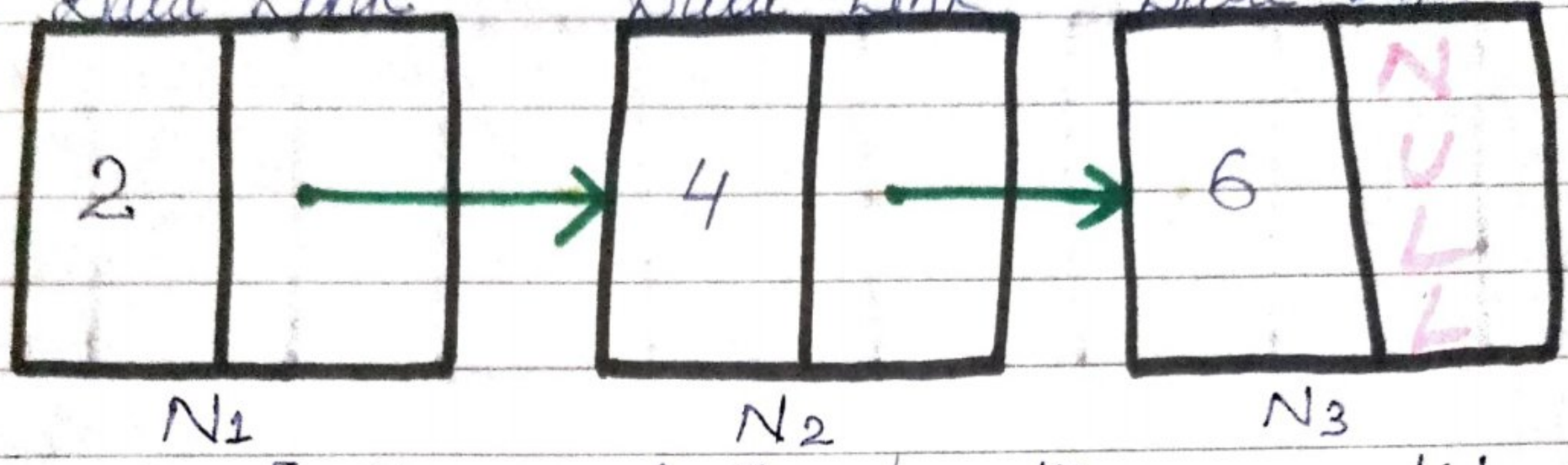


* عناصر N3 داخل ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔

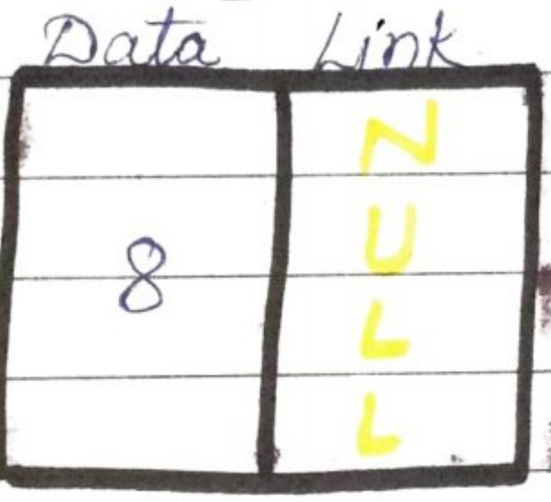


3. Insert at Last

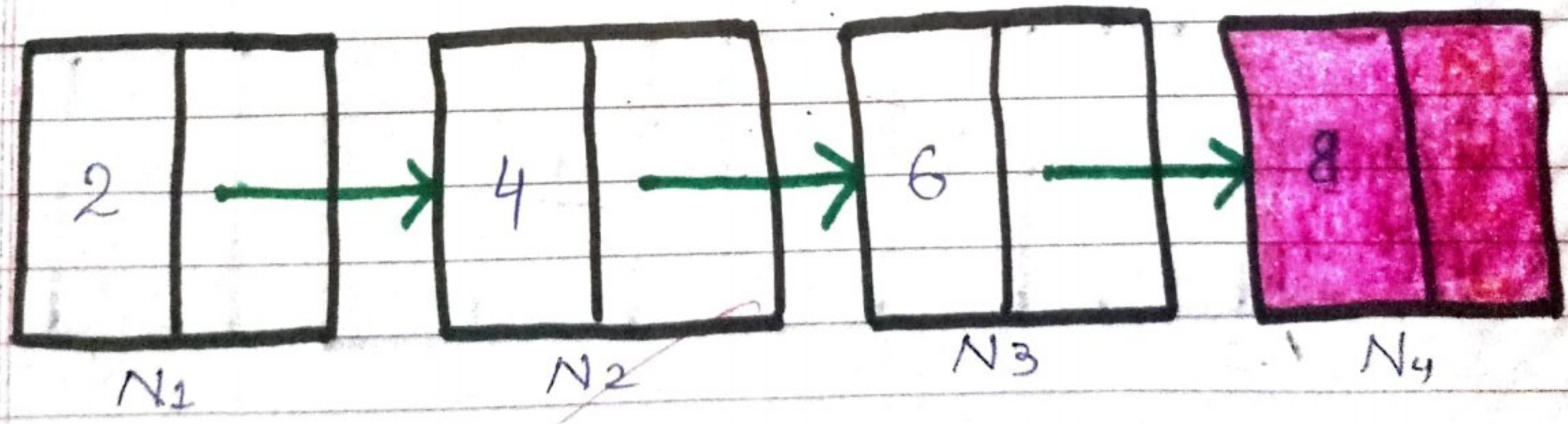
موجودہ Single Linked List



• * داخل ہونے والا عناصر N4 [Element]



* عناصر N4 داخل ہونے کے بعد List کچھ اس طرح نظر آتا ہے —

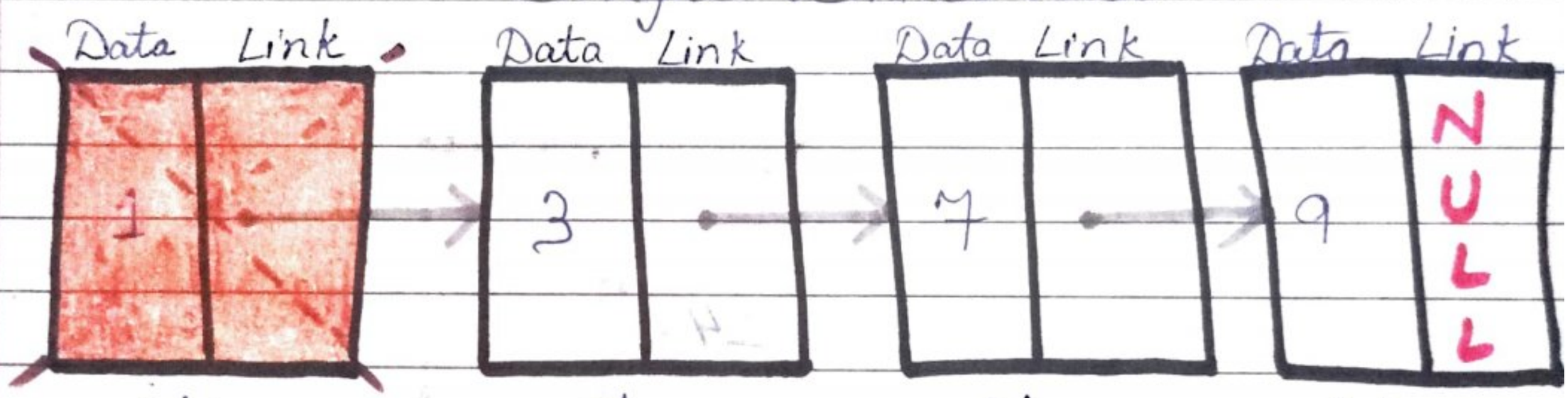


Deleting of Elements

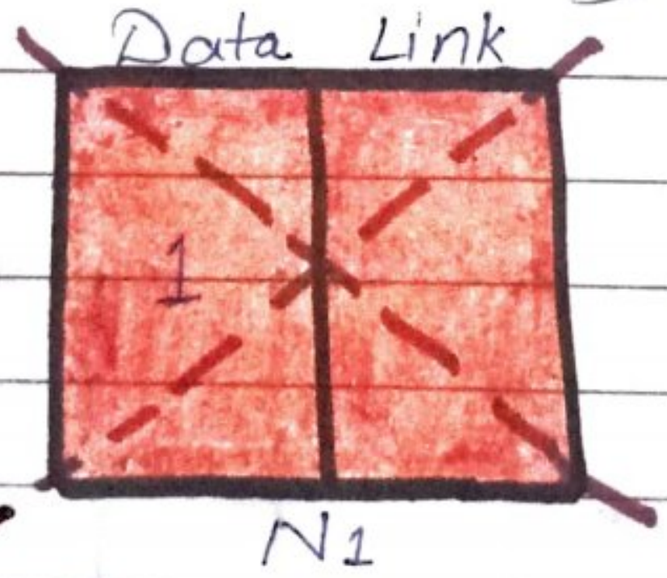
1. Delete at first
2. Delete at Middle
3. Delete at Last

1 Delete at first

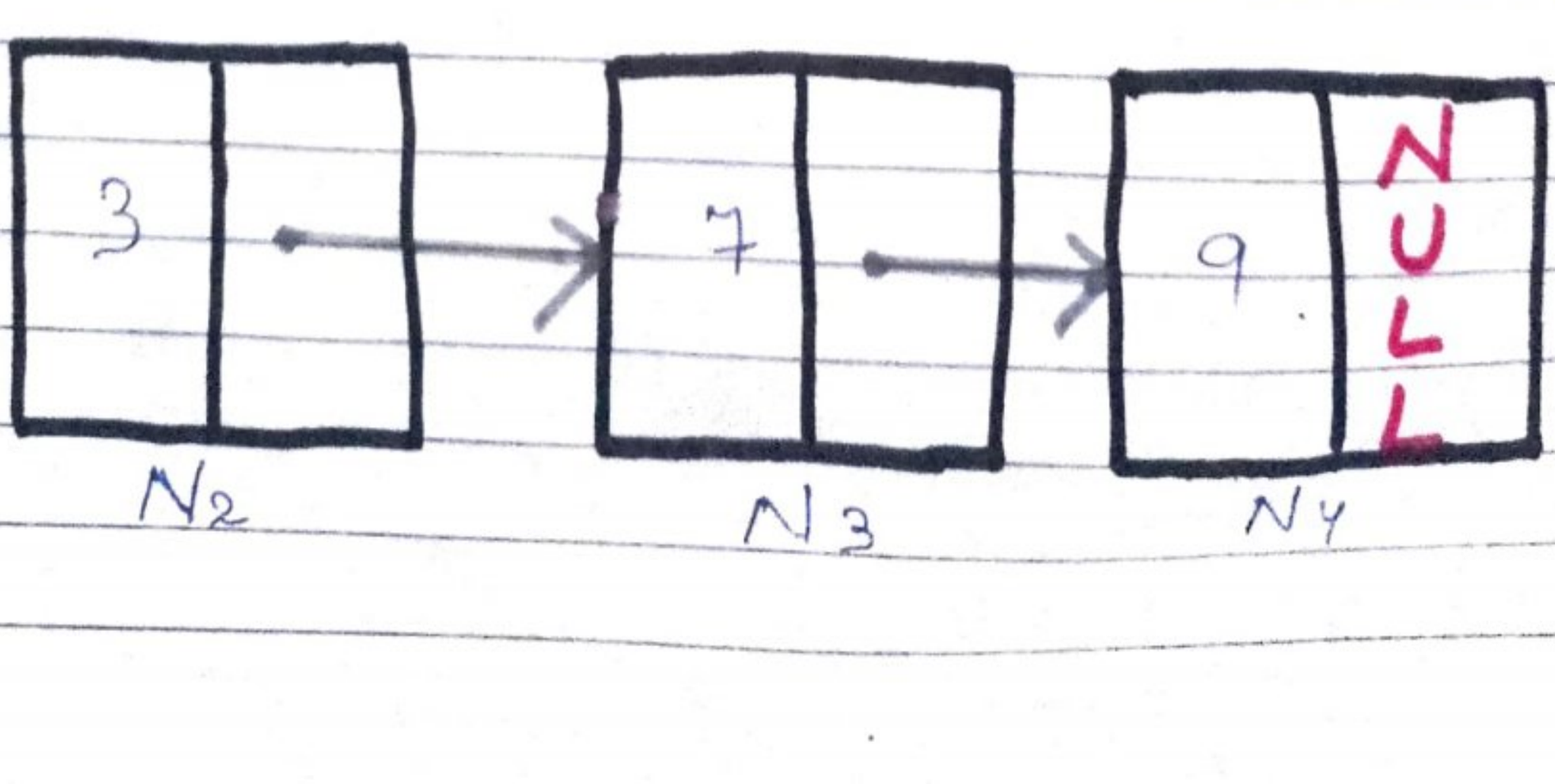
Single Linked List ہوتی ہے #



خارج ہونے والے عناصر N1 [Element]



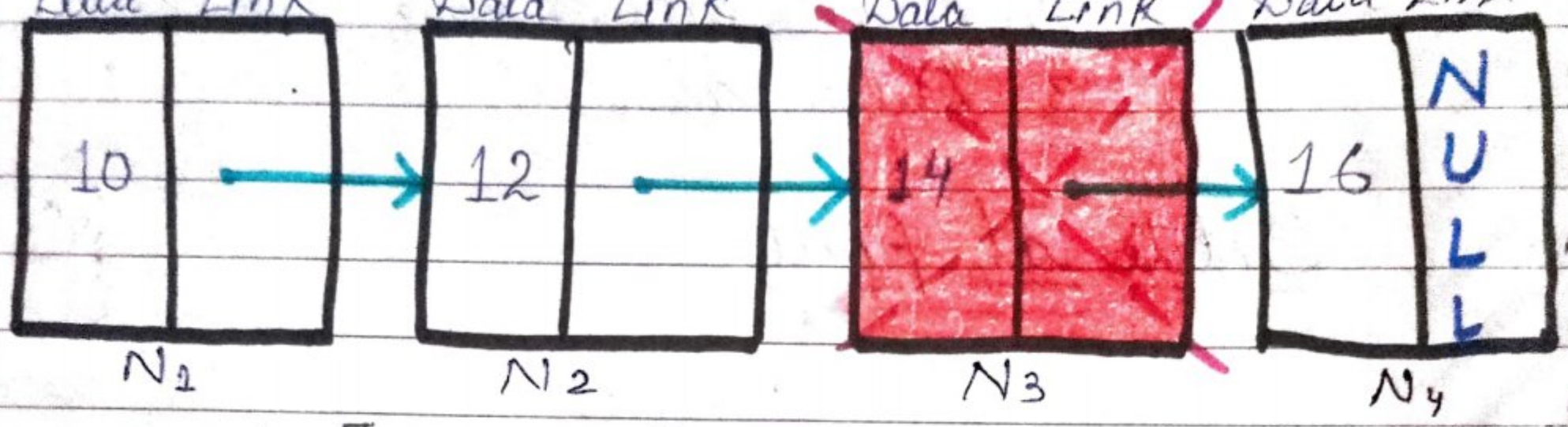
عناصر N1 خارج ہونے کے بعد List کچھ اس طرح نظر آتا ہے -



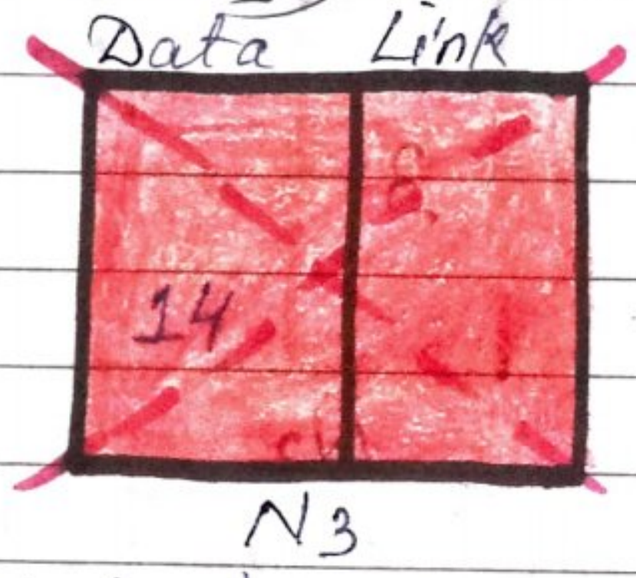
Date ___/___/___
Page _____

2. Delete at Middle

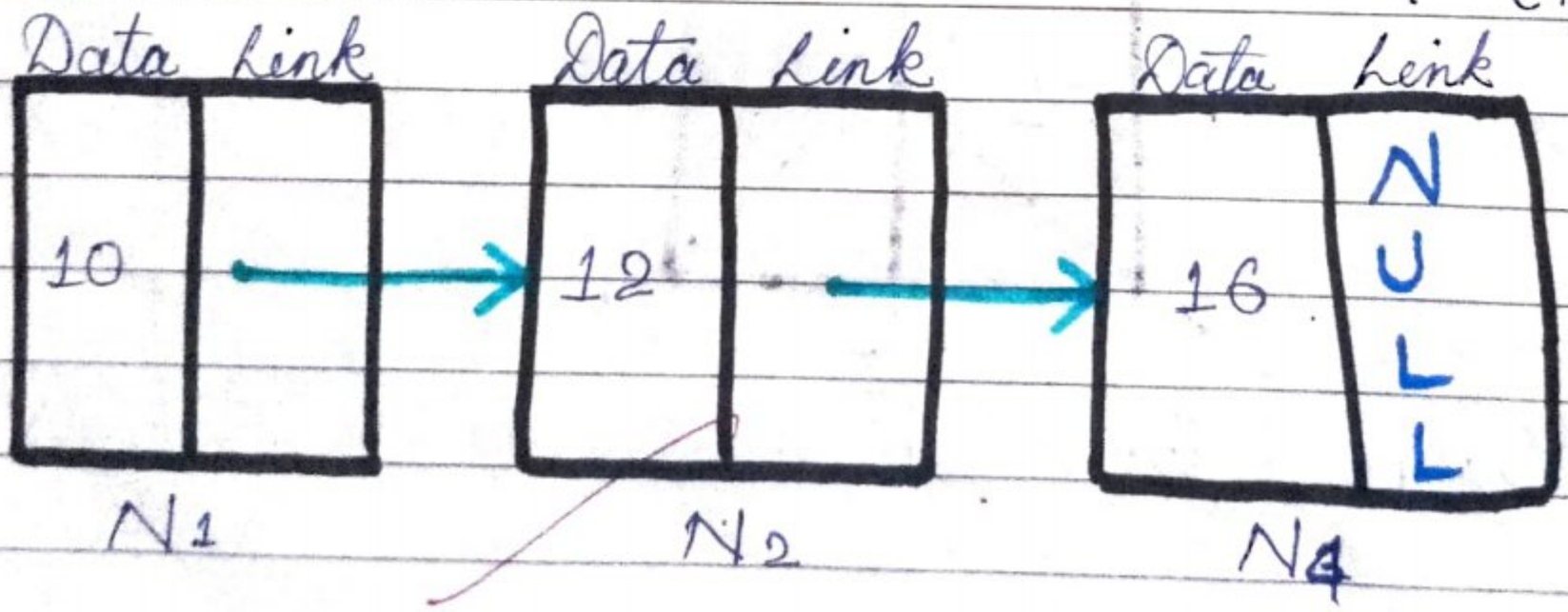
Single Linked List کو #



N3 [Element] والے عناصر خارج ہونے #



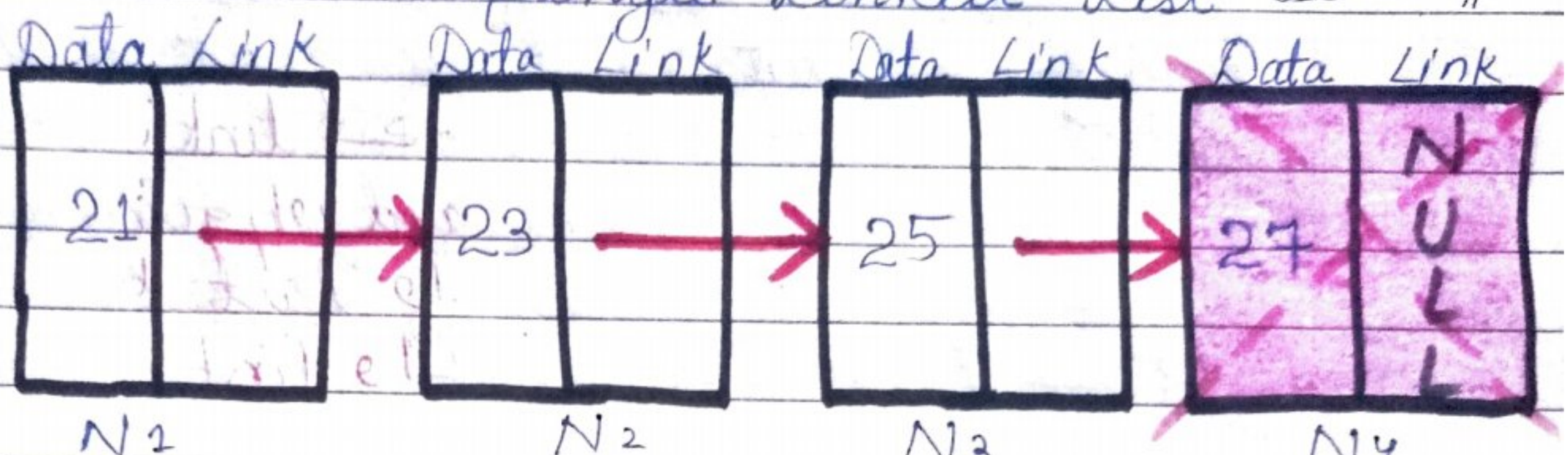
#عناصر N3 خارج ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔



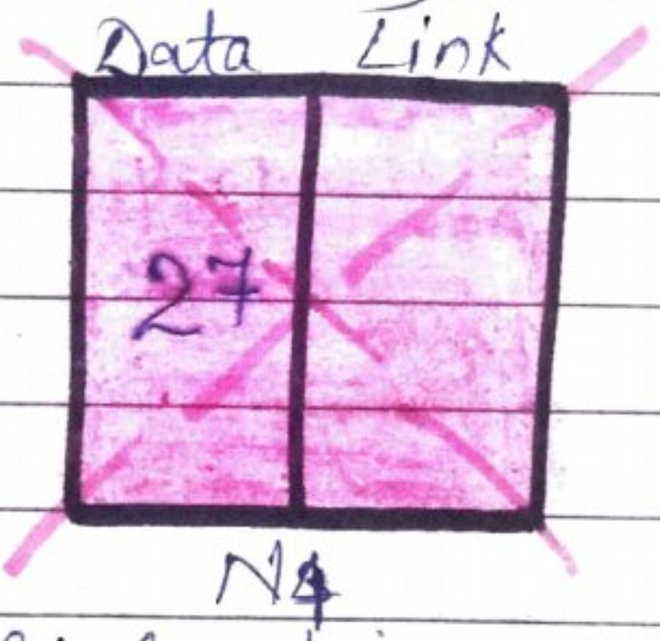
Date ___/___/___
Page _____

3. Delete at Last

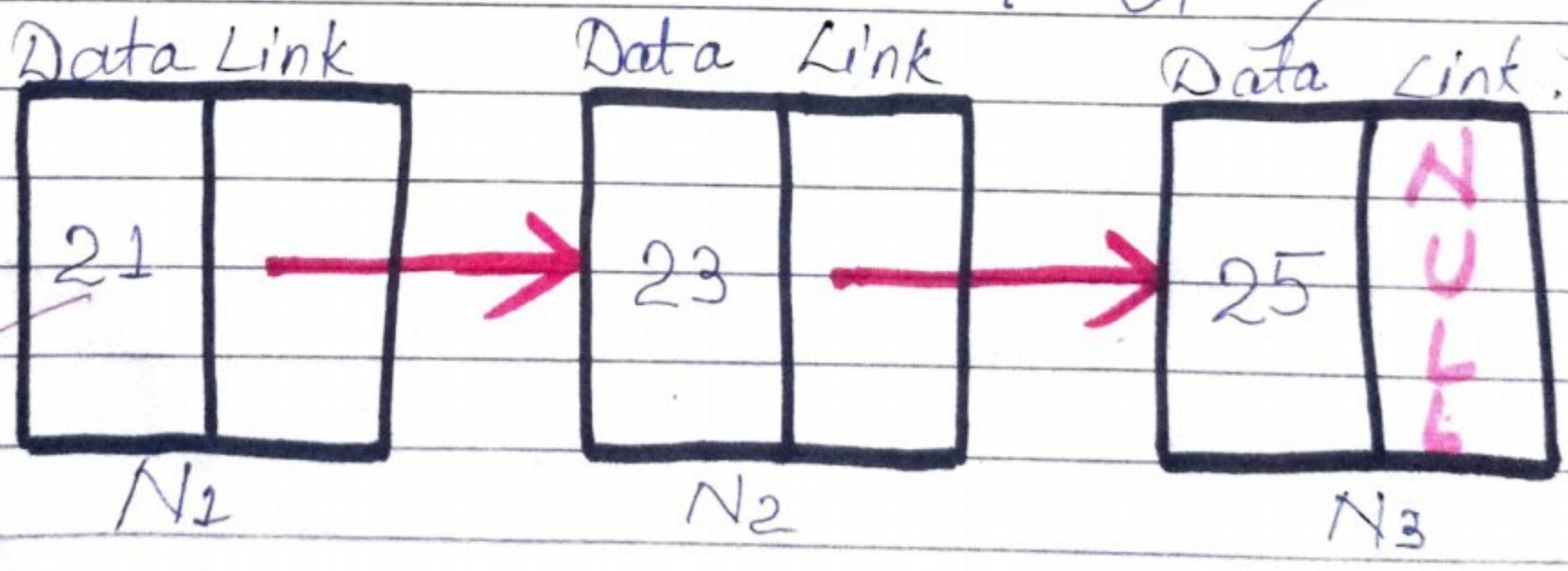
Single Linked List کی مثال #



N4 [Elements] عناصر # خارج ہونے والے



عناصر N4 خارج ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔



Date / /
Page

2 Circular Single Linked List

یہ ایک Linear data structure ہے جس میں دو data part ہوتا ہے ایک کے لیے اور دوسرا link کے لیے ہوتا ہے - اس میں ہم Backward اور forward دونوں جا سکتے ہے - اس کا Link کا node والے Last والا ~~node~~ Link کے node والے first part سے جڑا ہوتا ہے - اس میں NULL نہیں ہوتا ہے -

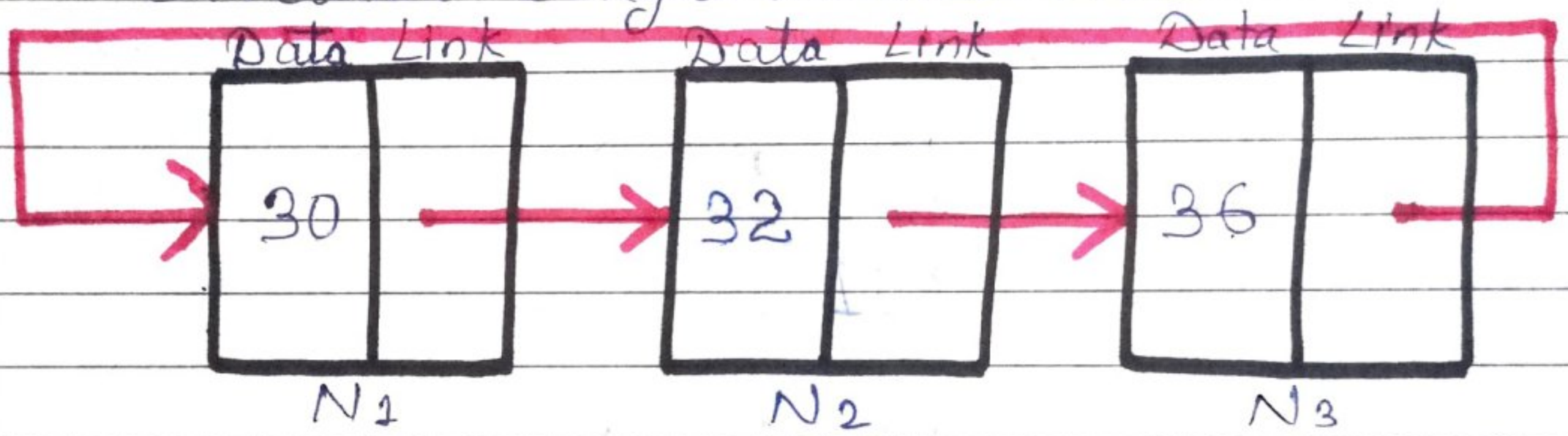
Date / /
Page

Insertation of Elements

1. Insert at first
2. Insert at Middle
3. Insert at Last .

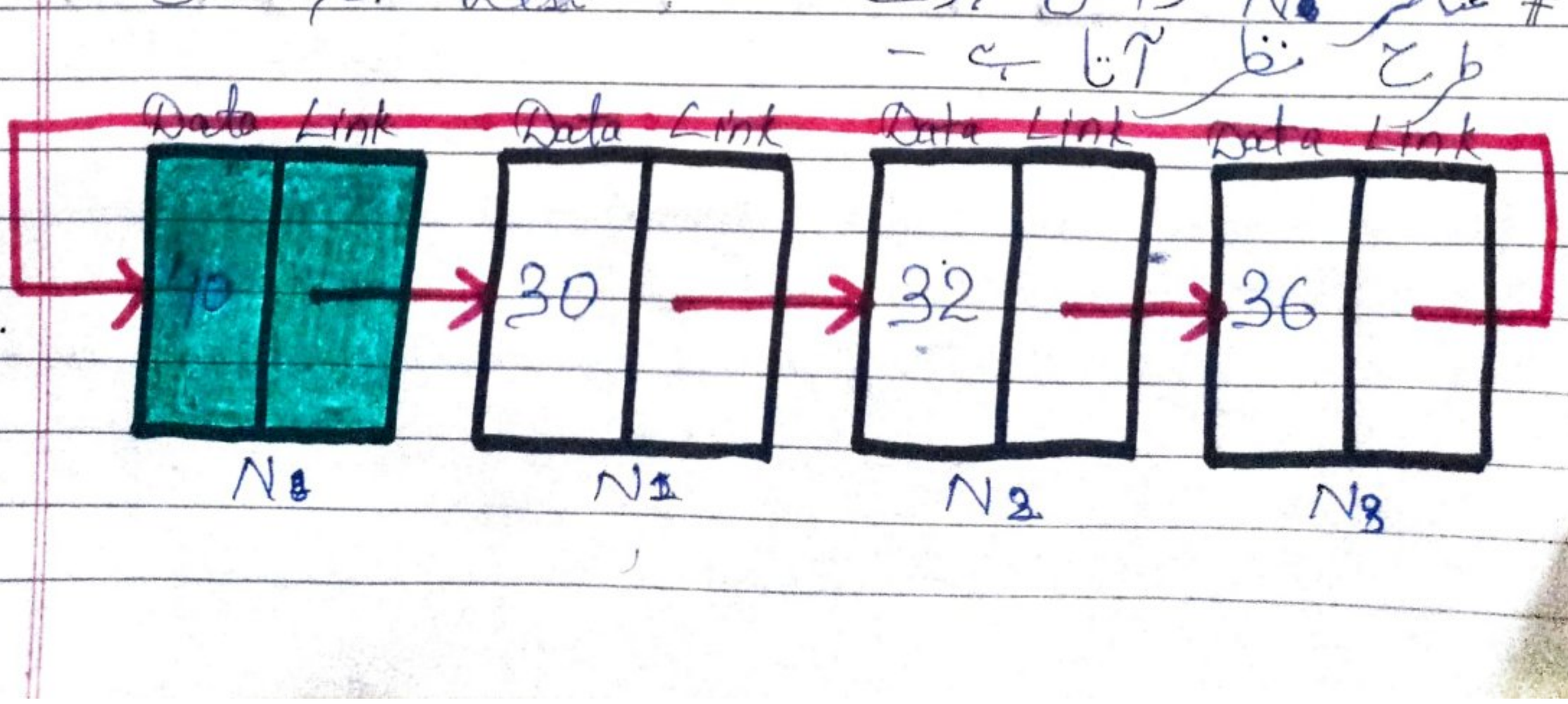
1. Insert at first

Circular Single Linked List کی طرح #



داخل ہونے والے عناصر کو Node [Element]

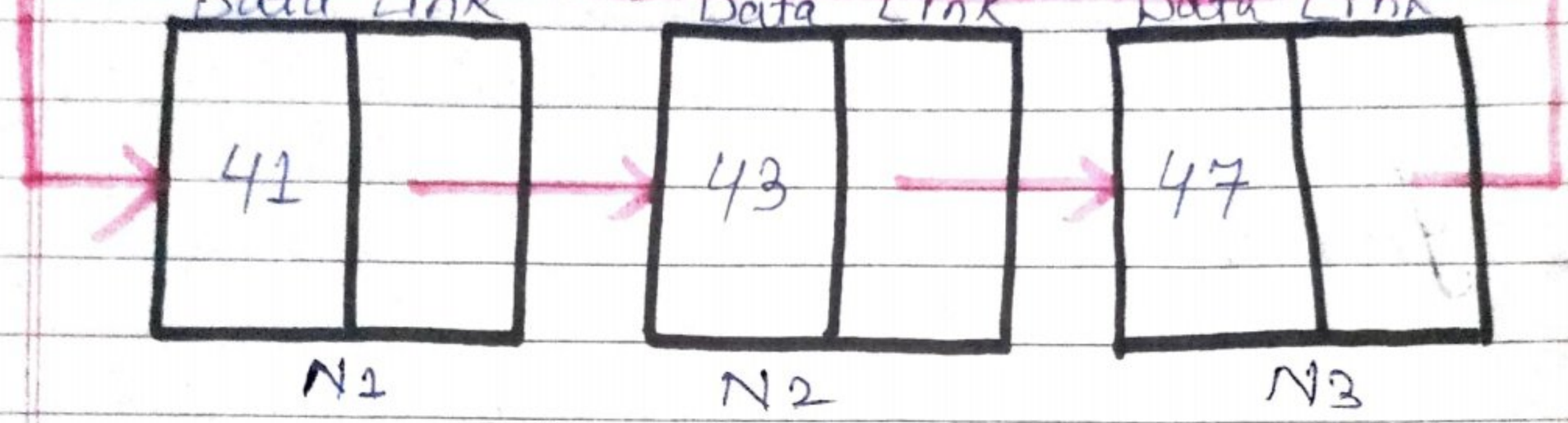
عناصر کو Node داخل ہونے کے بعد List کچھ اس طرح نظر آتا ہے -



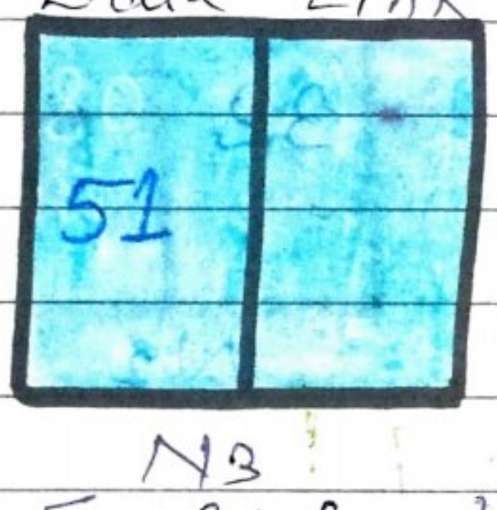
Date / /
Page

2 Insert at Middle

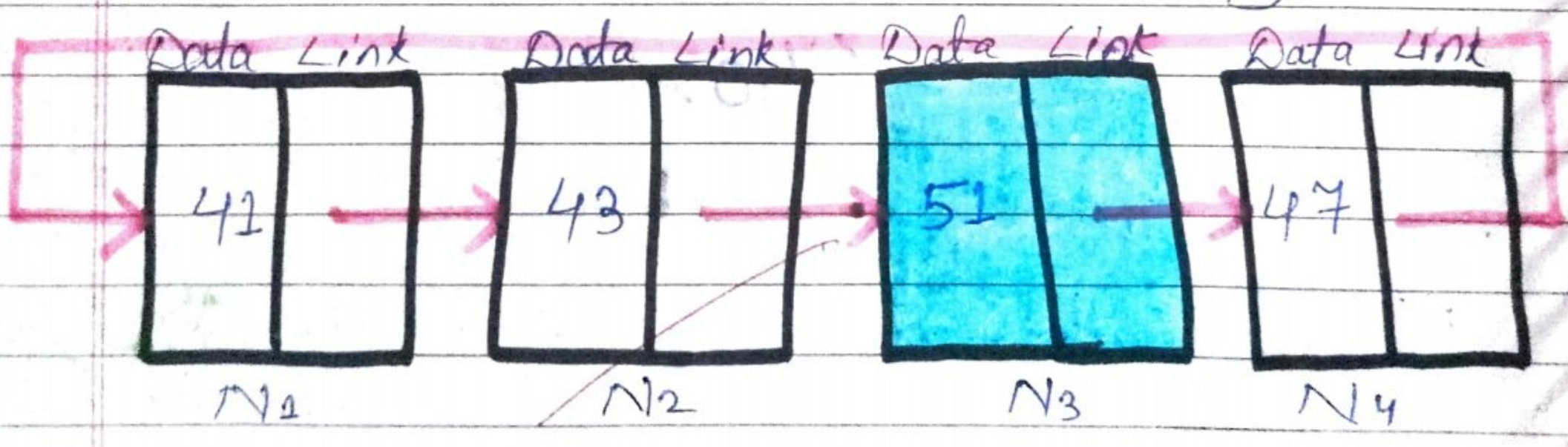
Circular Single Linked List کے اندر +



N3 [Element] عناصر + داخل ہونے والے



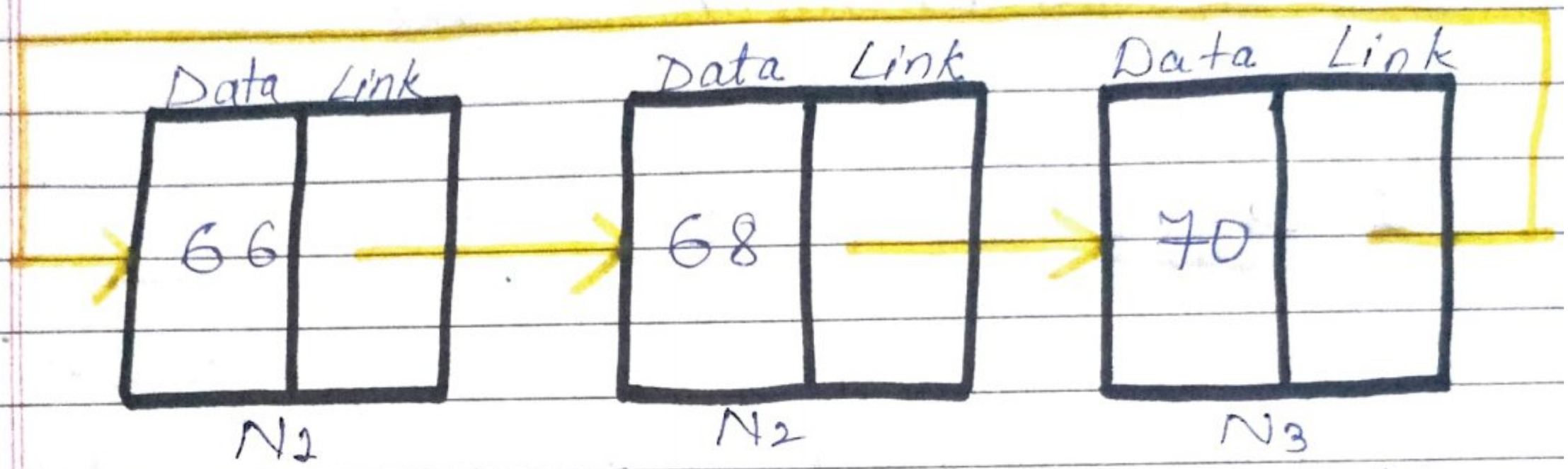
+ عناصر N3 داخل ہونے بعد List کے اس طرح سے نظر آتا ہے —



Date 1 1
Page

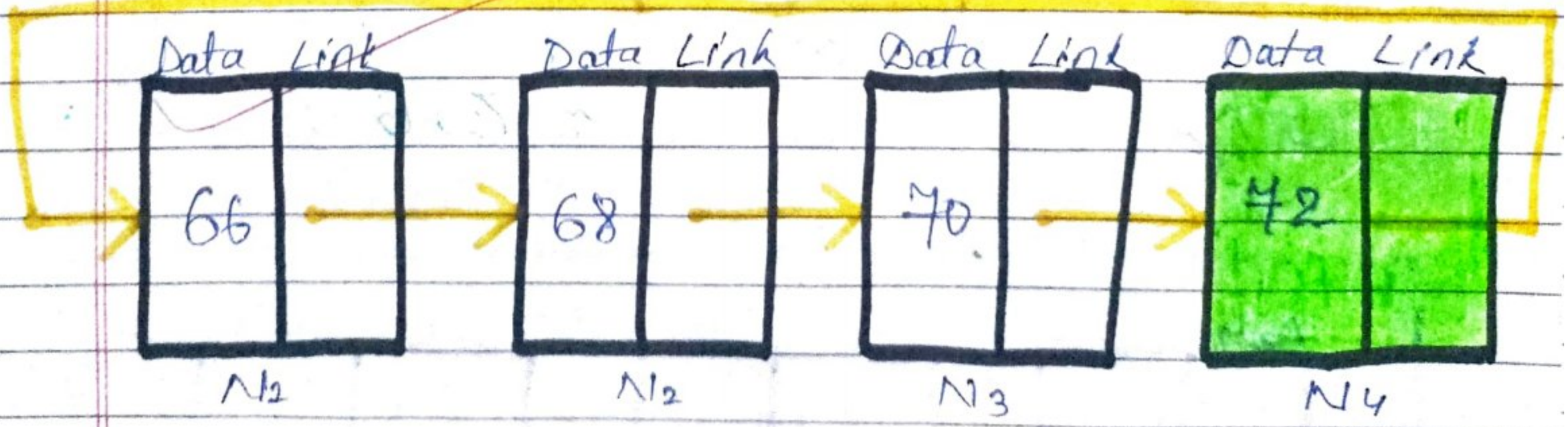
3. Insert at Last

Circular Single Linked List میں نوڈ کو +



+ داخل ہونے والے عناصر N4 [Element]

عناصر N4 داخل ہونے کے بعد List کو اس طرح نظر آتا ہے۔



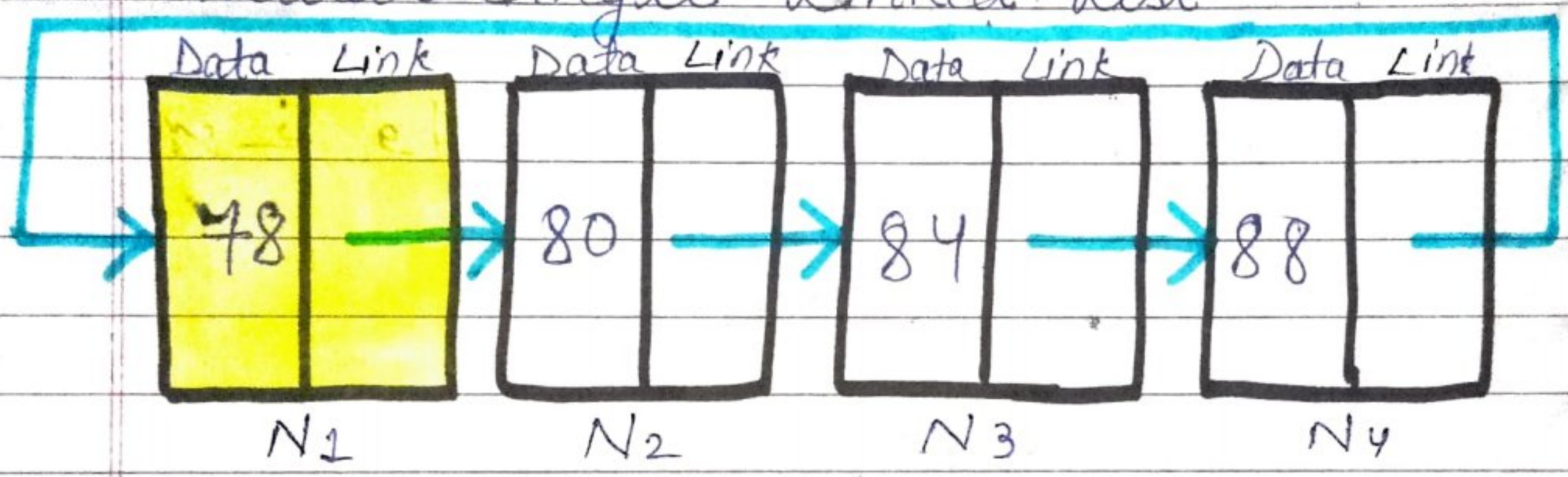
Date / /
Page

Deleting of Element

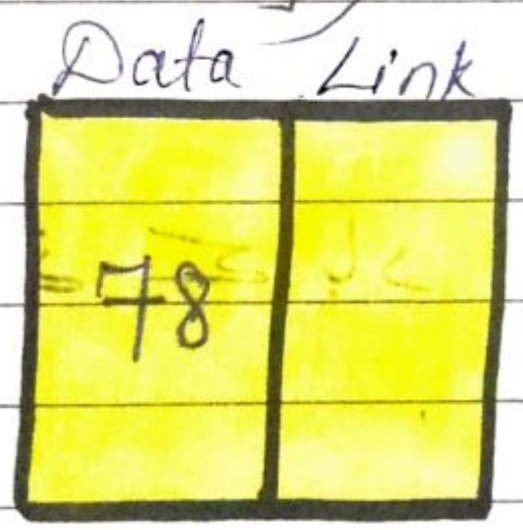
1. Delete at first
2. Delete at Middle
3. Delete at Last.

1 Delete at first

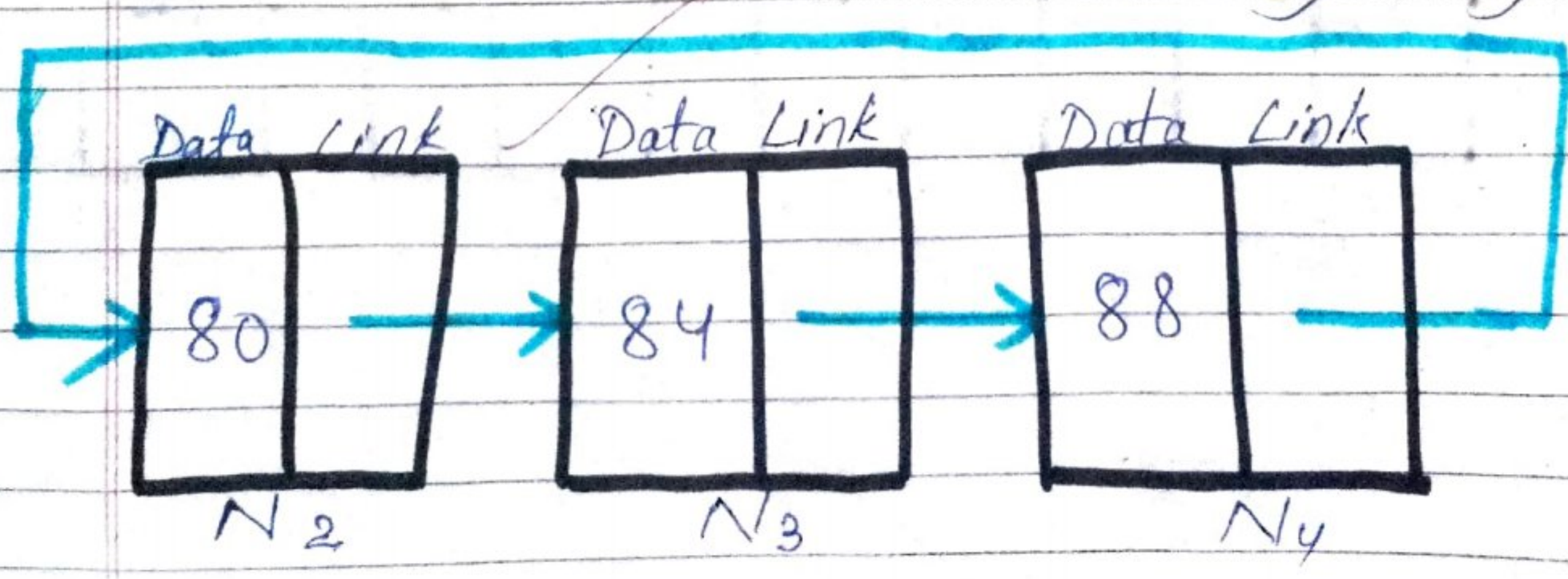
موجودہ Circular Single Linked List



خارج ہونے والے عناصر N1 [Element]



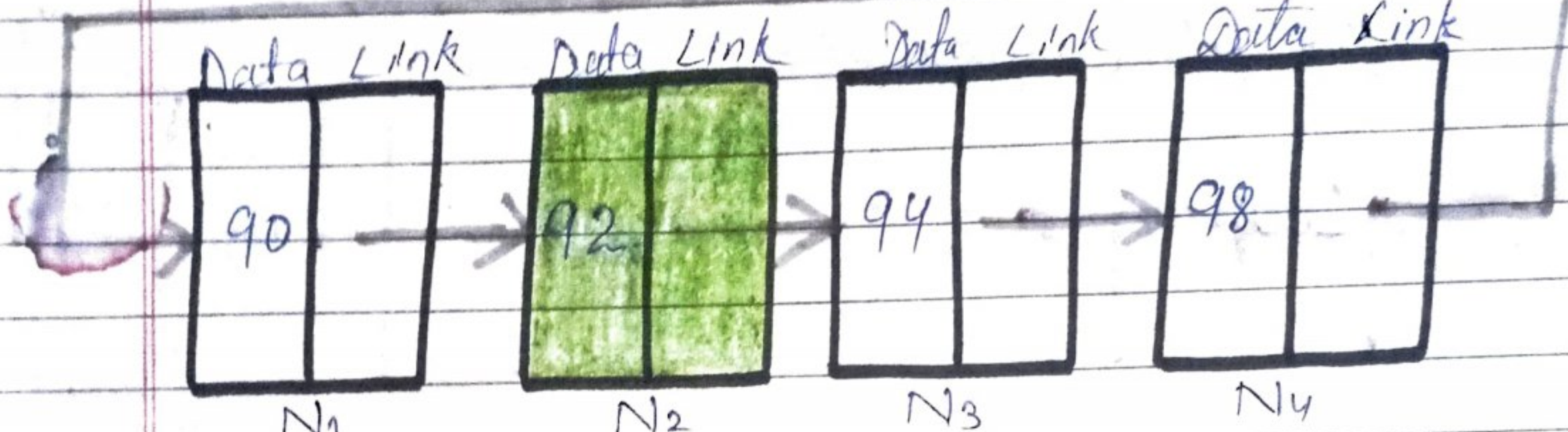
عناصر N1 خارج ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔



Date 1 1
Page

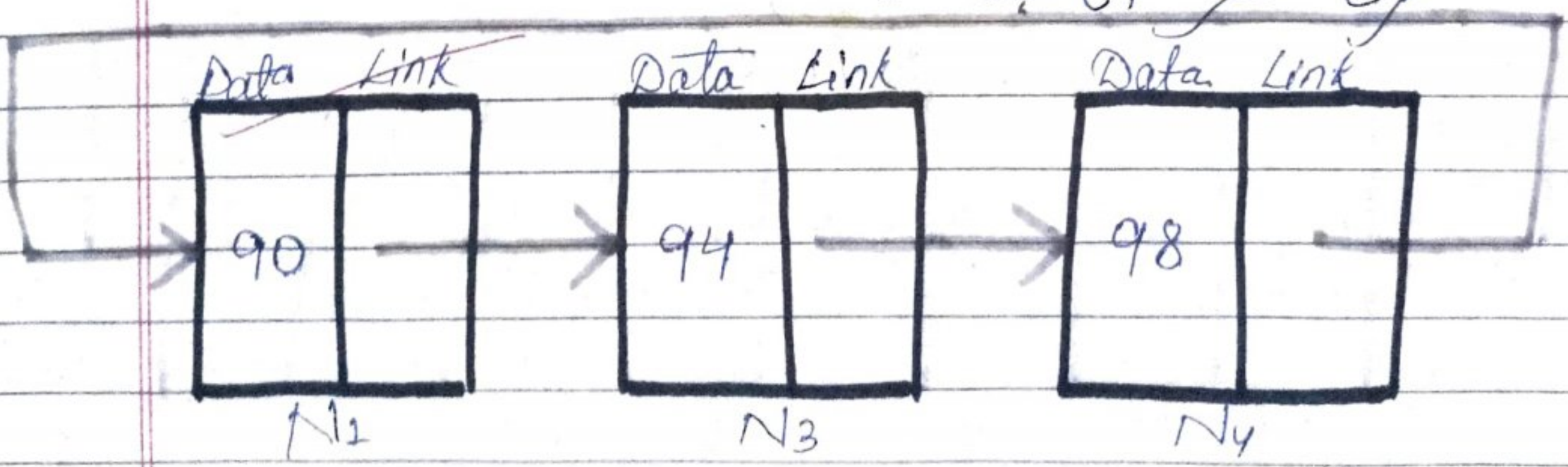
2 Delete at Middle

Circular Single Linked List کے لیے *



* خارج ہونے والے عناصر N2 [Element]

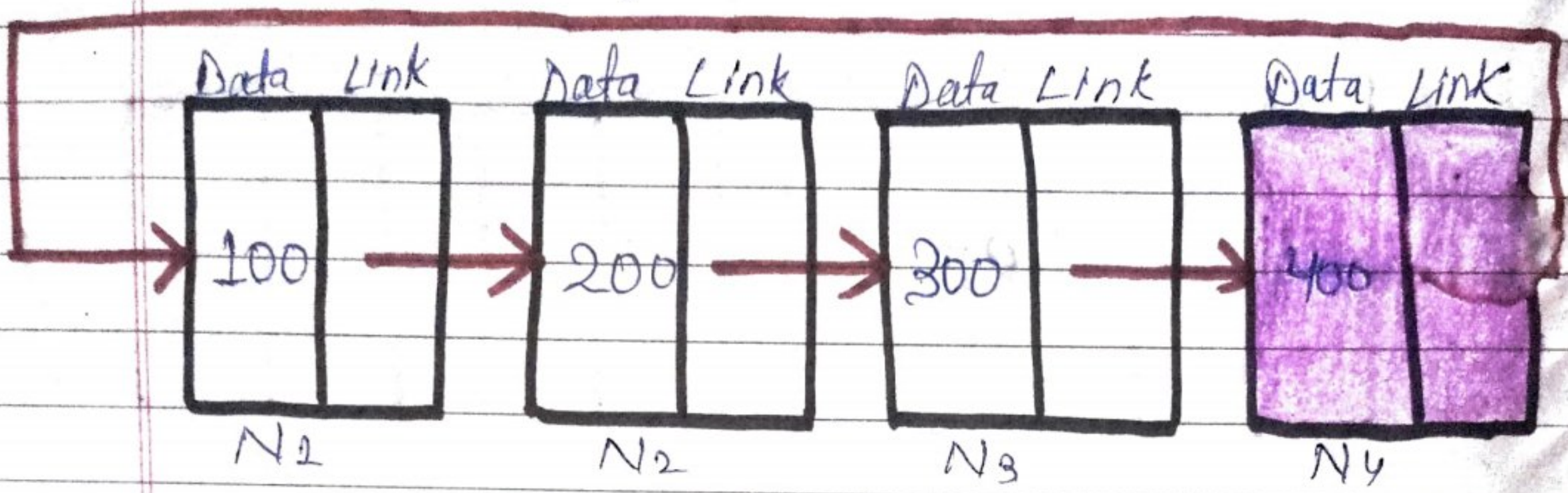
* عناصر N2 خارج ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔



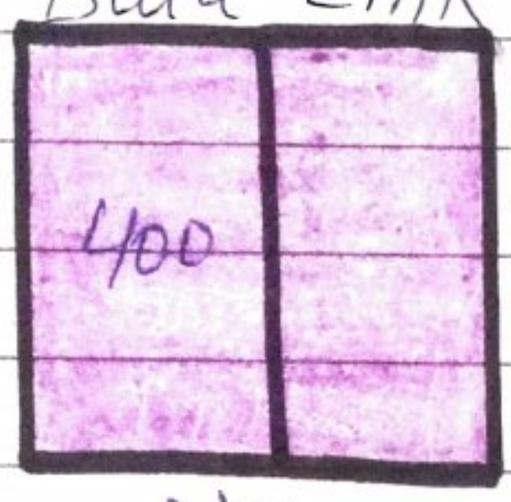
Date / /
Page

3. Delete at Last

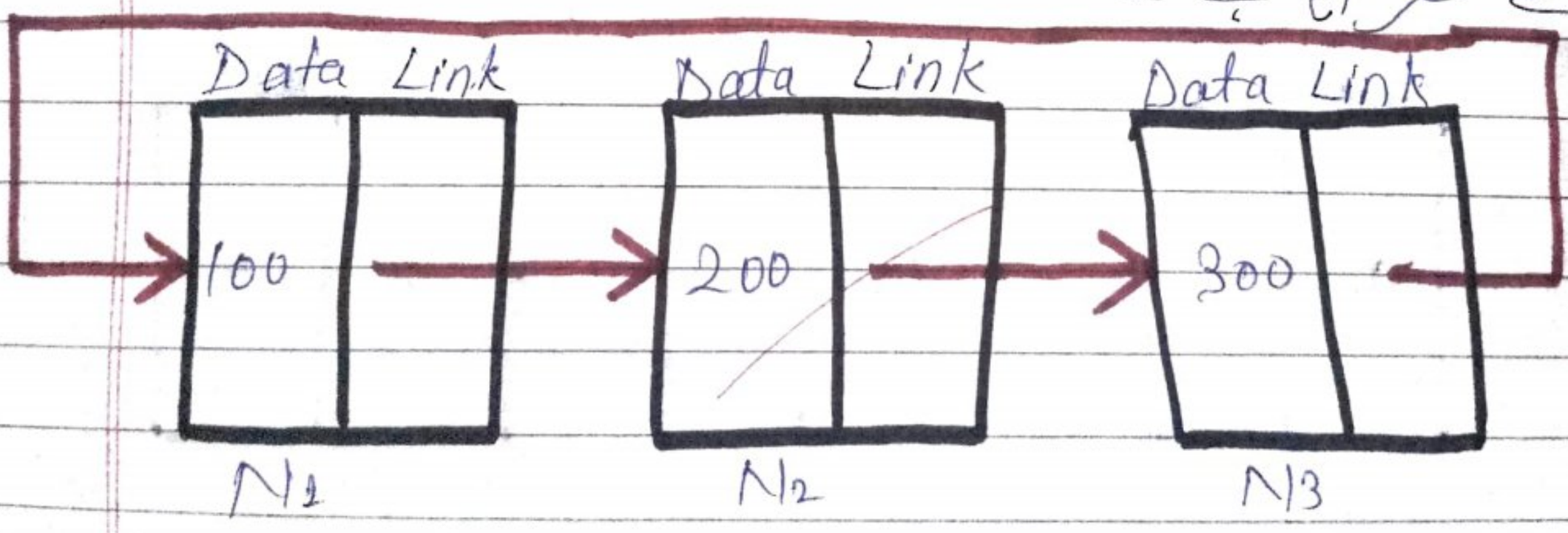
Circular Single Linked List میں سے #



خارج ہونے والے عناصر N4 [Element]



عناصر N4 خارج ہونے کے بعد List کچھ اس طرح سے نظر آتا ہے۔



Date / /
Page

3 Double Linked List

یہ ایک Linear data structure ہوتا ہیں اس میں
تین part ہوتا ہے Left اور part Right میں
part ہوتا ہے اور ایک data میں part ہوتا ہے NULL
اس میں ہم Backward اور forward دونوں ہوتا
ہے۔

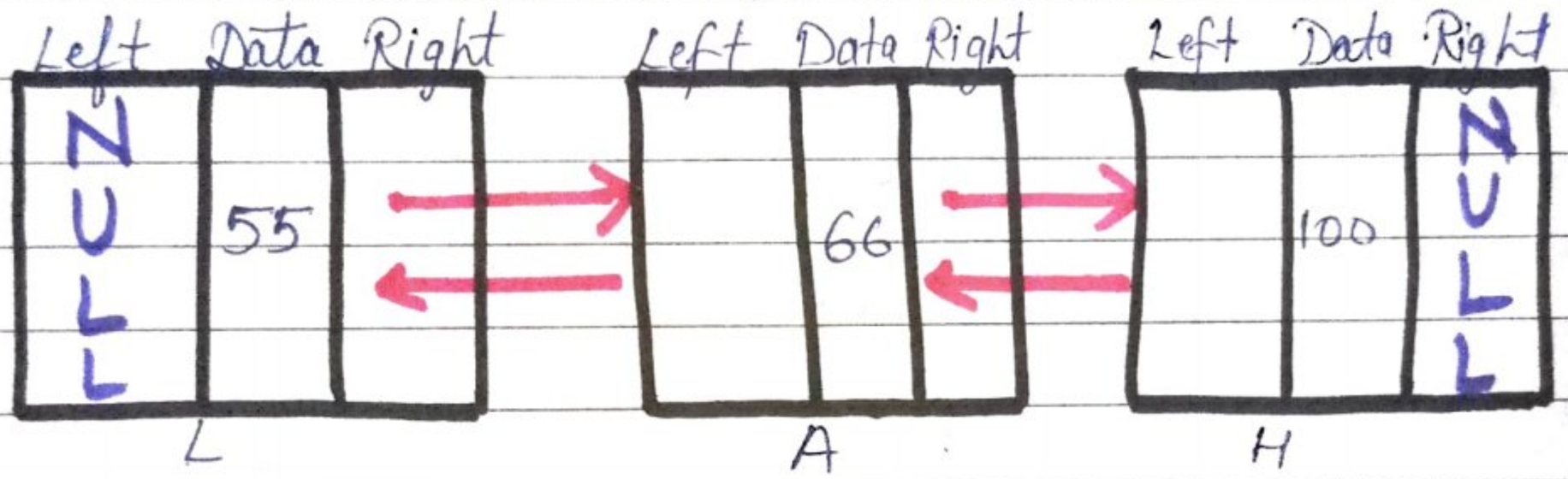
Date / /
Page

Insertation of Element

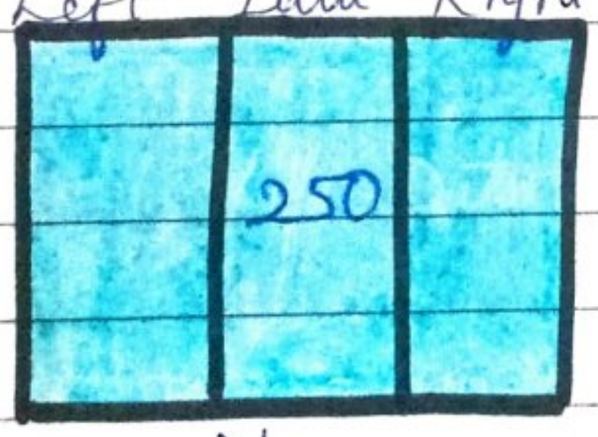
1. Insert at first
2. Insert at Middle
3. Insert at Last

1. Insert at first

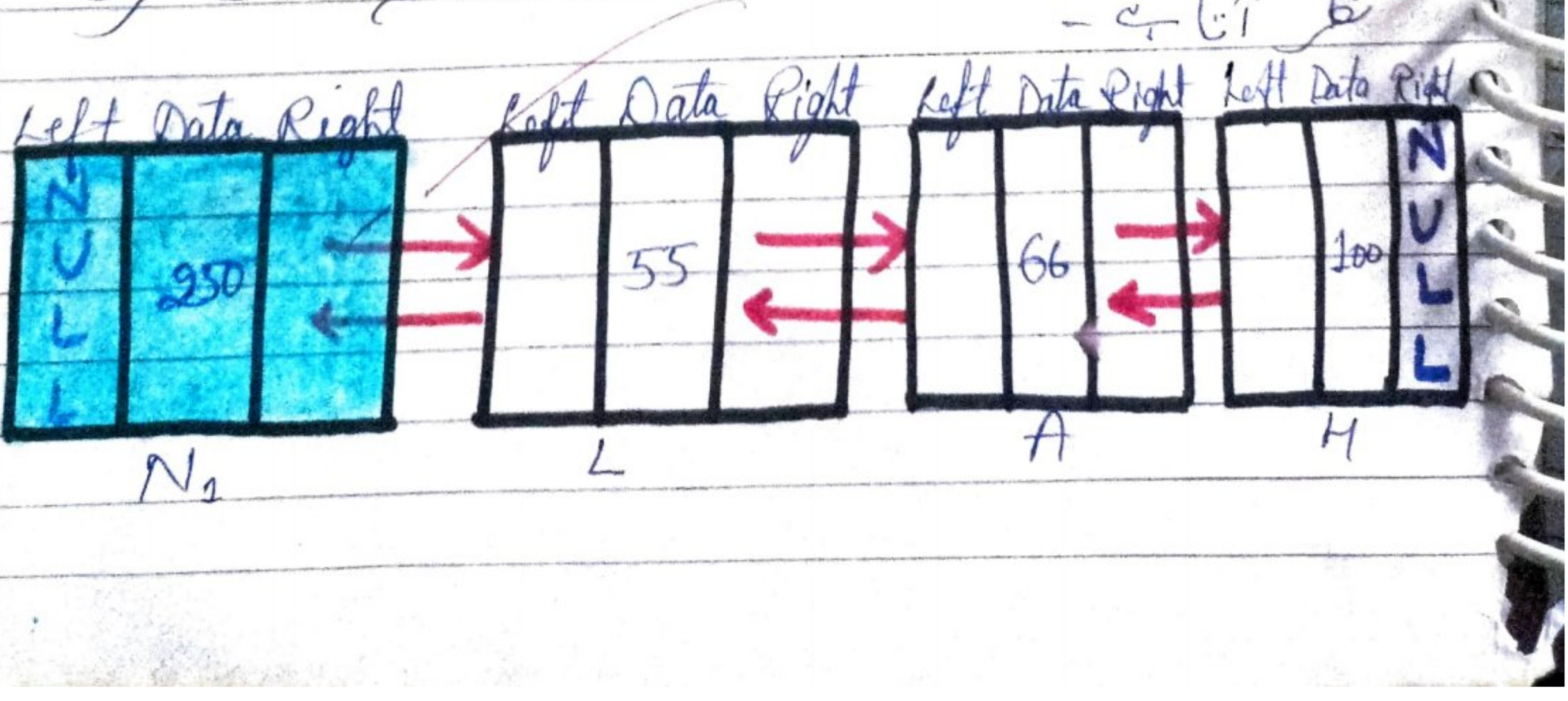
Double Linked List



+ داخل ہونے والا عناصر N1 [Element]



+ عناصر N1 داخل ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔

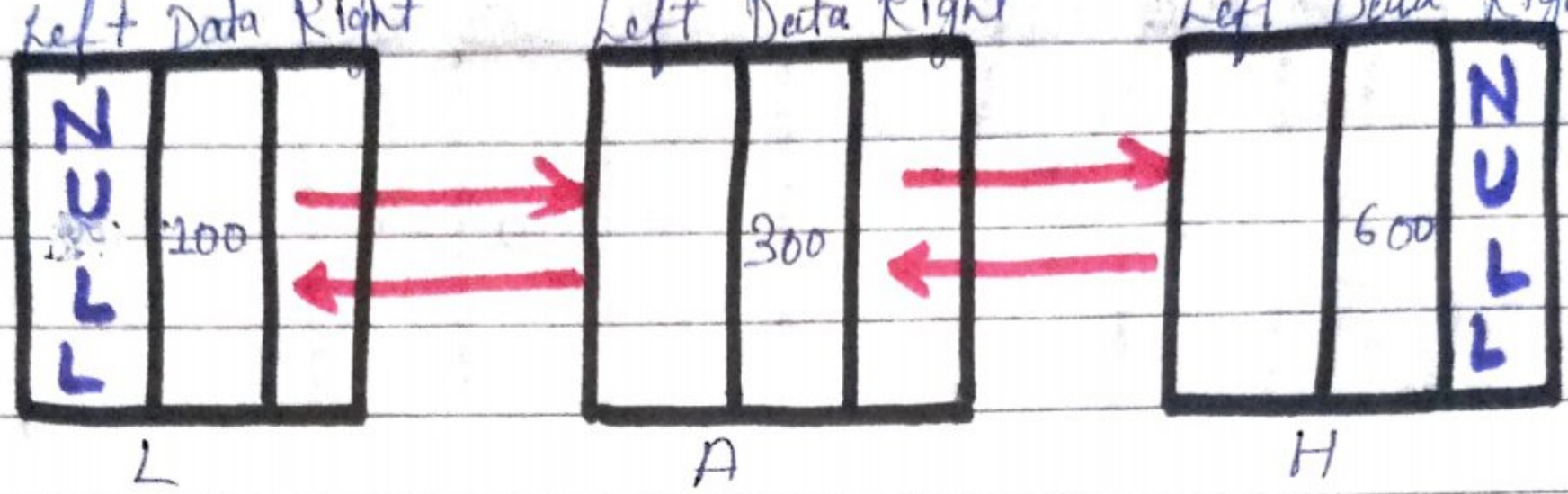


Date / /
Page

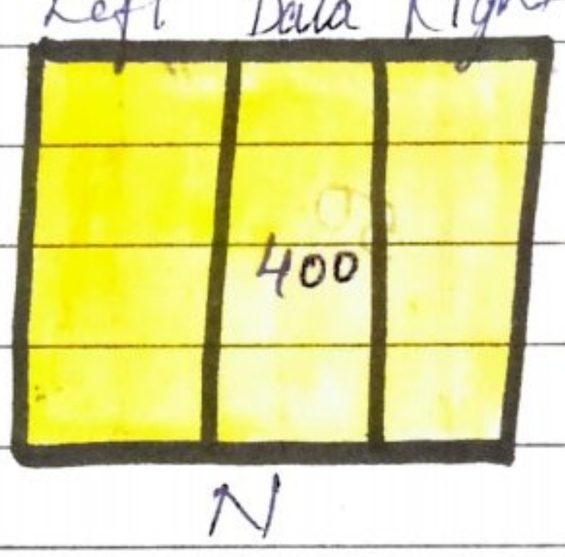
:

2. Insert at Middle

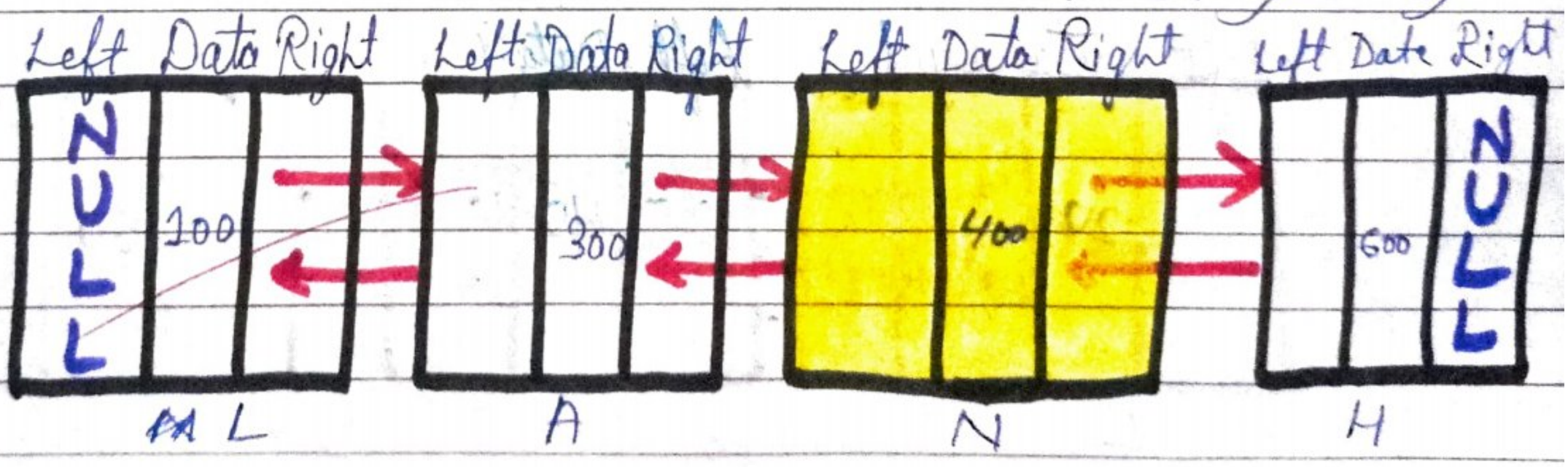
Double List List ہو جو کہ #



N [Element] عناصر والے داخل ہونے #



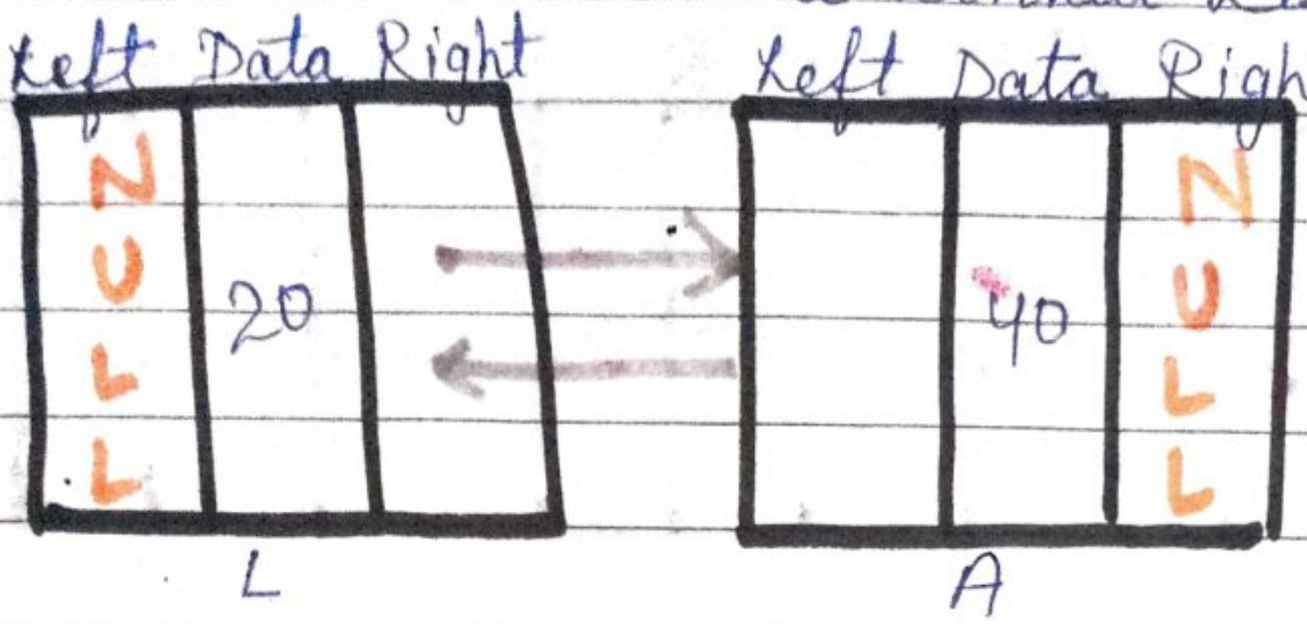
N عناصر داخل ہونے کے بعد List کچھ اس # طرح نظر آتا ہے۔



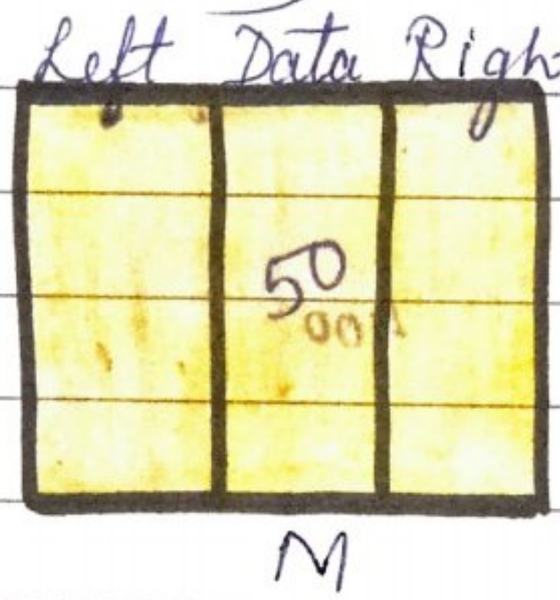
Date / /
Page

3. Insert at Last

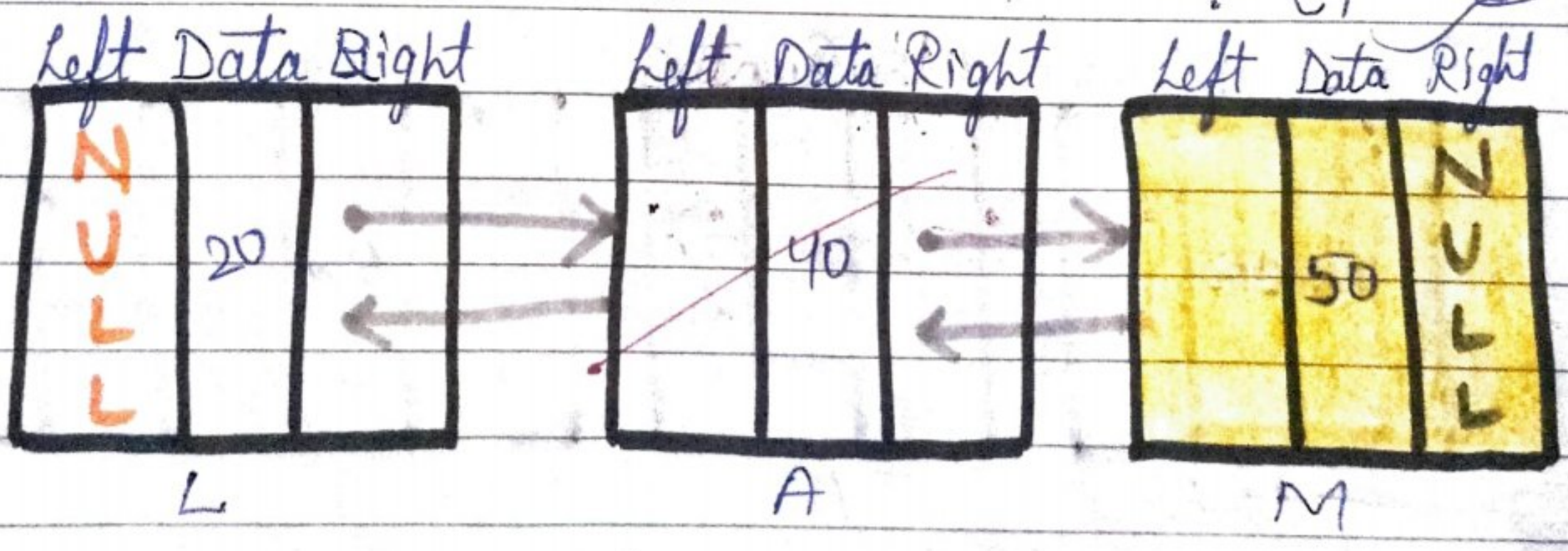
Double Linked List کے آخر میں # نیا node داخل کرنا



داخل ہونے والے عناصر M [Element]



عنصر M داخل ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔



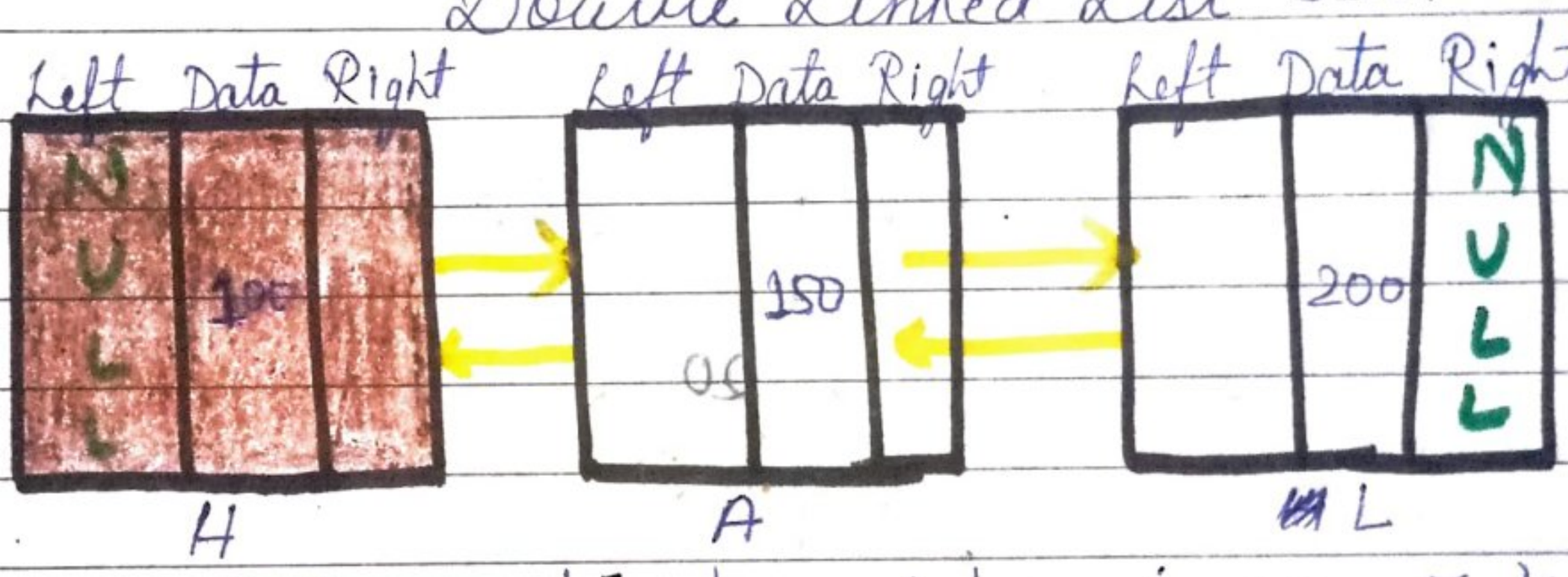
Date / /
Page

Deleting the Elements

1. Delete at first
2. Delete at Middle
3. Delete at Last

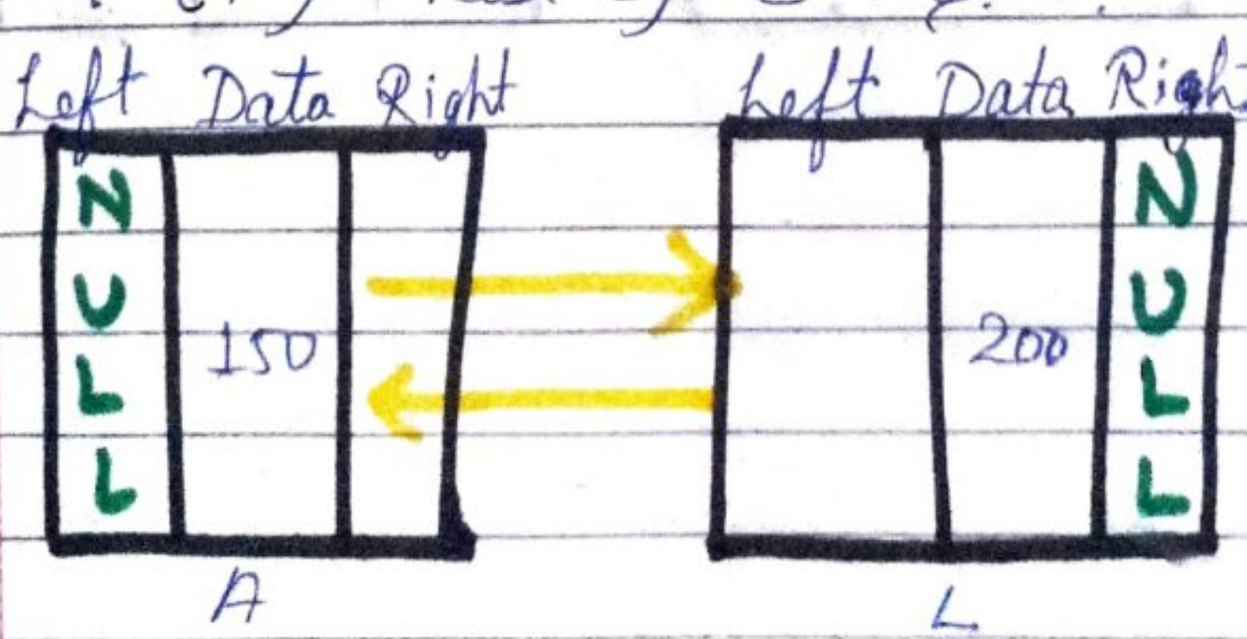
1. Delete at first

Double Linked List کو دیکھتے ہوئے



#خارج ہونے والے عناصر H [Element]

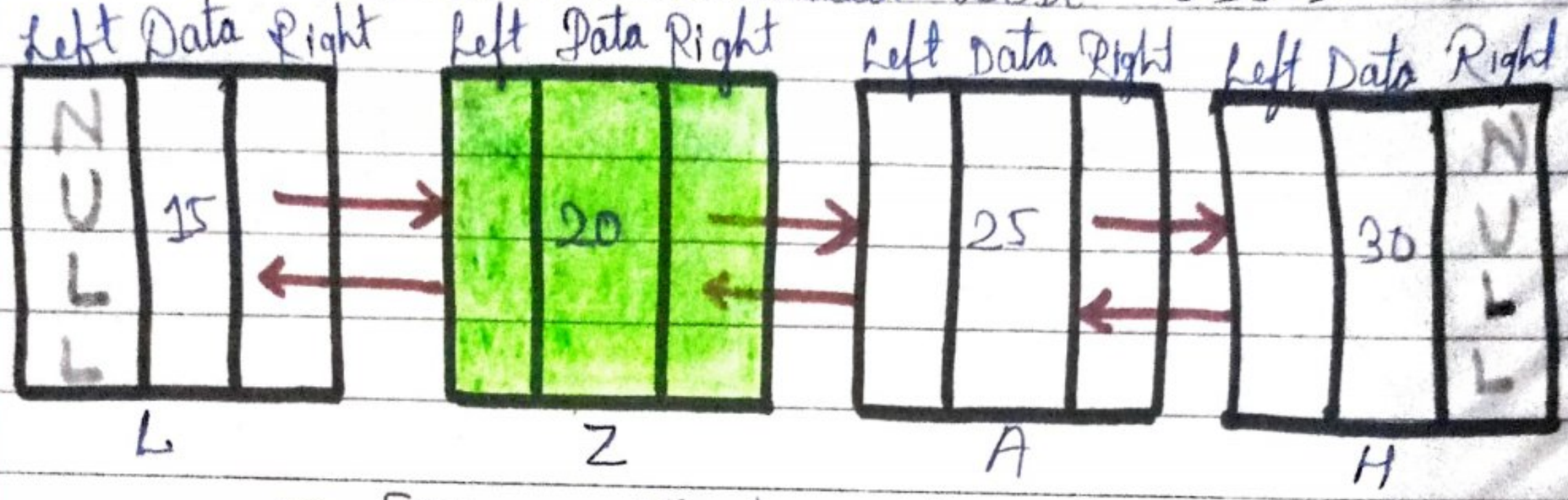
#عناصر H خارج ہونے کے بعد یہ اس طرح List نظر آتا ہے۔



Date / /
Page

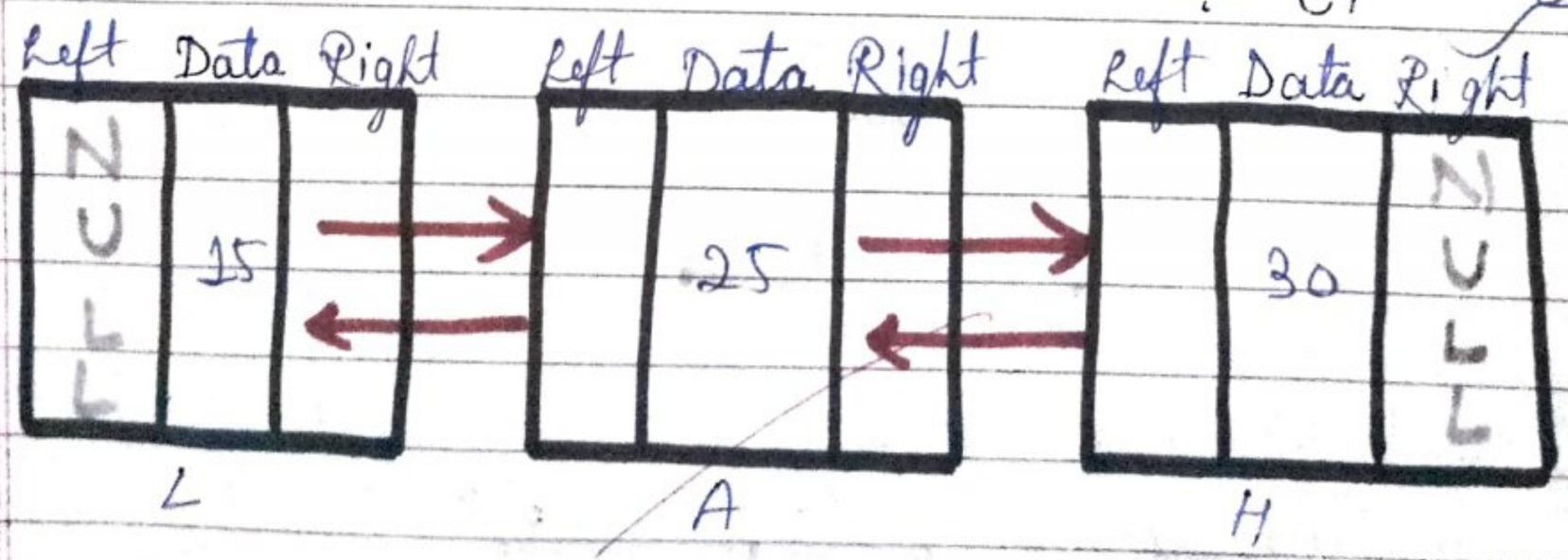
2. Delete at Middle

Double Linked List کو دیکھیں #



Z [Element] عناصر والا خارج ہونے #

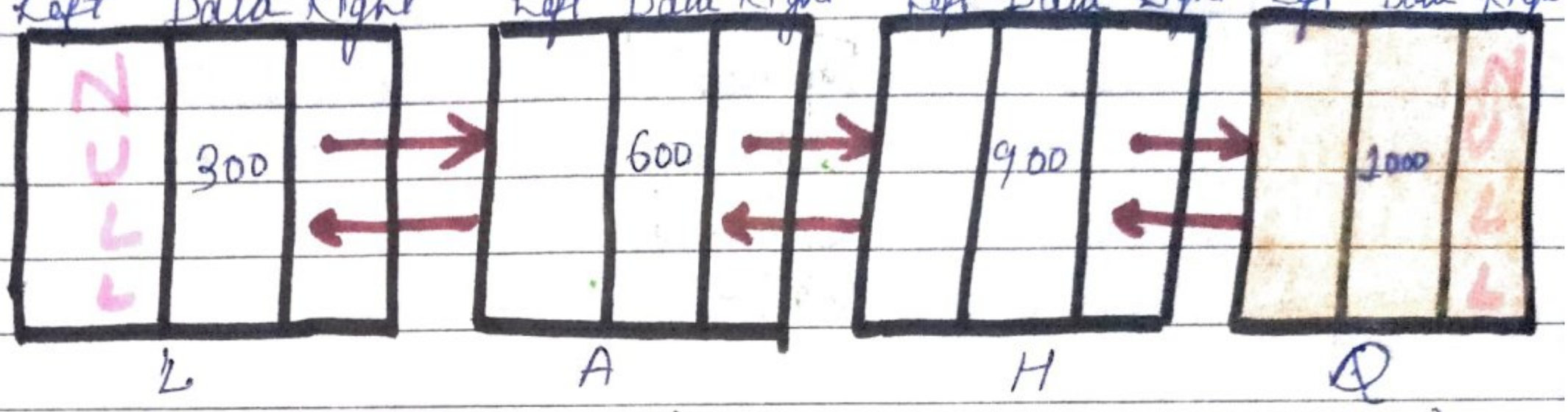
+ عناصر Z خارج ہونے بعد List کچھ اس طرح نظر آتا ہے -



Date / /
Page

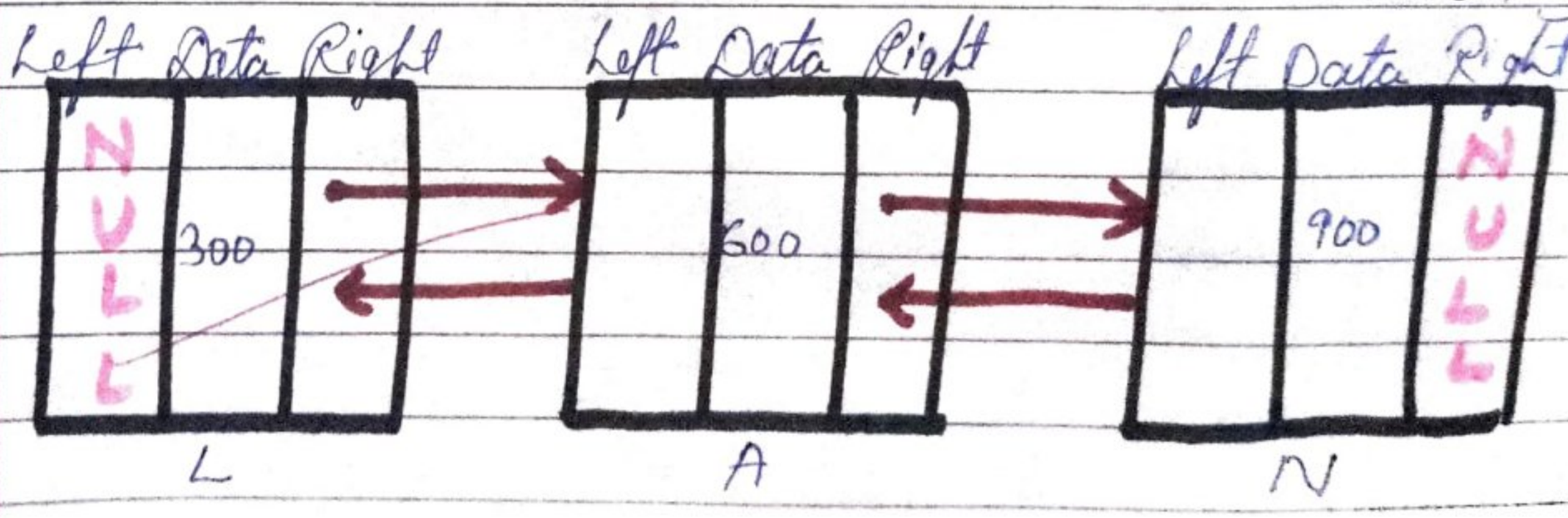
3. Delete at Last

Double Linked List میں #



Q [Element] عناصر والا ہونے خارج #

Q عناصر # خارج ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔



Date / /
Page

4 Circular Double Linked List

یہ ایک Linear data structure ہوتا ہے
اس میں تین Part ہوتا ہے Left اور Right
Part میں ایک Part اور ایک Link ہوتا ہے
right والے Last کے اس کے - ہوتا ہے data
Left k node والے first سے part والا
والا first جڑا ہوتا ہے - اور part
k node والے Last part والا left k node
سے جڑا ہوتا ہے - side right

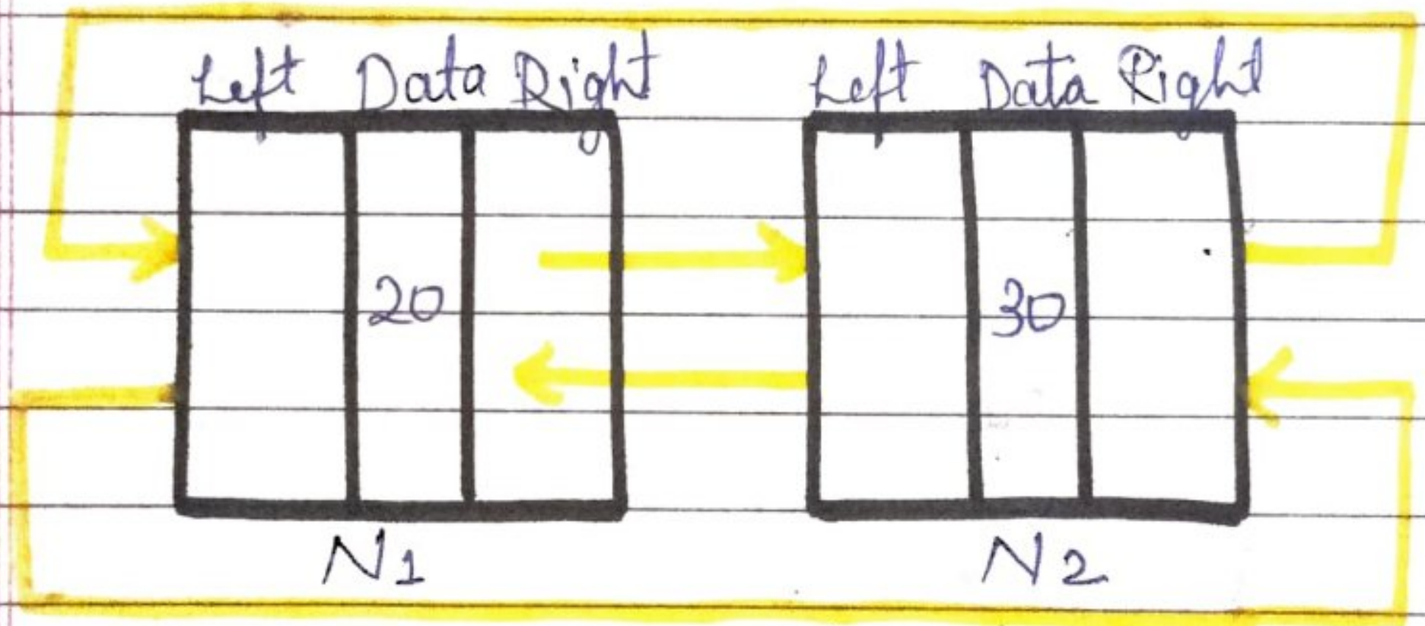
Date / /
Page

Insertation of Element

1. Insert at first
2. Insert at Middle.
3. Insert at Last.

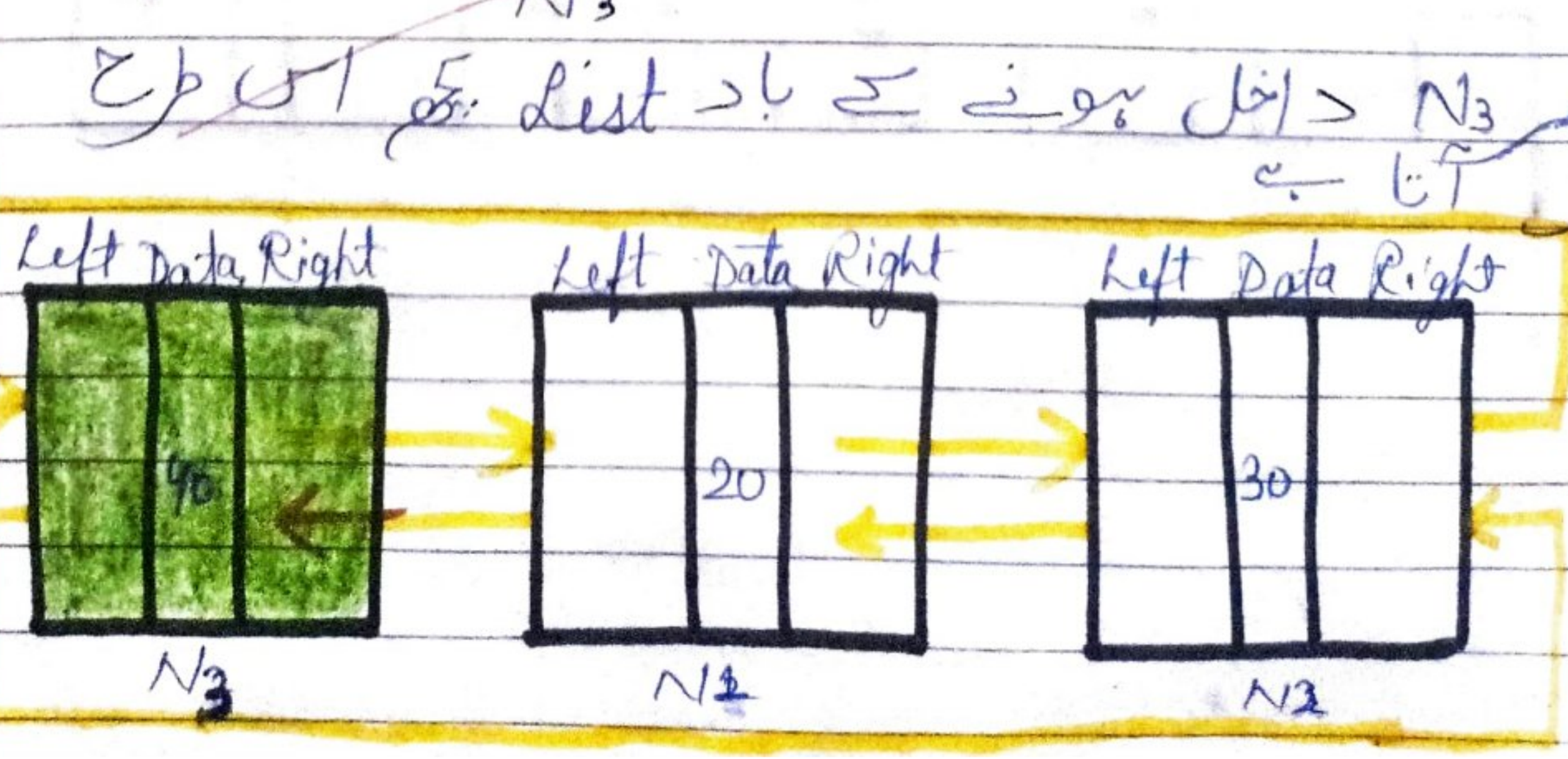
1 Insert at first

Circular Double Linked List ہے #



N3 [Element] میں داخل ہونے والے عناصر #

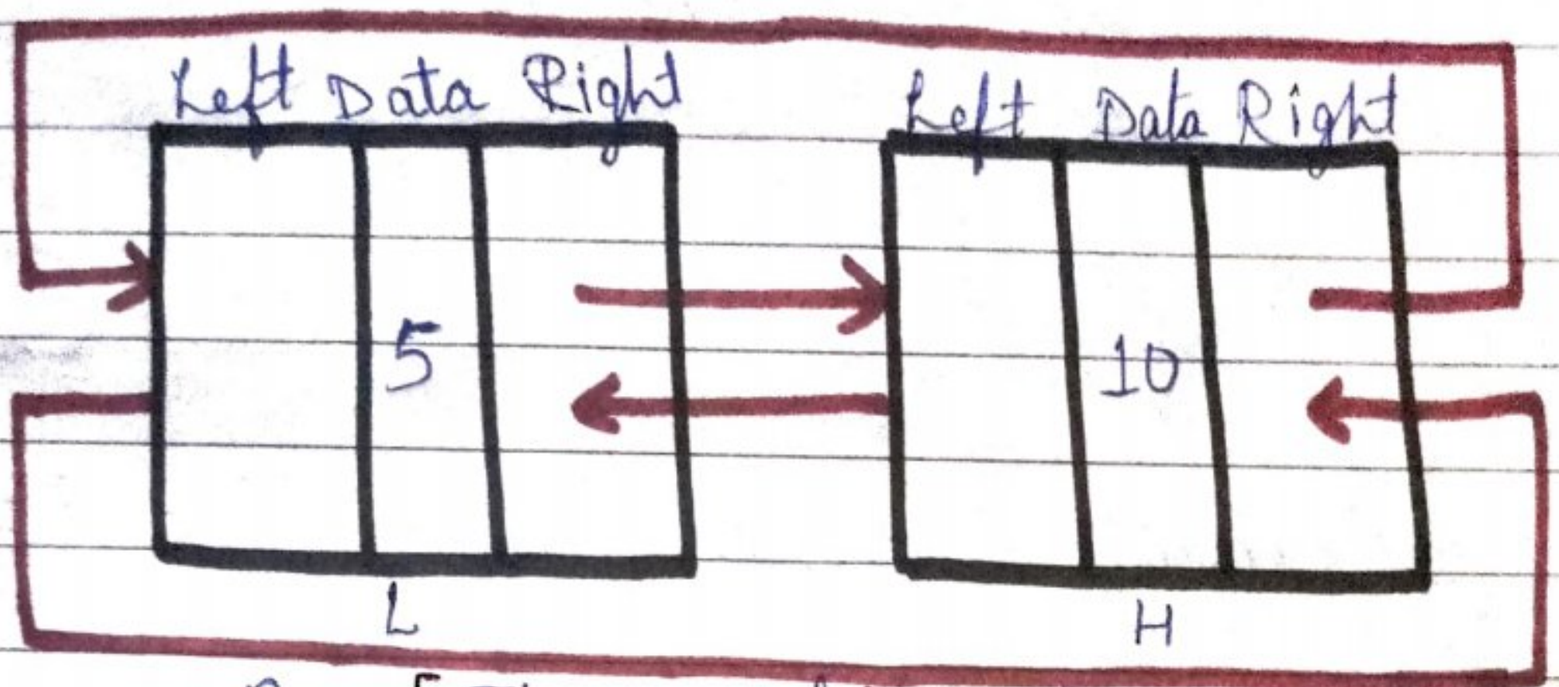
List میں داخل ہونے کے بعد N3 عناصر اس طرح نظر آتا ہے #



Date ___/___/___
Page _____

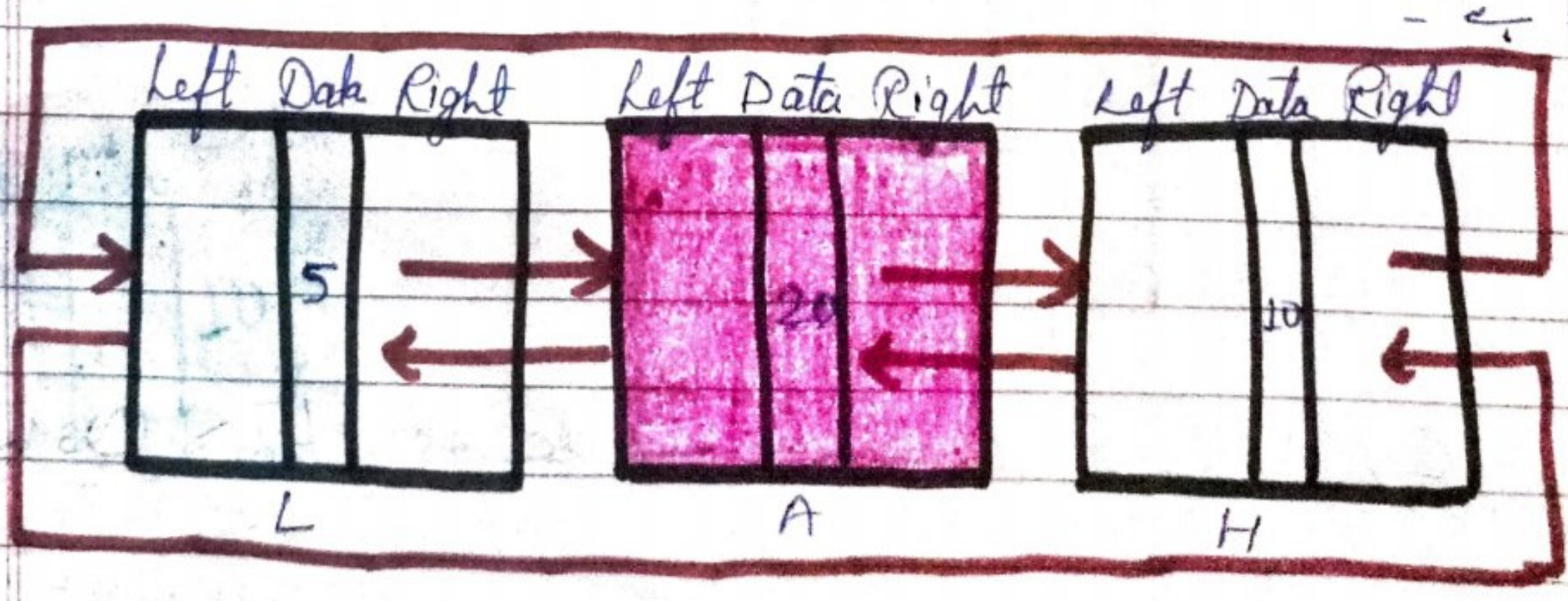
2. Insert at Middle

Circular Double Linked List کی طرح +



A [Element] # داخل ہونے والے عناصر

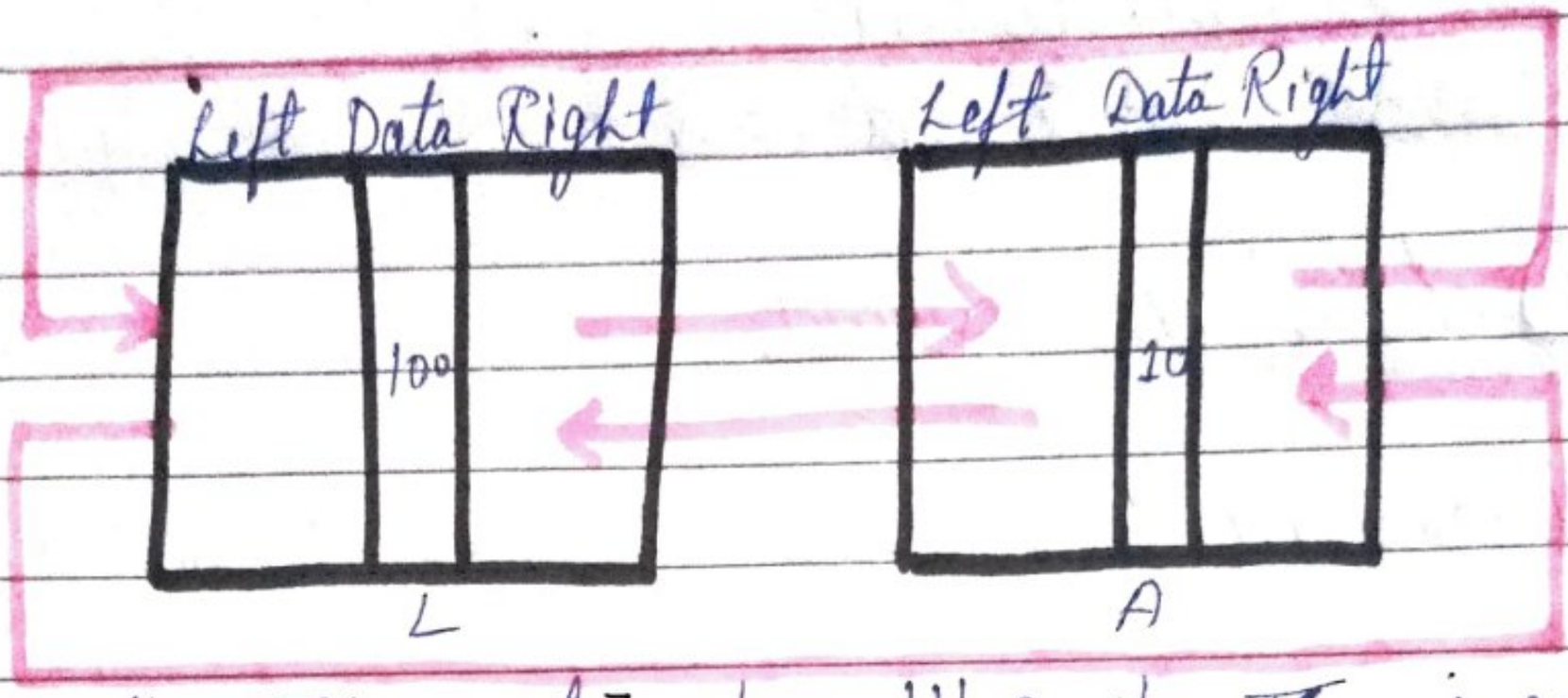
عناصر A داخل ہونے کے بعد Link List کچھ اس طرح نظر آتا ہے۔



Date / /
Page

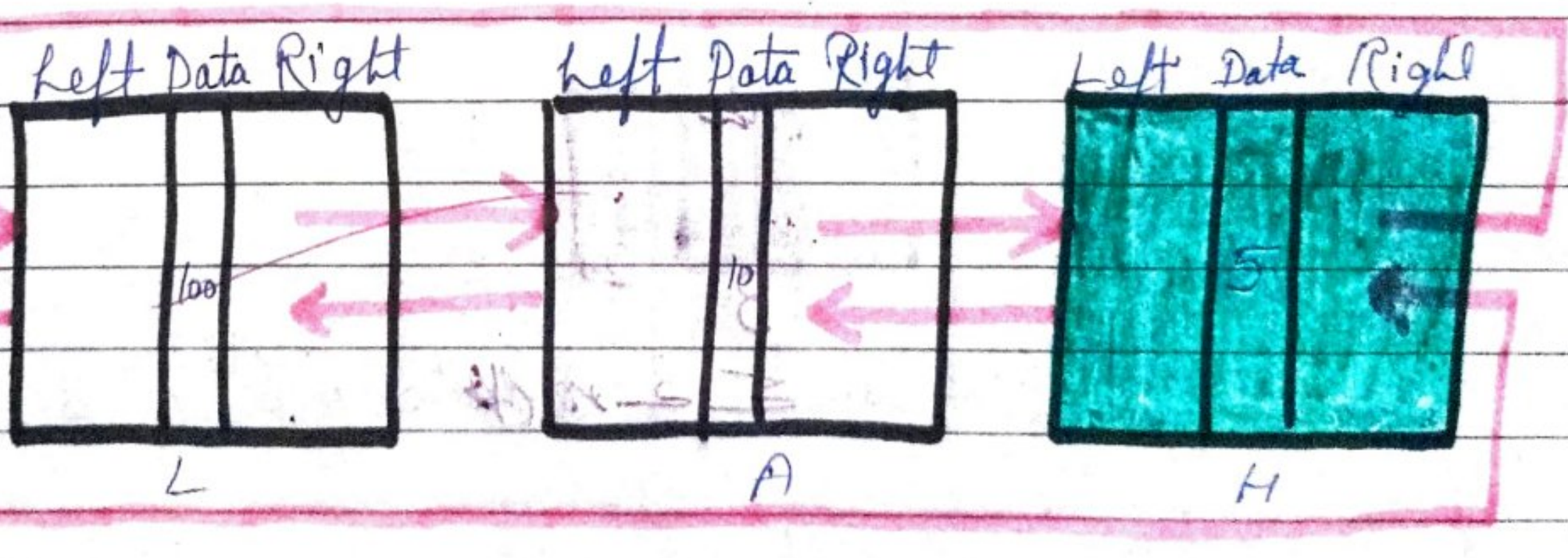
3. Insert at Last

Circular Double Linked List میں موجود



داخل ہونے والا عناصر H [Element]

عناصر H داخل ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔



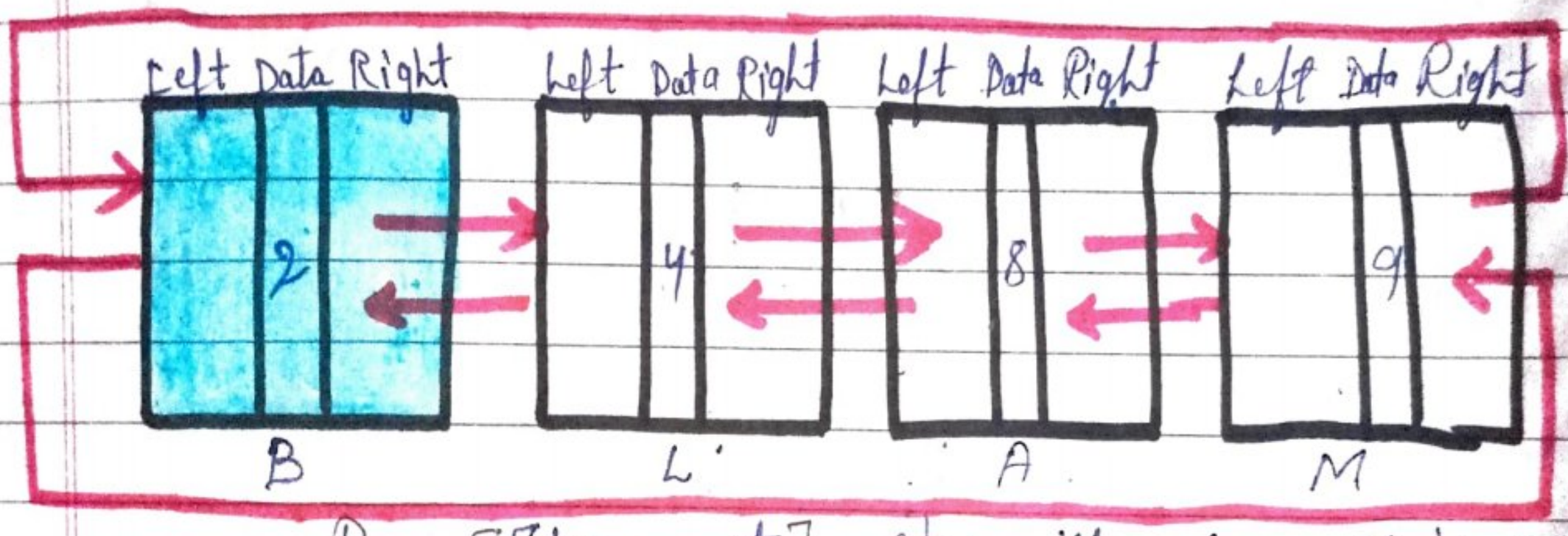
Date / /
Page

Deleting Of Element

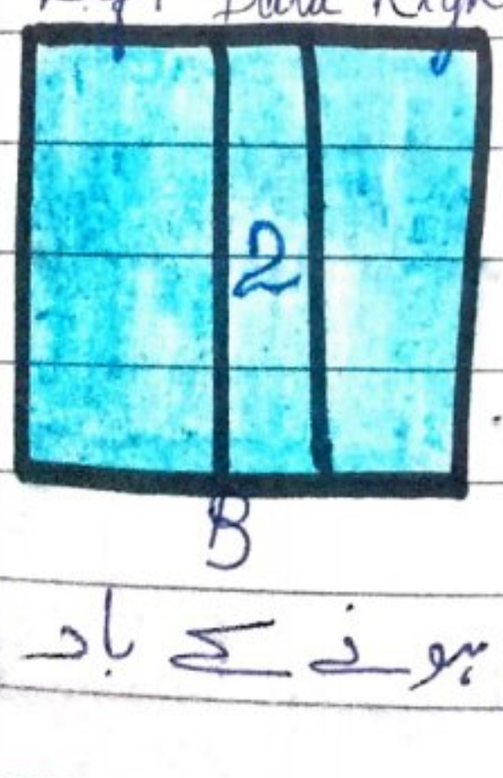
1. Delete at first
2. Delete at Middle
3. Delete at Last

1. Delete at first

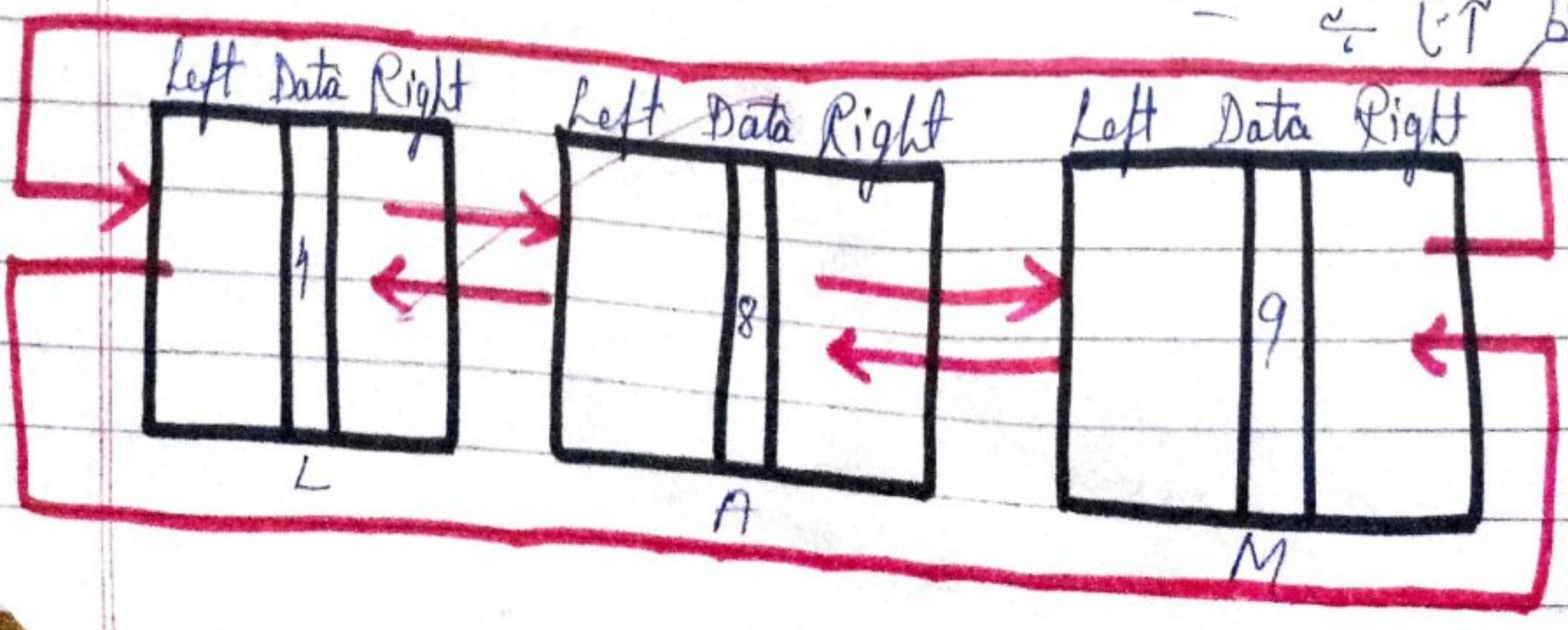
Circular Double Linked List



B [Element] عناصر والا خارج ہونے +



B عناصر خارج ہونے کے بعد List کا حال +

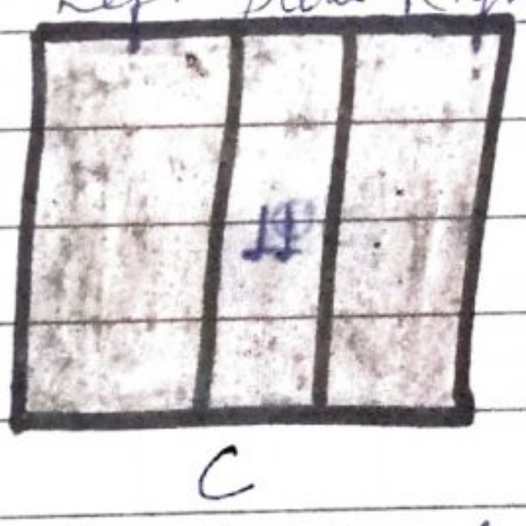


Date ___/___/___
Page ___

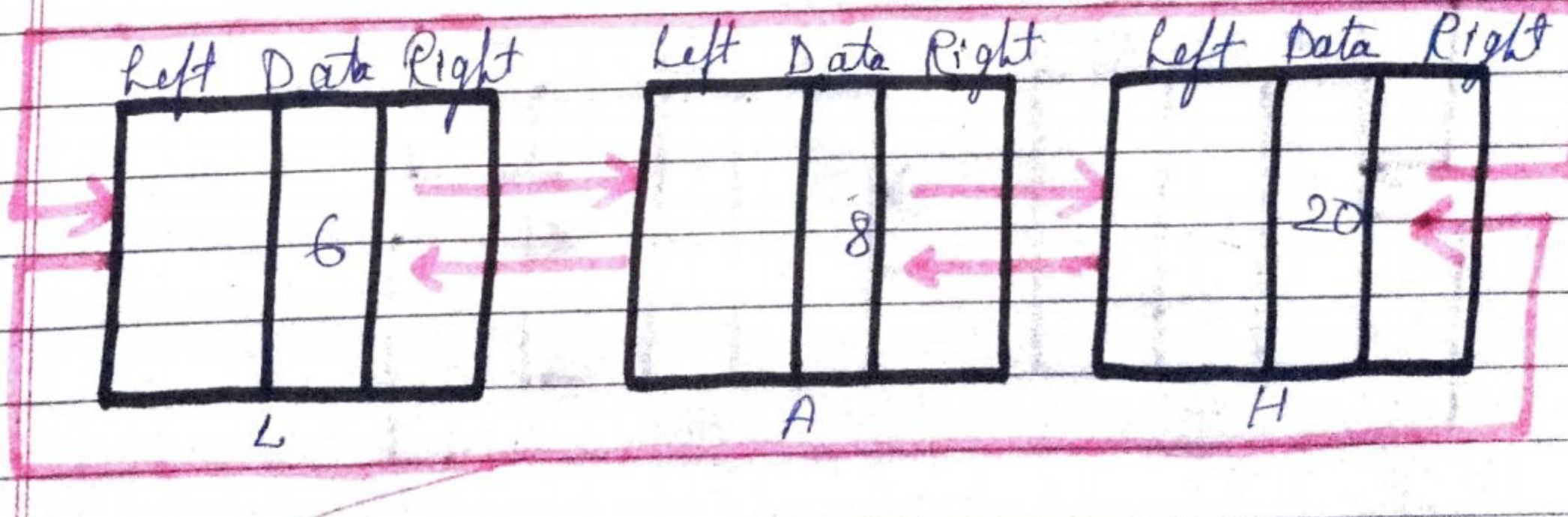
2. Delete at Middle

Circular Double Linked List ⇦ ⇨ #

C [Element] عناصر والا خارج ہونے #



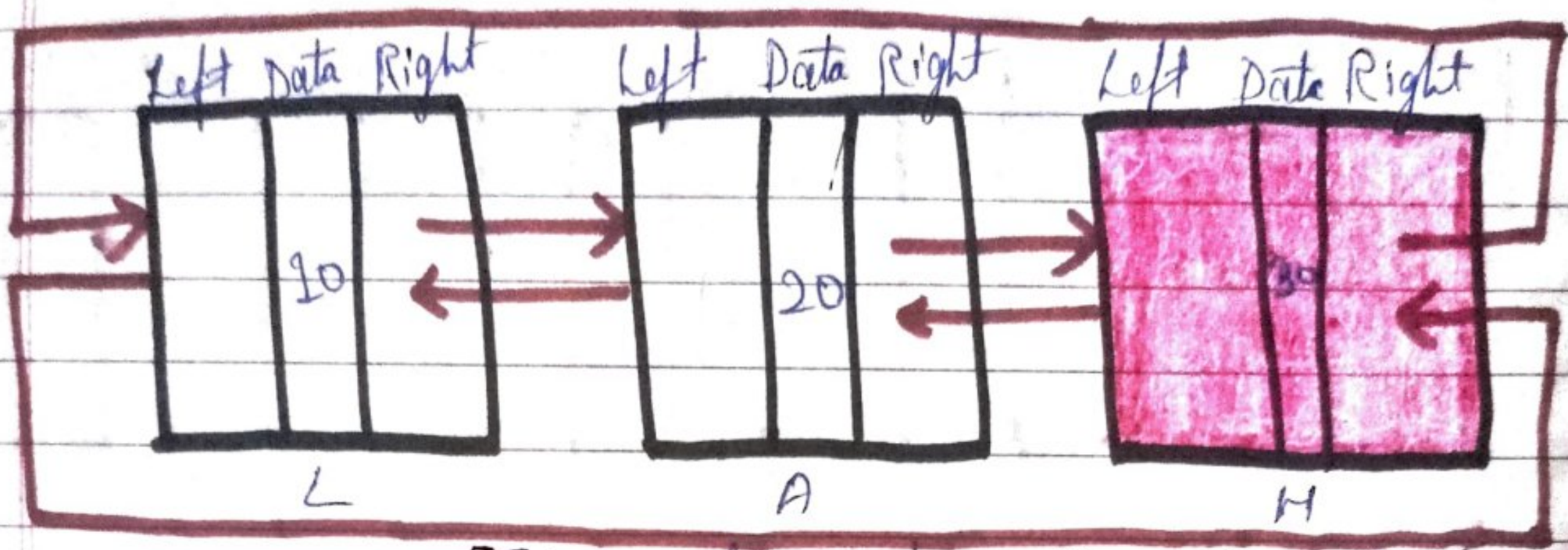
List سے اس طرح نظر آتا ہے۔ #عناصر خارج ہونے کے بعد



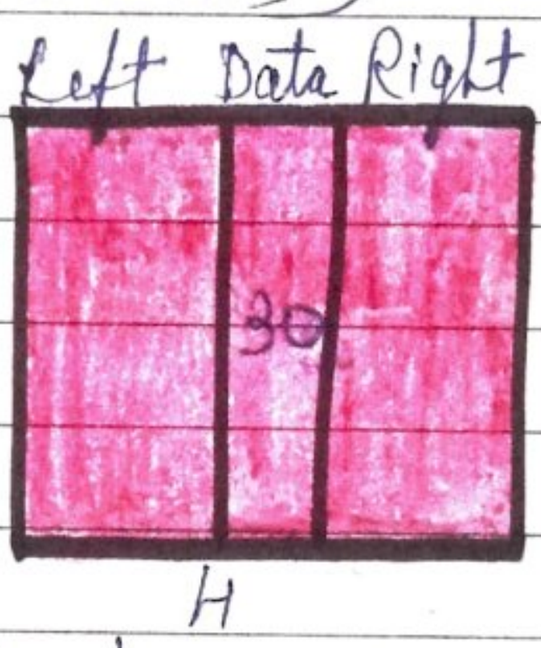
Date / /
Page

3. Delete at Last

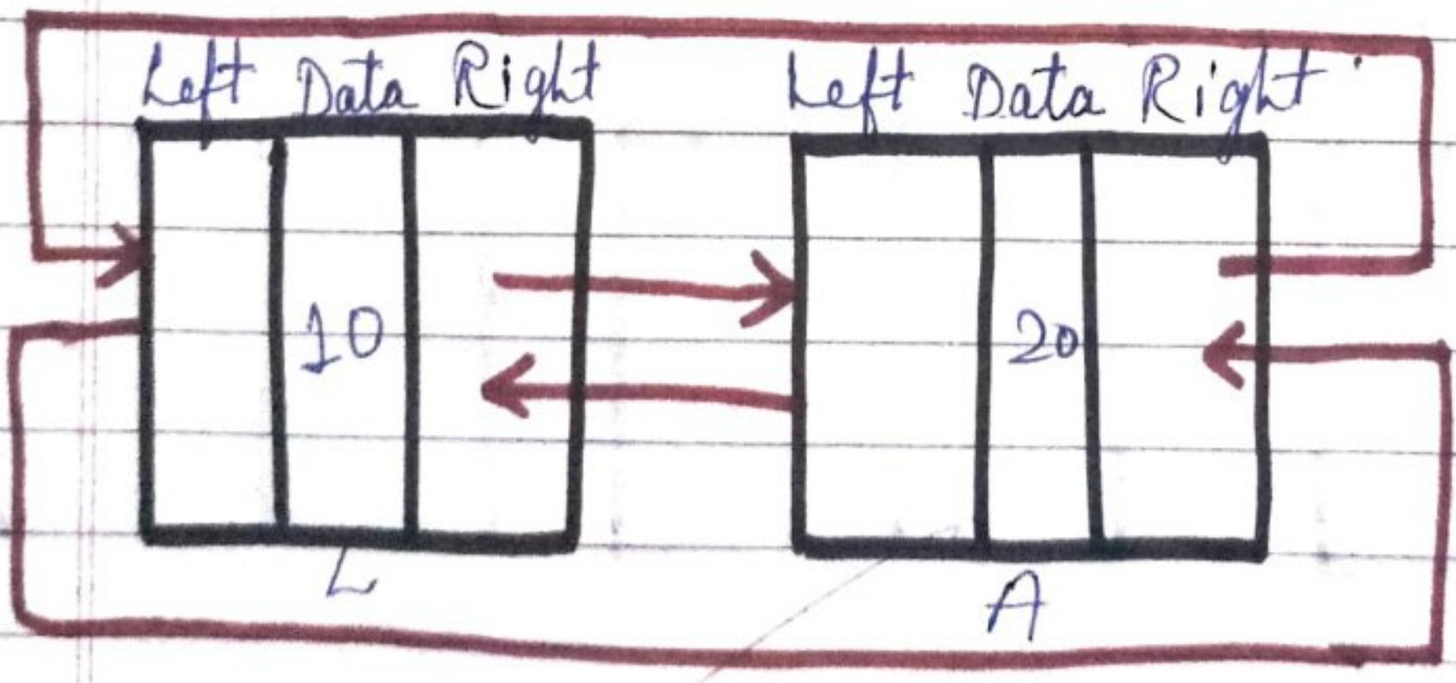
Circular Doubele Linked List ہے۔



خارج ہونے والا عناصر [Element] H



عناصر H خارج ہونے کے بعد List کچھ اس طرح نظر آتا ہے۔



Difference Between Single Linked List And Double Linked List

Single Linked List

* اس میں ہم Backward نہیں جا سکتے ہے -
* اس میں Last والے node میں Link کی NULL ہوتا ہے -
* اس میں link Single ہوتا ہے -
*

Date / /
Page

Double Linked List

اس میں ہم Backward اور forward دونوں ہوتا ہے۔

اس میں موجود node میں تین part ہوتے ہے دو link part کے لیے اور ایک data part کے لیے۔

اس میں last والے node کا right والا part NULL ہوتا ہے اور first والے node میں left کا part NULL ہوتا ہے۔

Date
Page

Difference Between Circular Single Linked List And Circular Double Linked List

Circular Single Linked List

* اس میں ہم Backward جا سکتے ہے۔
* اس کا node میں موجود دو link part اور data جو node والے میں وہ پہلے part والا link part والے data سے جڑا ہوتا ہے۔
* اس کے last والے node میں link part والے NULL نہیں ہوتا ہے۔

Circular Double Linked List

اس میں ہم Backward اور forward دونوں ہوتا ہے۔

اس میں تین part ہوتا ہے دو part link کے لیے اور ایک part data کے لیے۔

اس میں node میں NULL نہیں ہوتا ہے۔ last والے node کے right والے part سے جڑا ہوتا ہے۔

first node کے left والے part سے last node کے right والے part سے جڑا ہوتا ہے۔

11/3/22
C.W

Date / /
Page

Stack And Queue

chapter :- 2

Stack

Introduction to stack, Array representation of static operation on stack, Application of stack (Infix - to - Postfix) Evaluating Postfix Expression.

Queue

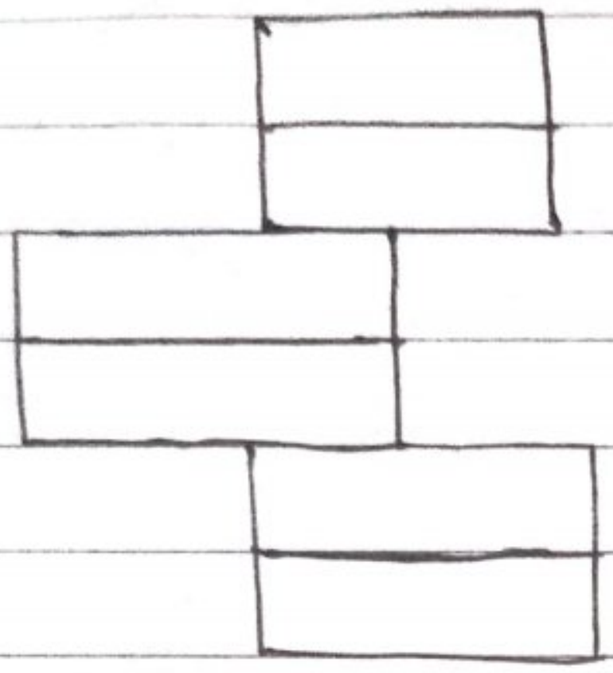
Introduction to Queue, Array representation of Queue operation on Queue. Types of Queue & De Queue, Circular Queue, Application of Queue, Round Robin Algorithm, sparse matrix and its representation.

Overview of stack and Queue

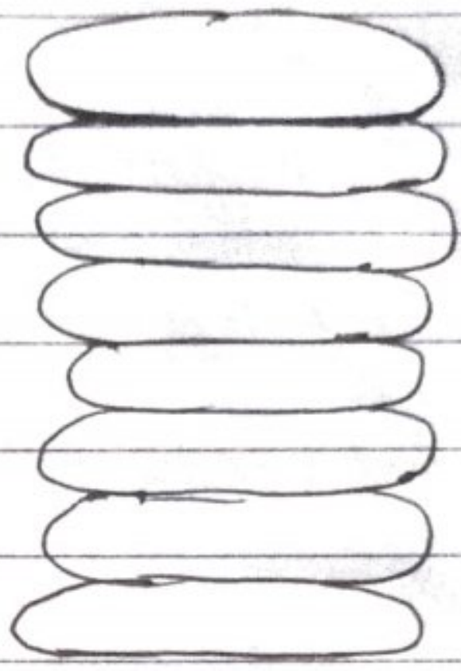
structure data linear structure data دو ہے
ہیں جو عام اور بہت زادا استعمال کیا جاتا
example کا ہے۔ یہ دونوں structure data کو Array
ذریعے عمل action میں لایا جاسکتا ہے۔
type data defined Pre ہوتا ہیں جو عام Array
طور پر ہر programming language میں موجود ہوتا
ہیں جسکو استعمال کرنا بہت easy ہوتا ہے۔
Stack ایک structure data Linear ہوتا ہیں جس
میں خارج اور داخل کا عناصر جو کی ایک ہی جگہ سے
ہوتا ہیں جس کو "Top" یا "Top of the stack"
بھی کہتے ہے۔

[Last in first out] LIFO کو Stack عام طور پر
کہتے ہیں name سے

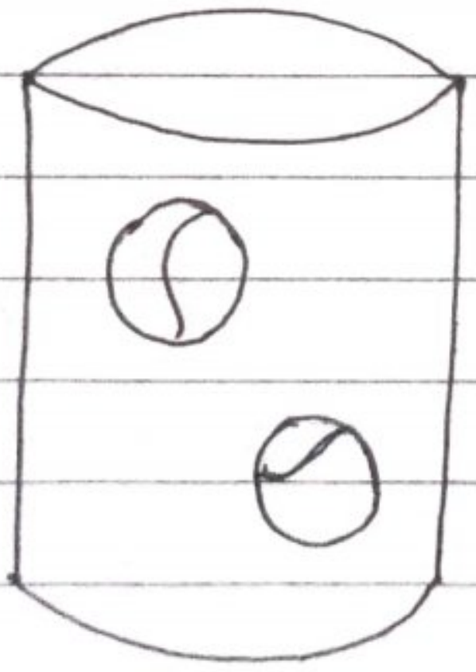
Real Time Example of Stack



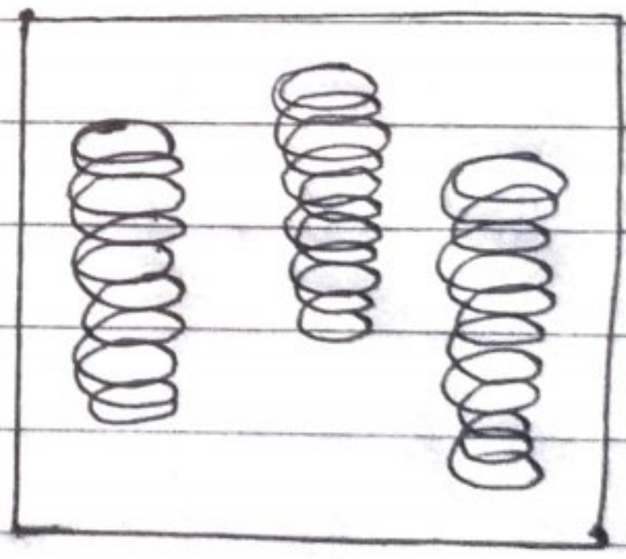
Stack BOX



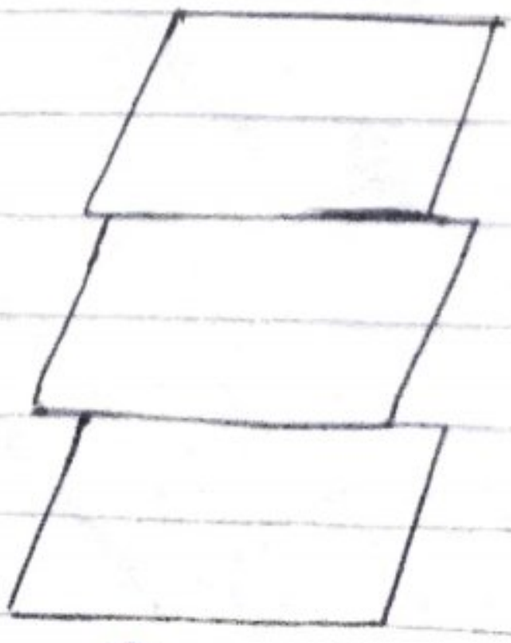
Plates



Ball



Coins.



Books

1. Stack of Books
2. Stack of Bill Papers
3. Stack of Dishes
4. Stack of Coins
5. Stack of Folding clothes.

Operation of Stack

1. Inserting an Element [عناصر کو داخل کرنا]
2. Deleting an Element - [عناصر کو خارج کرنا]

Stack کی زبان میں Insertion [یا Push] اور Deletion کو [یا POP] کو ایک نئی تکنیک [new technique] کا نام دیا گیا ہے -

stack میں کوئی بھی operation کرنے کے لیے "stack" کی مدد لی جاتی ہیں جسکو "TOP" بھی کہتے ہے "Top of the" -

Algorithm

It is a step by step process to solve any problem is called Algorithm.

Algorithm کا PUSH

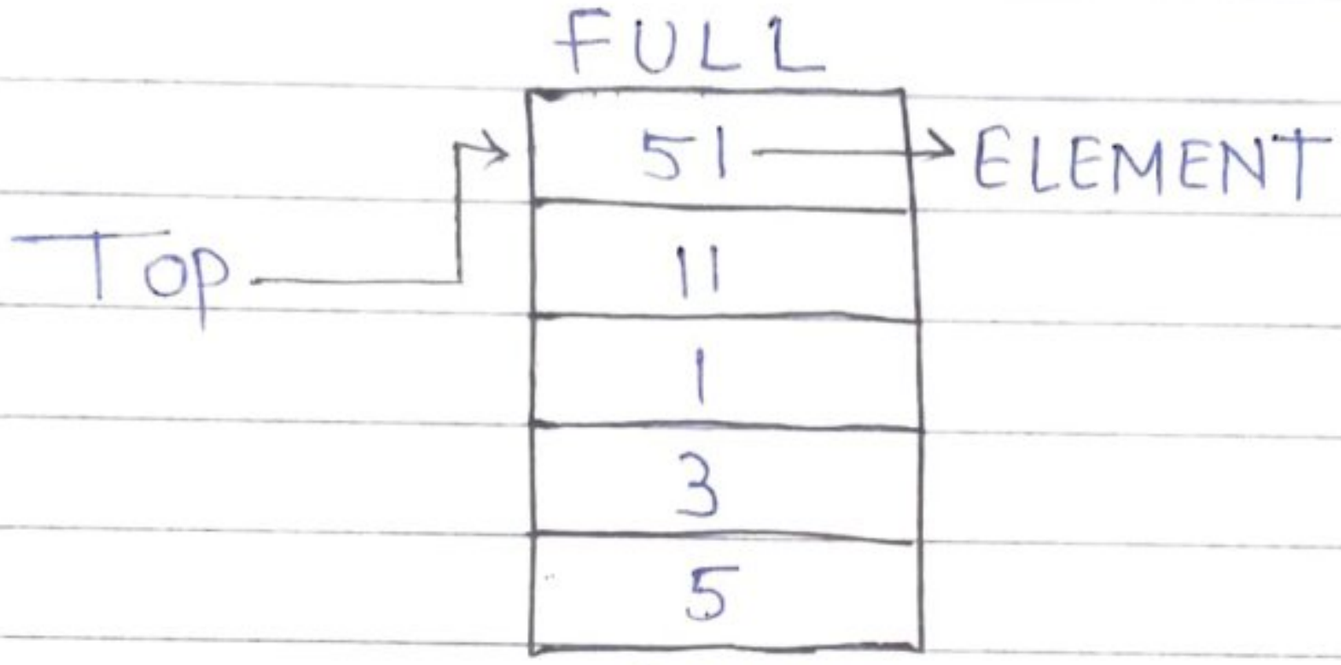
stack میں PUSH کے عمل سے عناصر کو داخل کیا جاتا ہے - اس کو عمل میں لانے کے لیے سب سے پہلے ہم stack کی جانچ کرتے ہیں اگر stack "Full" ہو گی تب ہم screen پر user کو پیغام message ظاہر کرینگے -

For Example :- "STACK OVERFLOW"

اگر Stack FULL نہیں ہیں تب اس میں PUSH والا عمل ہوگا جسکو ایک Algorithm کے ذریعے سمجھایا گیا ہے۔

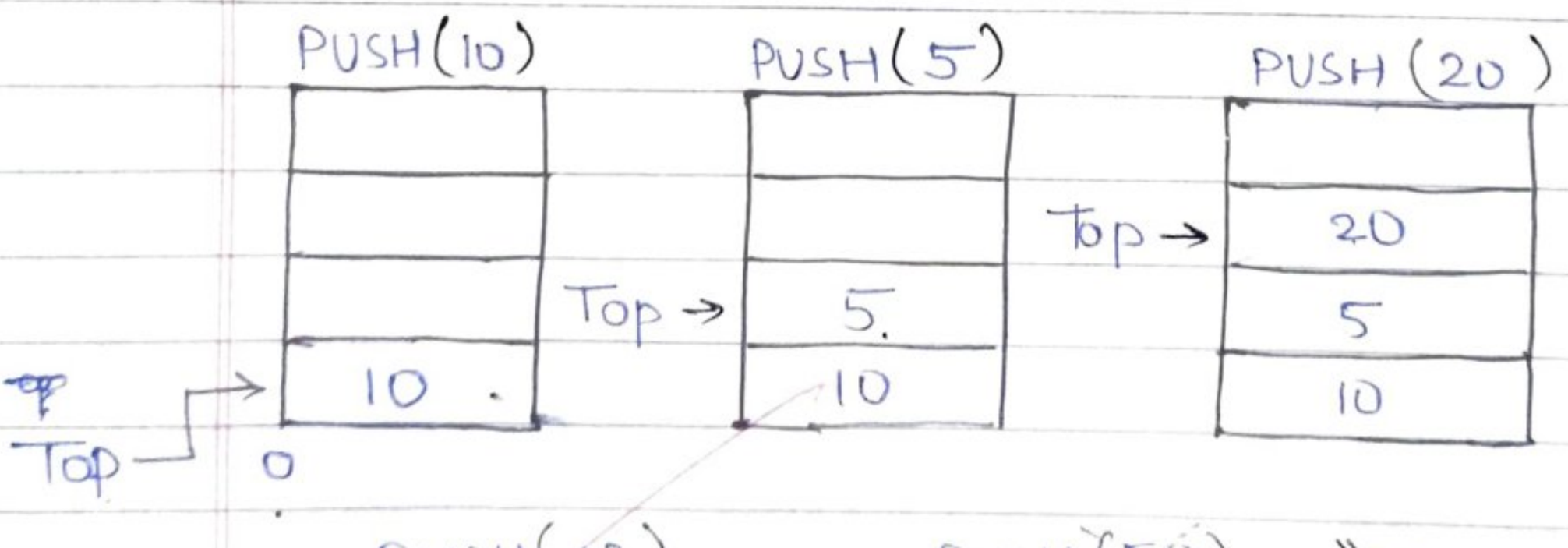
PUSH ALGORITHM

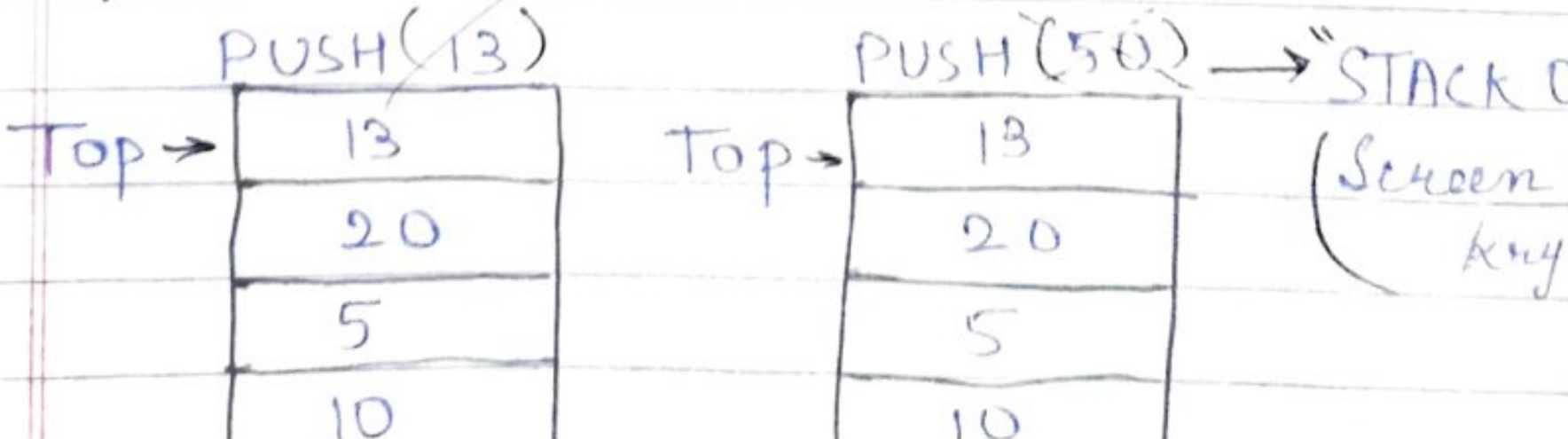
Step 1:- If Top = n then write "Stack Overflow" Exit.
Step 2:- Top ⟵ Top + 1
Step 3:- Stack[Top] ⟵ element
Step 4:- Return.



STACK

PUSH OPERATION





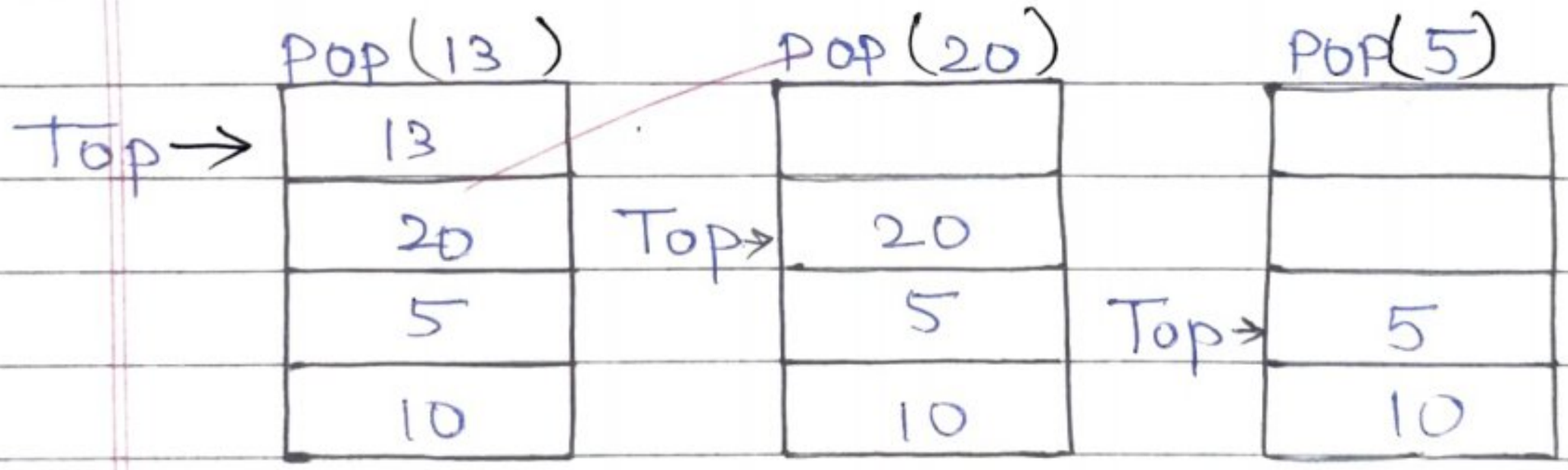
Date ___/___/___
Page _____

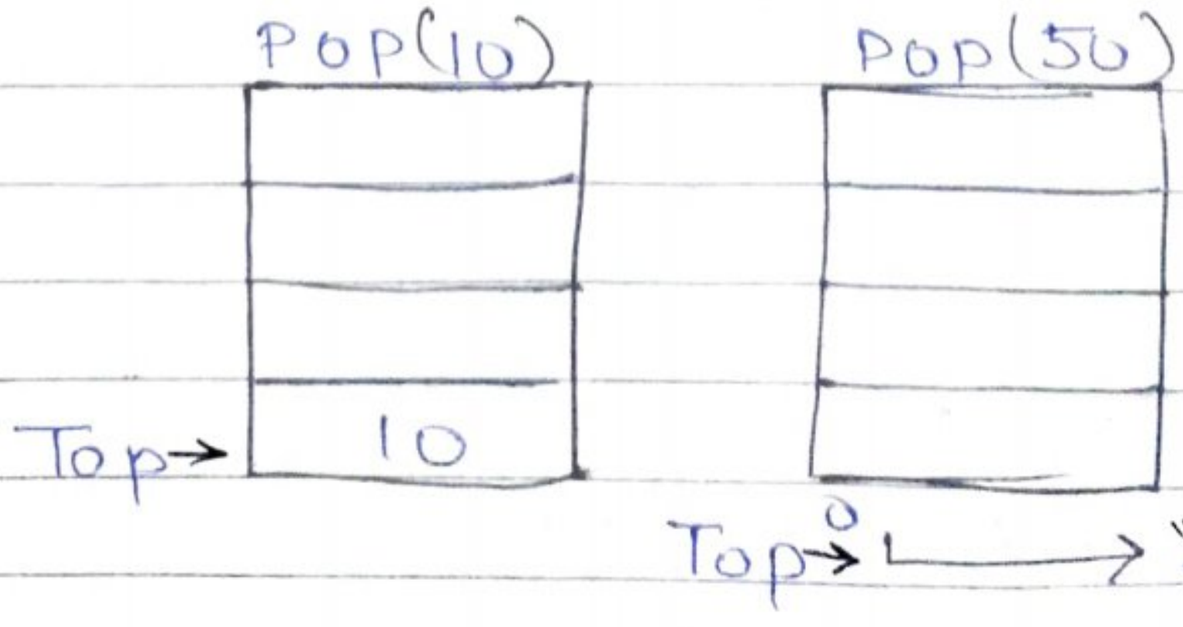
POP OPERATION

Stack میں POP کے عمل کو عناصر Element کو خارج کہتے ہے - اس کو عمل میں لانے کے لیے Stack کی جانچ کی جاتی ہے - اگر Stack "Empty" ہوگی تب Screen پر ایک message ظاہر کرینگے -
اور اگر "Empty" نہیں ہیں تب Top "STACK UNDERFLOW" کی مدد سے elements کو POP [Delete] کیا جائیگا -

POP ALGORITHM

Step1:- If Top = 0 then write "STACK UNDERFLOW" then Exit.
Step2:- Element ⟵ Stack Top
Step3:- Top ⟵ Top - 1
Step4:- Return.





Date / /
Page

Stack Application

1. String Reverse
2. Balance Parenthesis
3. Redo/undo.
4. System stack for Activation record (Recursion).
5. Infix, Prefix, Postfix Expression Evaluation.
6. Depth first search (DFS).

(1) * String reverse
Gatebook
koobetag

STACK.

(2) Balance Parenthesis

```
main ()
{ int x;
  clrscr ()
{ if (x = 10)
  pf ("      ");
  if (      )
}
else
{
}
}
```

(3) Redo/undo

operator

y = a + b → Expression.

operands

Infix

Expression. (<operand1> <operator> <opuand2>)

• 2+3, A+B, (p*q), (2+3)*4, (p+q)*(r+s)

y = 4 + 6 * 2 10 * 2 20	y = 4 + 6 * 2 = 4 * 12 = 16.
AMBIGUITY	

Date ___/___/___
Page ___

* Order of operator

1. Paranthesis (), {}, []
2. Exponents left to right (e^o)
3. Multi & division (left to right)
4. Add & Subt (left to right)

1. $y = 2 * 6 / 2 - 3 + 7$

$12 / 2 - 3 + 7$

$6 - 3 + 7$

$3 + 7$

10.

2. $a + b * c - d / e \wedge f$

$2 + 3 * 4 - 8 / 2 \wedge 2$

$2 + 3 * 4 - 8 / 4$

$2 + 3 * 4 - 2$

$2 + 12 - 2$

$14 - 2$

12.

Prefix operation (<operator><operand1><operand2>)

+ab

*ab

/ab

• Polish Notation •

Postfix operation (<operand1><operand2><operator>)

• reversepolish Notation

ab +

abc */

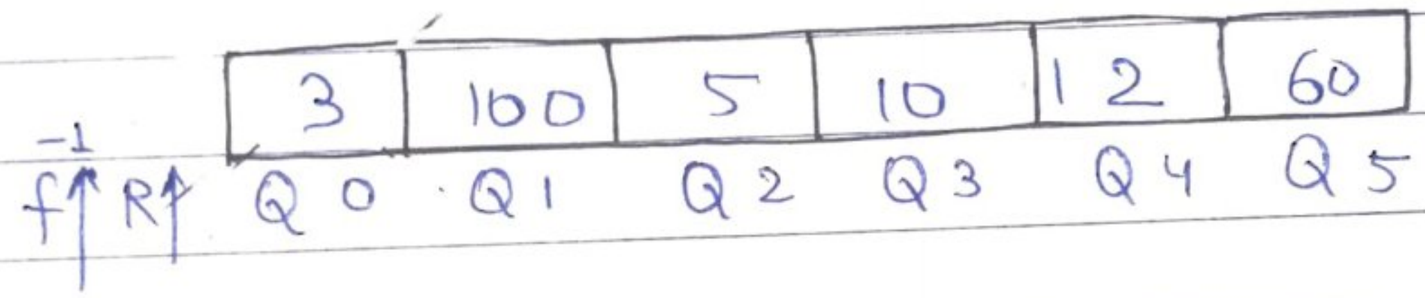
pq *

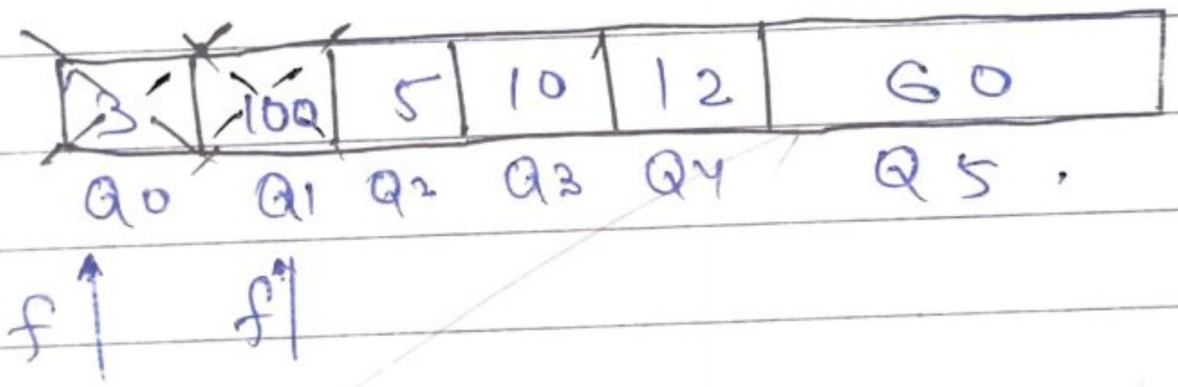
Infix	A+B+C	A+B-C * D	(A+B)/(C*D)
Port	ABC++	ABCD + - *	ABCD .
Pre	++ABC	+ - * ABCD	

chapter :- 2

* Queue

Rear (Tail) → Insert
Front (Head) → Delete





4/4/22

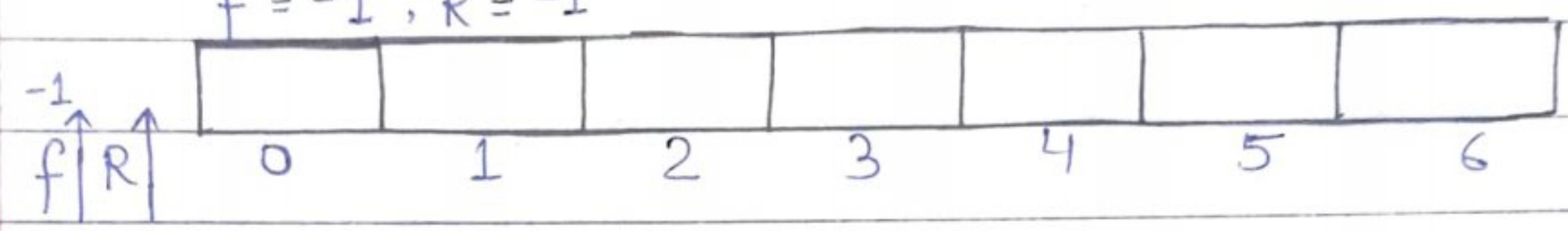
Chapter-2

Date ___/___/___
Page ___

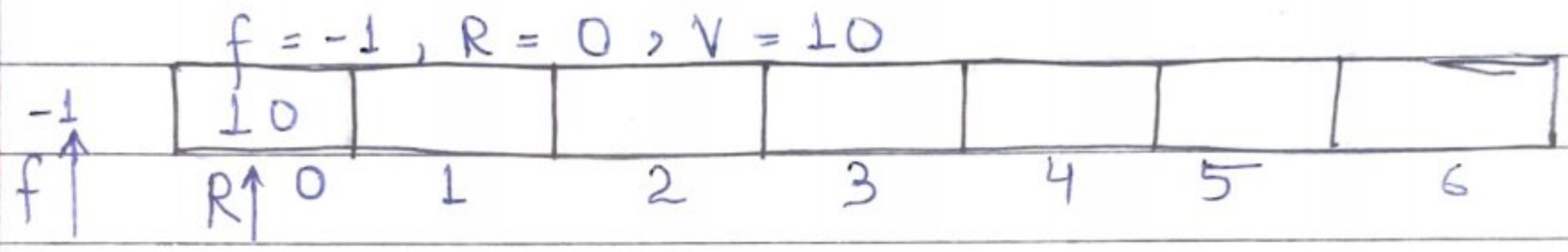
Queve [front, Rear] → Insert
└→ Delete.

* Queve Insert

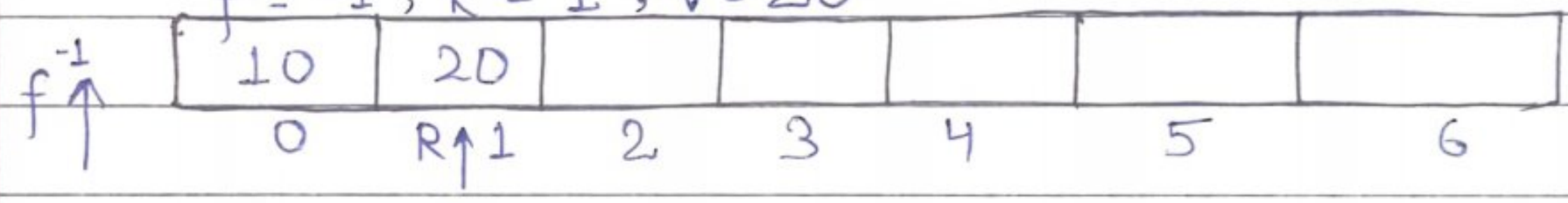
f = -1 , R = -1



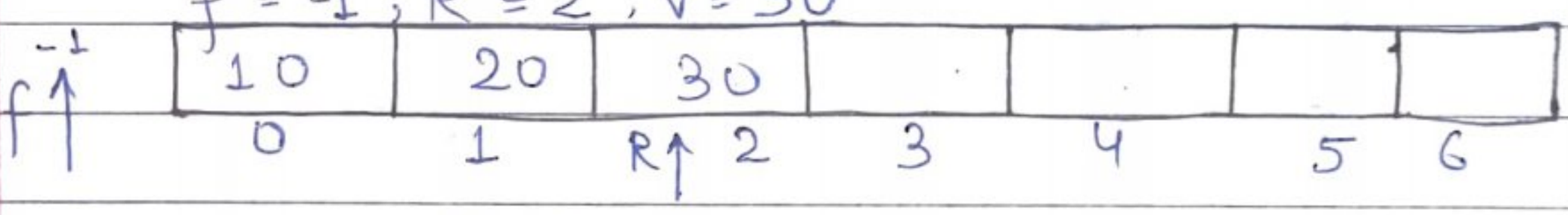
f = -1 , R = 0 , V = 10



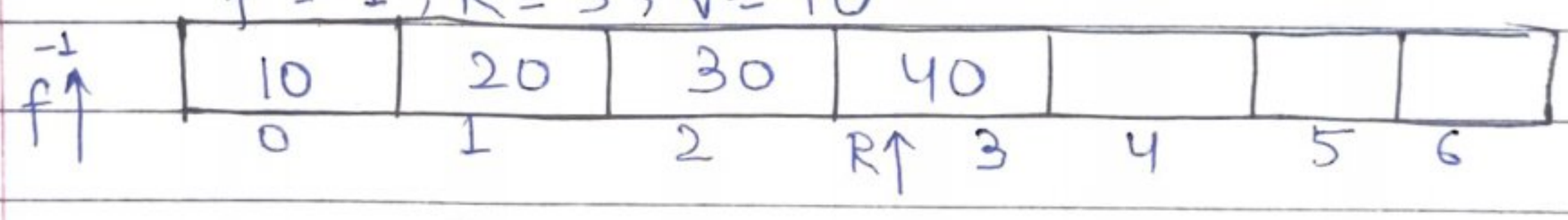
f = -1 , R = 1 , V = 20



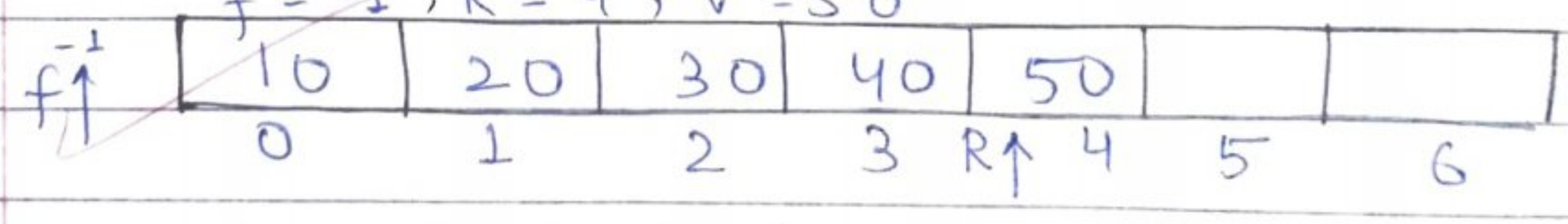
f = -1 , R = 2 , V = 30



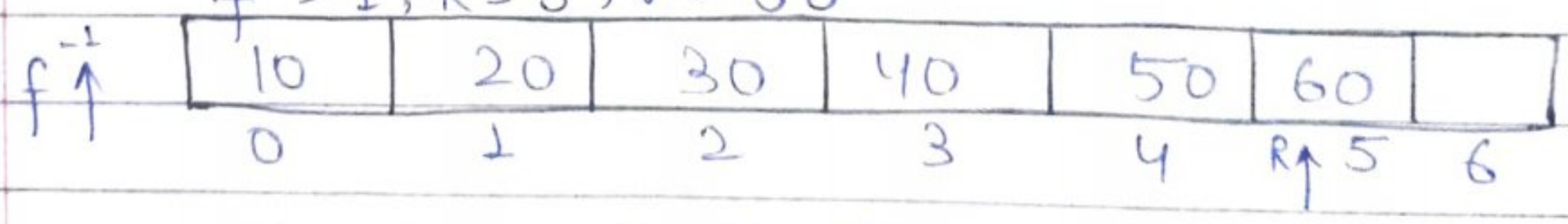
f = -1 , R = 3 , V = 40



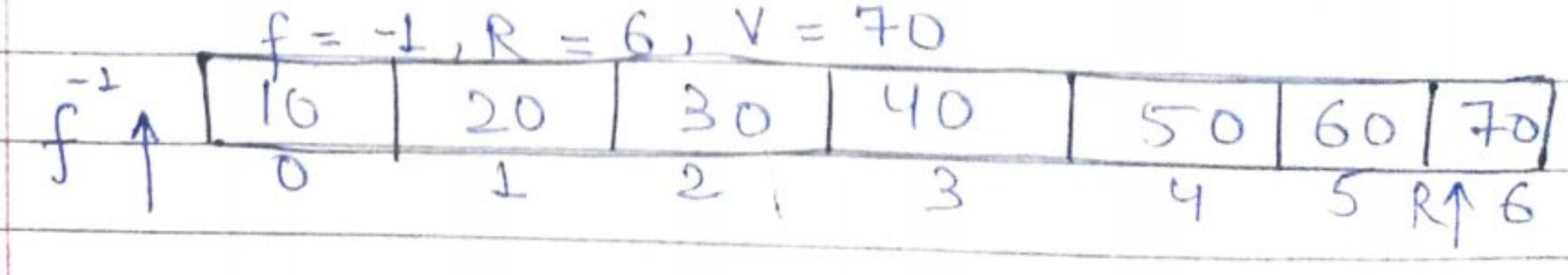
f = -1 , R = 4 , V = 50



f = -1 , R = 5 , V = 60

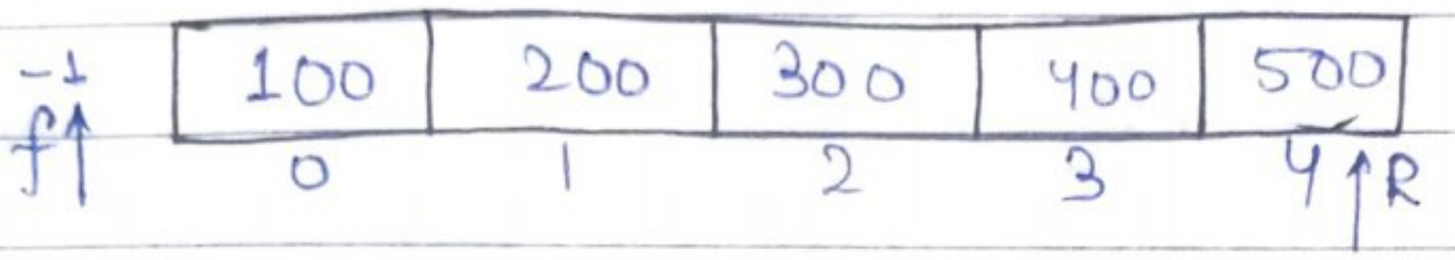


f = -1 , R = 6 , V = 70

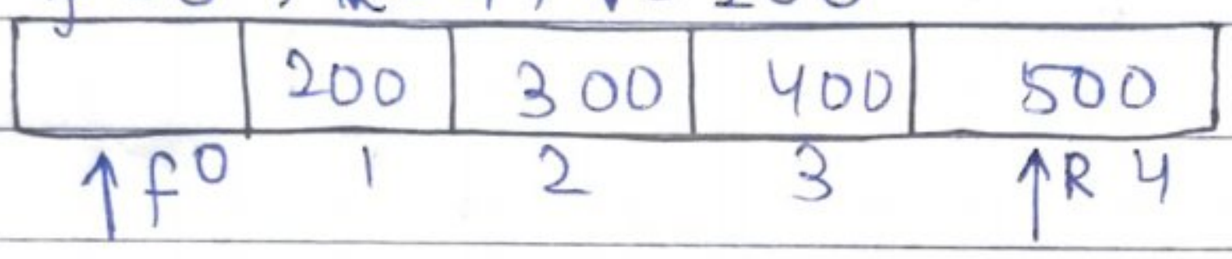


Date ___/___/___
Page _____

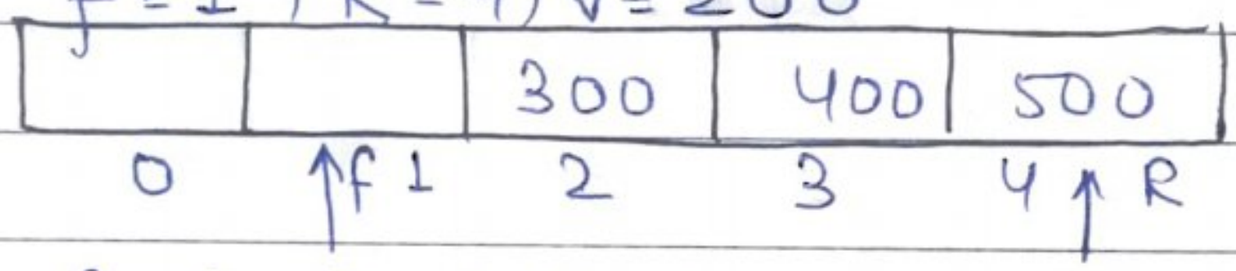
Queue Delete



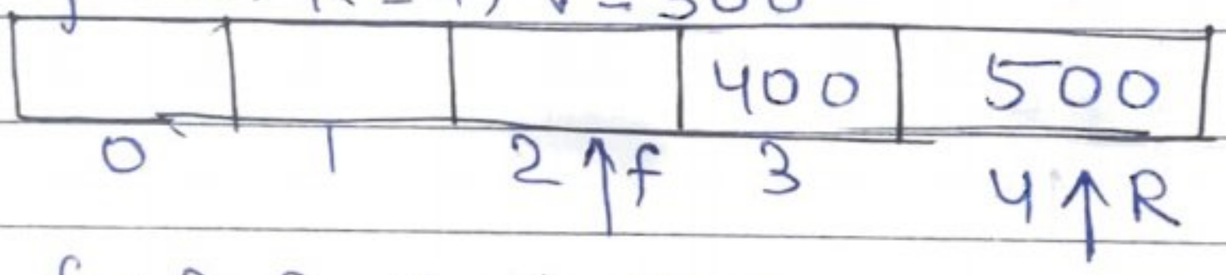
f = 0 , R = 4 , V = 100



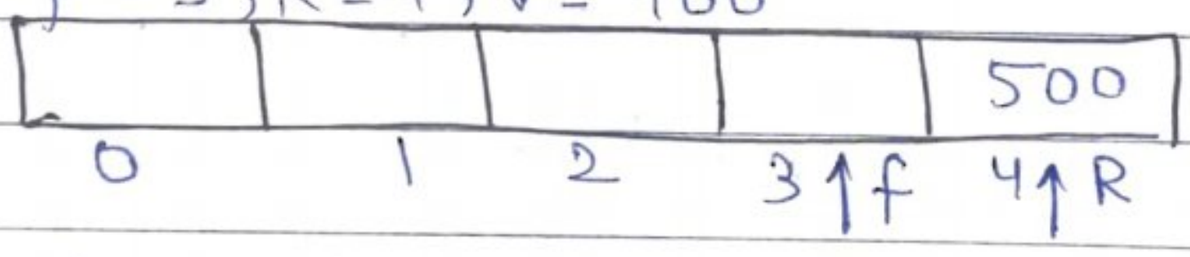
f = 1 , R = 4 , V = 200



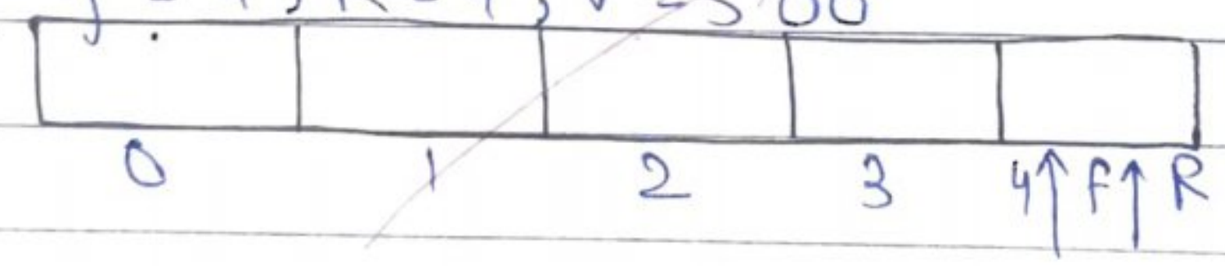
f = 2 , R = 4 , V = 300



f = 3 , R = 4 , V = 400



f = 4 , R = 4 , V = 500



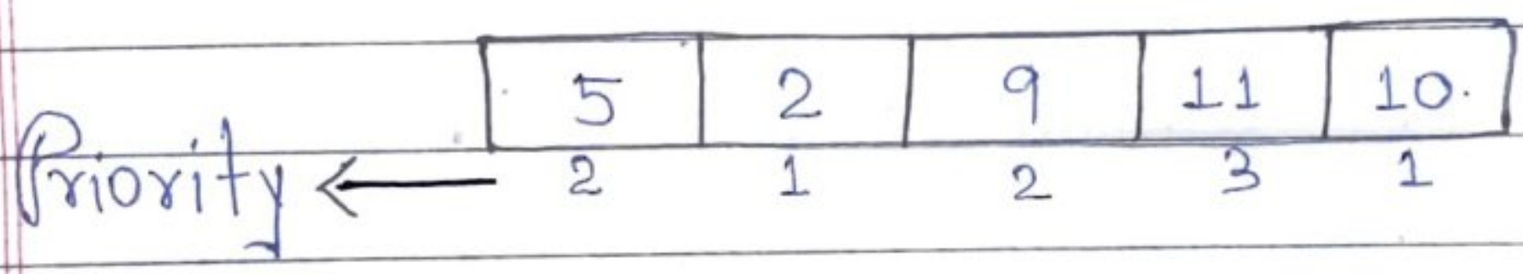
Date ___/___/___
Page ___

Types of Queue

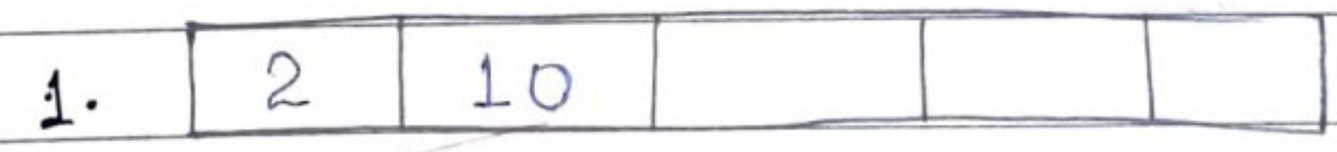
1. Simple Queue.
2. Circular Queue.
3. Double Ended Queue
4. Priority Queue. (ترجیحی)

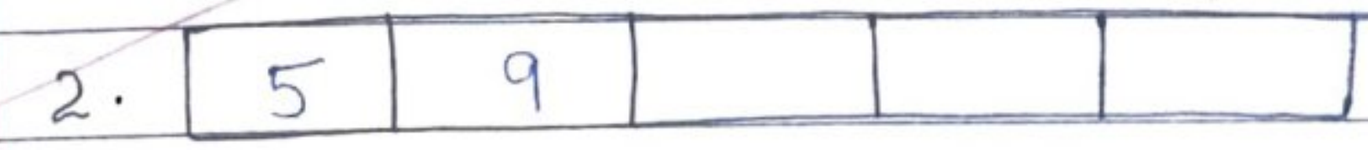
Priority Queue

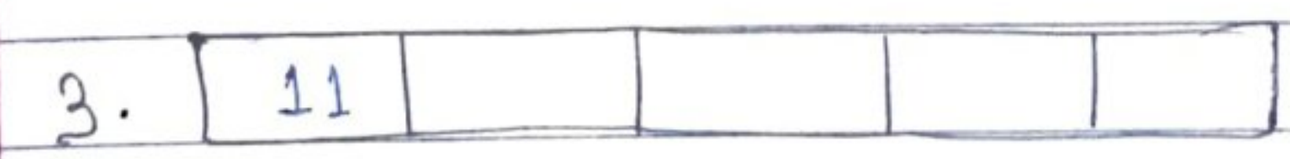
تمام عناصر کو ایک Priority [ترجیحی] اس میں کچھ عناصر کو اگر $P_1 = P_2$ تب FIFO انداز میں Higher Priority - حل کرینگے -



Multiple Array Queue [Insert]



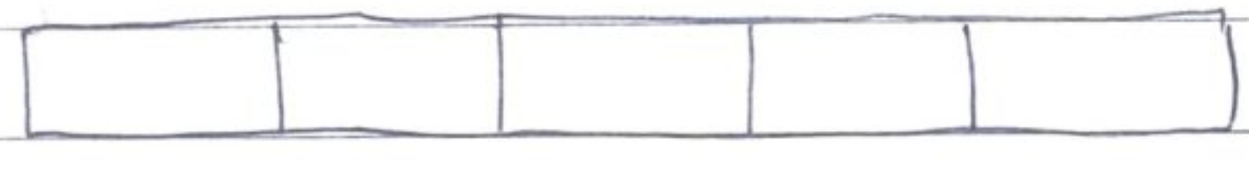




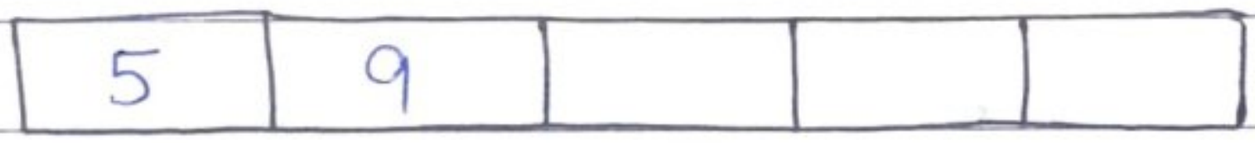
Date ___/___/___
Page _____

Multiple Array Queue

1.



2.



3.

Sparse Matrics

If zeros are more is called sparse Matrices.
for Example :-

$$\begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 9 \\ 0 & 0 & 3 \end{bmatrix}_{3\times 3}$$

Non-zero element = 4
Zero element = 5.

Date ___/___/___
Page _____

Example of :-

1. Low Triangle Matrix

$$\begin{bmatrix} 4 & 0 & 0 \\ 3 & 2 & 0 \\ 1 & 5 & 8 \end{bmatrix}_{3\times 3}$$

2. High Triangle Matrix

$$\begin{bmatrix} 2 & 4 & 5 \\ 0 & 7 & 8 \\ 0 & 0 & 1 \end{bmatrix}_{3\times 3}$$

3. Tridiagonal Matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}_{3\times 3}$$

#

	C_0	C_1	C_2	C_3
$R_0 \rightarrow$	0 (00)	0 (01)	1 (02)	3 (03)
$R_1 \rightarrow$	5 (10)	6 (11)	0 (12)	0 (13)
$R_2 \rightarrow$	0 (20)	0 (21)	9 (22)	0 (23)

3×4

$12 \times 2 = 24$ Byte

R	C	V
0	2	1
0	3	3
1	0	5
1	1	6
2	2	9

$5 \times 2 = 10$ Byte

Queue

یہ ایک ~~data~~ Linear data structure ہے- جو FIFO کے اندر میں کام کرتا ہے- FIFO کا جو Element سب سے پہلے اندر جائیگا وہ ہی سب سے پہلے delete ہوگا-

First in First out = FIFO

اس FIFO کو action میں لانے کے لئے Queue دو چیزوں کا استعمال کرتی ہے Front/Rear -

Rear Pointer کی مدد سے Queue سے Data کو ~~Delete~~ Insert کیا جاتا ہے -

Front Pointer کی مدد سے Queue سے Data کو ~~Insert~~ Delete کیا جاتا ہے -

* Queue کو دو چیزوں سے Action میں لایا جاسکتا ہے -

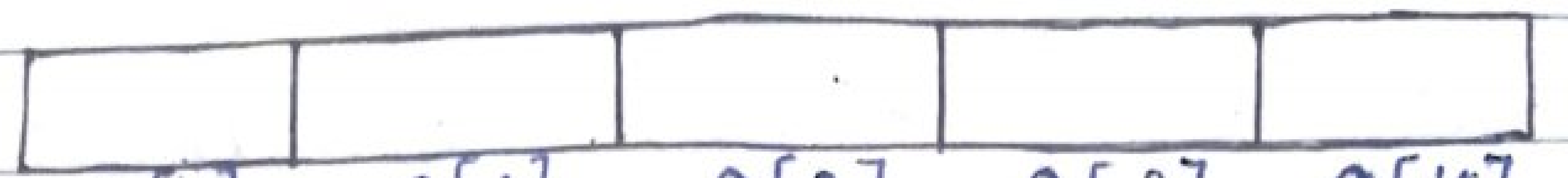
1. Array
2. Linked List

Types of Queue are :-

1. Simple Queue
2. Circular Queue
3. Double Ended Queue.
4. Priority Queue

1. Simple Queue

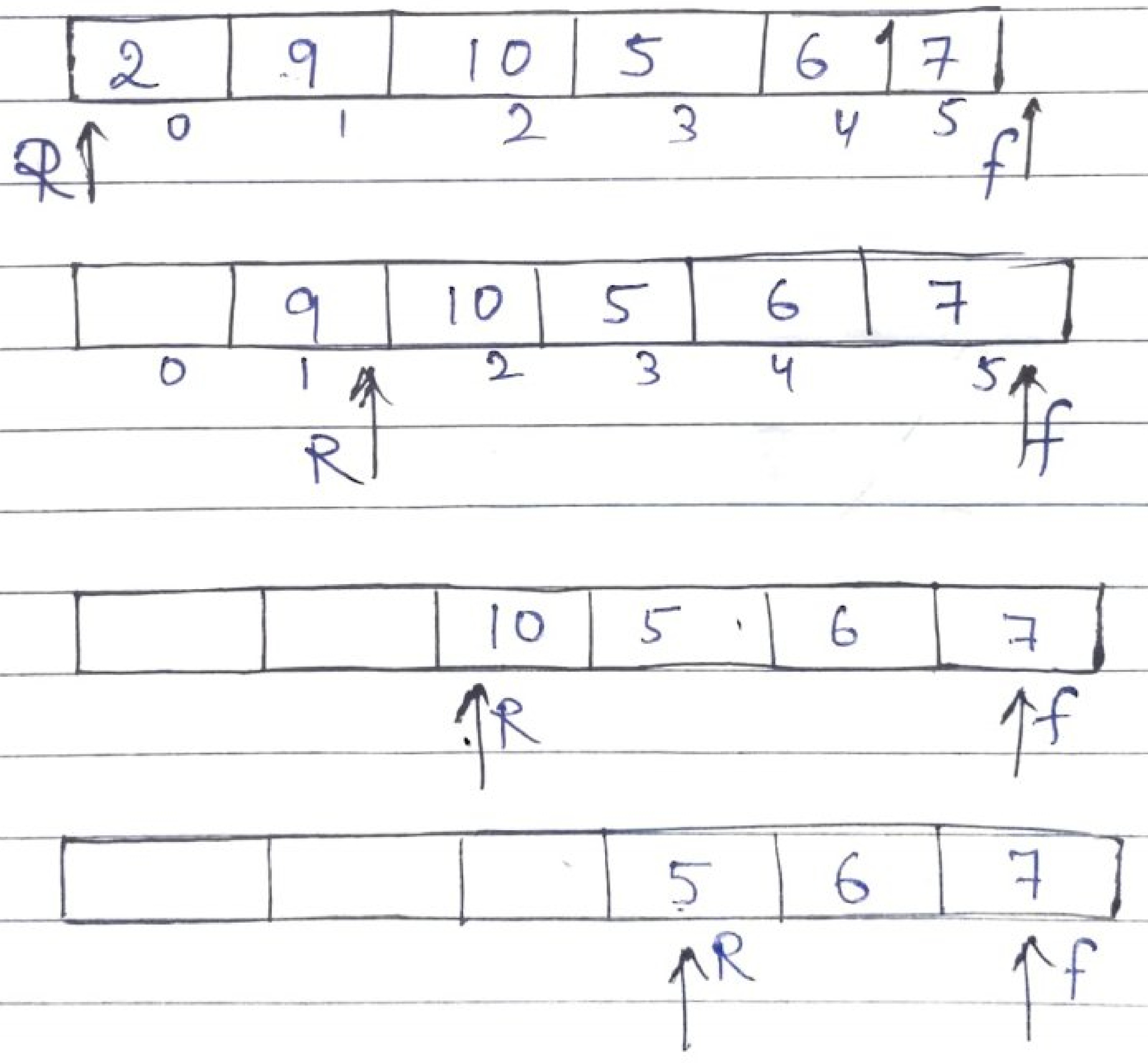
Rear -1
front -1



Rear / Front کی مدد سے سارا کام End پر ہوتا ہے۔

اگر f = a[4] تب کوئی بھی Element کو Insert نہیں کر سکتے ہے -

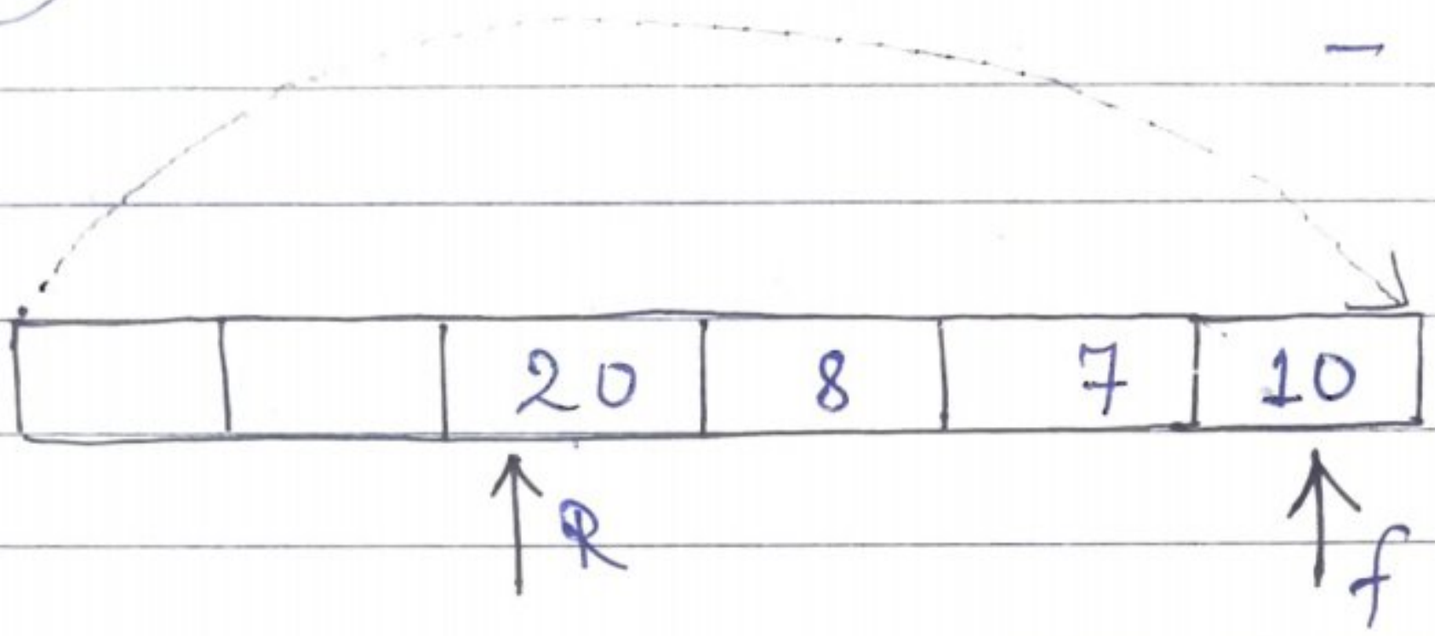
اگر Array میں کچھ Element کو delete ہونے کے بعد بھی Simple Queue میں ہم Insert data نہیں کر سکتے -

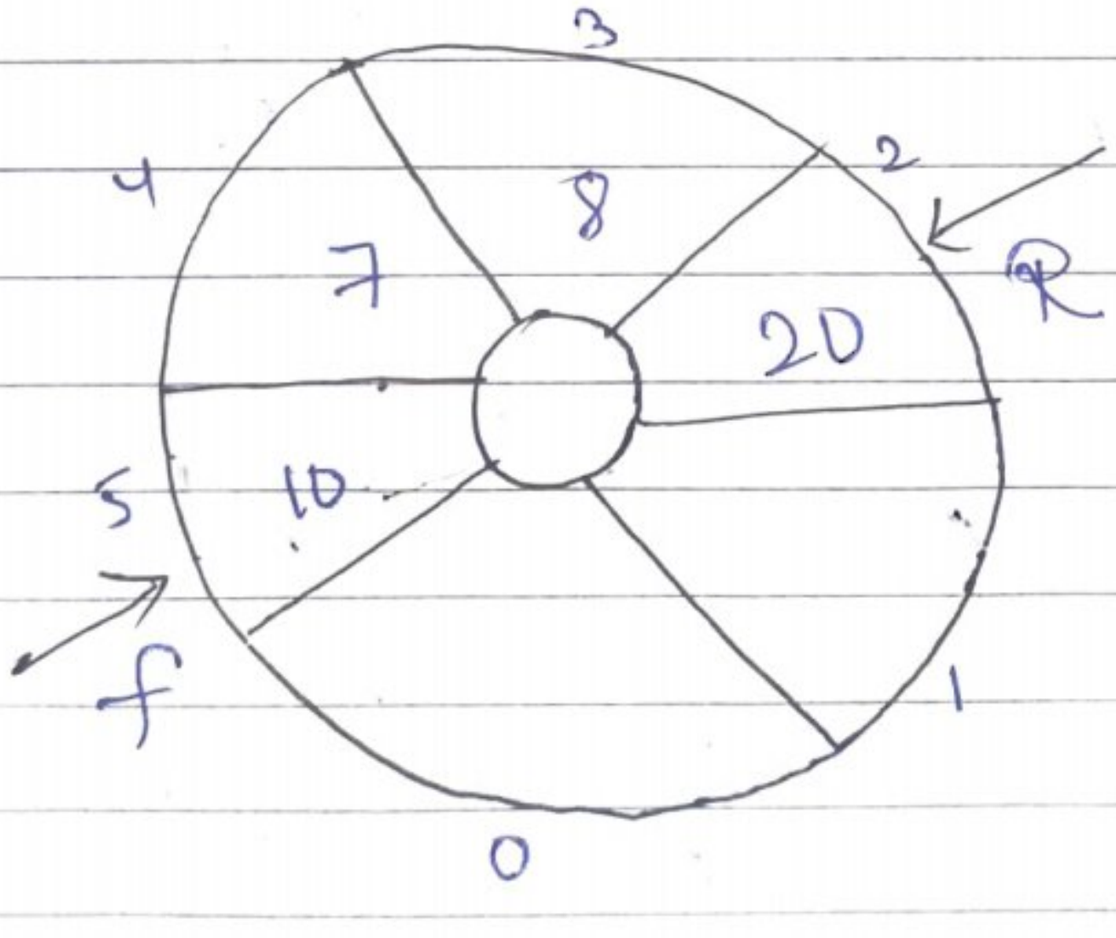


Date / /
Page

2. Circular Queue

جس طرح Simple Queue میں جگہ ہونے کے باوجود ہم کوئی بھی Element کو Insert نہیں کر سکتے جس سے ہماری Memory میں موجود جگہ waste ہو گی اس خامی کو دور کرنے کے لئے Circular Queue کی ضرورت پیش آئی ہے۔





یعنی جب Queue Empty ہو گی تب $f/R = -1$

لیکن ہو سکتے

$f = -1$

$R = -1$

4. Priority Queue

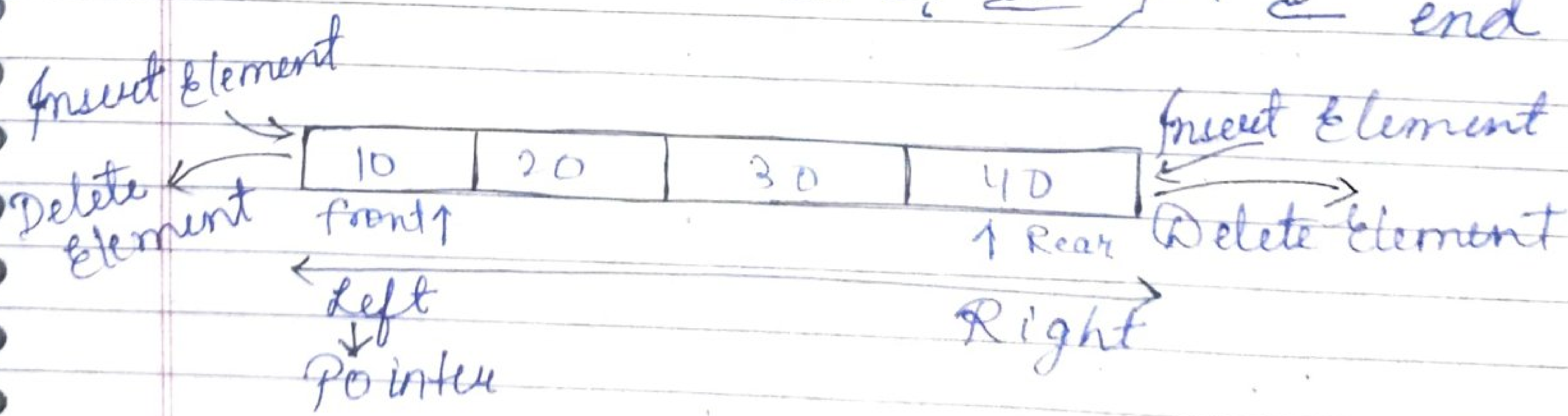
روز مرہ کی زندگی میں ہم عام طور پر To-Do-List
بناتے ہیں یعنی آج کے تمام کام کو ایک فہرست (List)
بناتے ہیں - اس میں کون سے کام کو priority دینا
چاہیئے اس طرح ہم Queue کو بھی استعمال کرتے ہے -

	5	2	9	1	8	10
priority ←	2	1	2	3	1	2

تمام Element کو ایک priority دی جاتی ہے -
جس Element کو "Higher priority" دی جاتی ہے -
اگر دو Element کو Same priority ہو جائے تب ہم
FIFO کا رول سے اس کی priority کرتے ہے -

3. Double Ended Queue

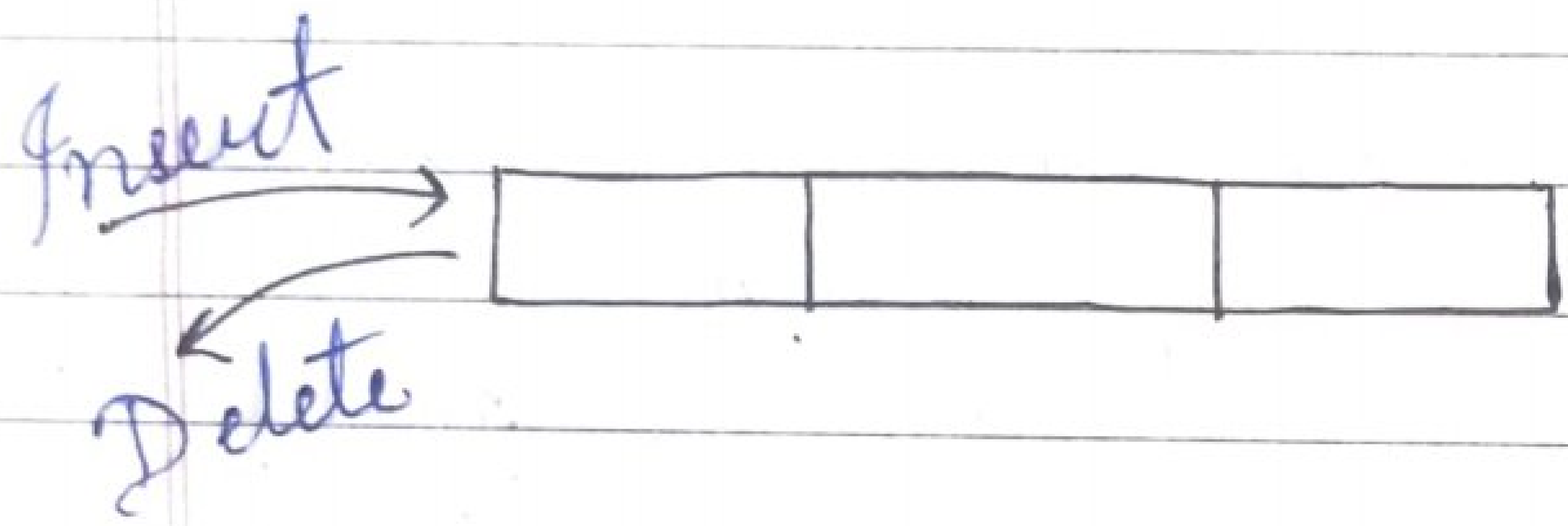
یہ بھی Linear data structure ہے -
اس میں ہم Insertion اور deletion دونوں Rear/front
end سے کر سکتے ہے -



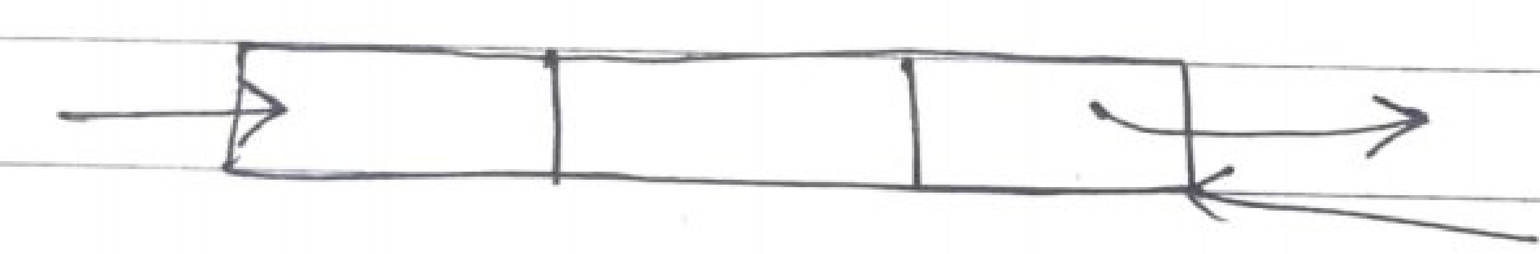
اس کو ہم یہ بھی کہہ سکتے ہیں کہ اس میں Stack
اور Queue کی خصوصی بات کو استعمال کیا گیا ہے۔

Types
1. Input Restricted de-Queue,
2. Output Restricted de-Queue

1. Input Restricted de-Queue
اس میں Input کسی ایک end سے ہی ہوتا
اور deletion بھی

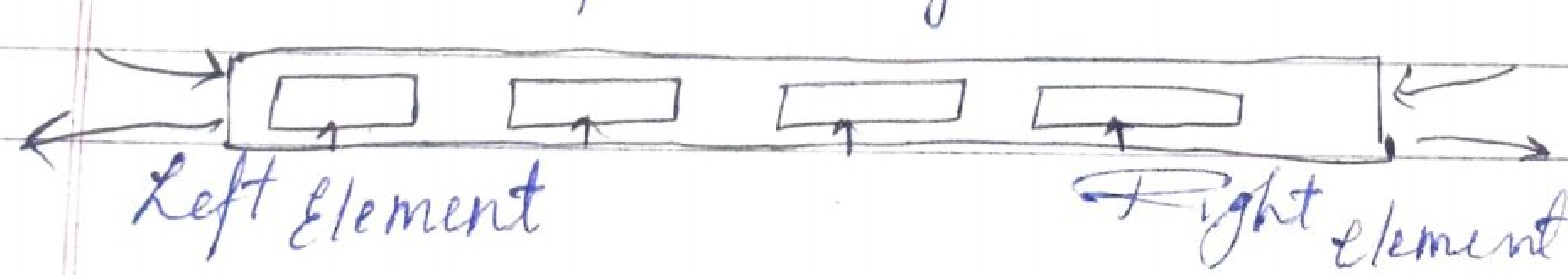


2. Output Restricted de-Queue
output کسی ایک end سے ہی ہوتا ہے -



* Application of double ended Queue
1. palindrome. Madam, DAD
2. Multiprocess
3. undo & Redo operation

* operation on de queue
Insertion from left end
Insertion from Right end
Deletion from left end
Deletion from Right end.



Date ___/___/___
Page _____

اس کو بھی Dc - Queue ہم کہتے ہیں۔ جب
وہ کوئی end دونوں deletion اور insertion
کی کرے double ended Queue

اس میں بیچ میں Element
delete اور Insert کو
نہیں کر سکتے ہے۔

follow کی fashion FIFO
نہیں کرتا ہے۔

Types :-
1. Input Restricted De-Queue
2. Output De-Queue.

1. Input Restricted De-Queue.
اس allow کو صرف ایک end insertion
کی ہے لیکن deletion کو دونوں side سے کر سکتے ہے۔

2. Output De-Queue
اس میں دونوں side insertion کر سکتے ہیں۔ لیکن
deletion ایک ہی end سے کر سکتے ہے۔

Ch-3

Searching & Sorting

* Searching

Data structure میں وہ technique کوئی بھی Element کو ~~موجودہ کو Element~~ موجودہ list میں تلاش کرنے کے لیے استعمال کیا جاتا ہے۔

اگر Element list میں موجود ہو تب اس کو successful search کہتے ہیں۔ اور اگر Element موجود نہ ہو تب اسکو unsuccessful search کہتے ہیں۔

Searching کے program میں search کرنے والی value موجود ہونے پر "Match found" والا Message ظاہر کرینگے اور اگر موجود نہ ہو تو "Match Not Found" والا Message ظاہر کرینگے۔

Two type of Searching are :-

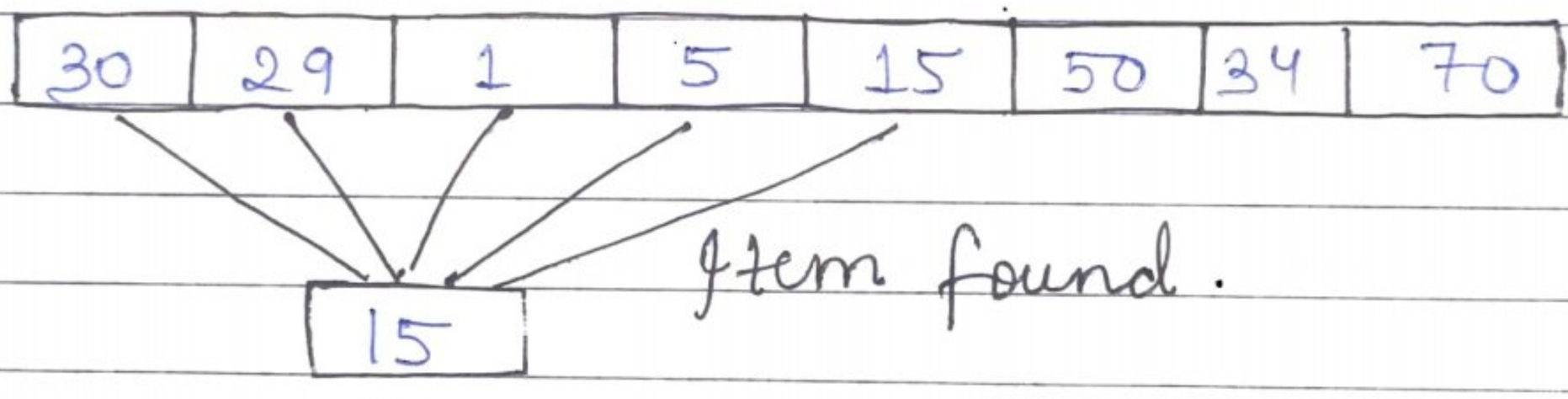
1. Linear Search
2. Binary Search.

Date ___/___/___
Page ___

1. Linear Search

- یہ سب سے Search Sequential پر ہوتا
- یہ searching کی method simplest اور sequentially Element پر کی جانے والی Method ہوتا
- یہ list میں کرتے ہیں found کے لئے searched unsorted اور sorted میں کرنا applied Method ہے۔
- یہ list کو searching کو 0's سے شروع کرینگے اور تب تک searching کرینگے تب تک کے found Element نہ ہو جائے۔

for ex:-



Key

Time complexity = 5 sec

Algorithm:-

```
for (i=0; i<10; i++)
{
    if (key == A[i])
    printf("Item found");
    else
    printf("Item not found");
}
```

Date ___/___/___
Page _____

2. Binary Search

efficient اور تیز fast اور Method ہے۔

* Sorting

Sorting کا مطلب ہوتا ہے set of data کو
arrange کرنا ہوتا ہے ascending اور descending
order میں ۔

Two categories are:-

1. Internal Sorting
2. External Sorting

1. Internal Sorting

پورا data میرا جب sort ہو جاتا ہے Memory
میں ایک وقت میں ایڈجسٹ ہو جاتا ہے تب ہم
Internal sorting Method کا استعمال کرتے ہے ۔

2. External Sorting

جب data میرا sort ہوتا ہے large میں
تب ہم data store ہوتا ہے Memory میں اور
ہم data store ہوتا ہے Auxiliary Memory
(harddisk, floppy, tape..etc) میں تب ہم
External Sorting Method کا استعمال کرتے ہے ۔

Types of Sorting are:-

1. Selection Sort
2. Insertion Sort
3. Quick Sort
4. Merge Sort
5. Bubble Sort

1. Selection Sort

یہ simplest Method ہے sorting کا -
اس میں ہم data کو sort کرتے ہیں
ascending order میں -
0th Element کو ہم compared کرتے ہیں other سے
Element سے اگر جو element greater ہو تو
اس کو آگے compare کر کے interchanged کر دیتے ہیں -

Algorithm.

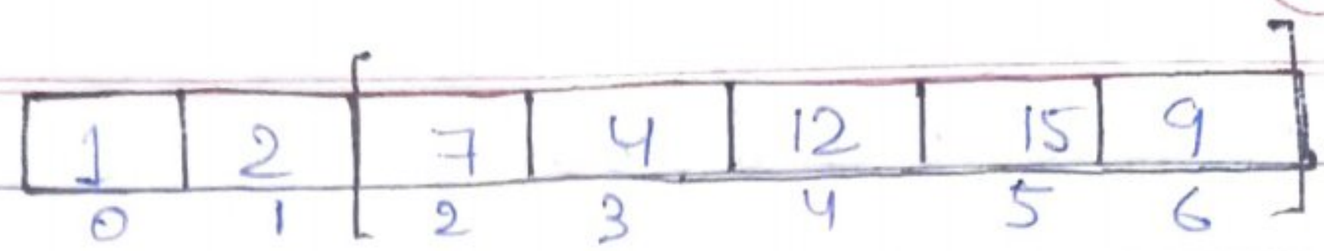
1. Set min = 0
2. Search minimum in the list
3. Swap the value at min location.
4. Increase min value to point next element.
5. Repeat until list is sorted.

2	9	7	4	12	15	1
0	1	2	3	4	5	6

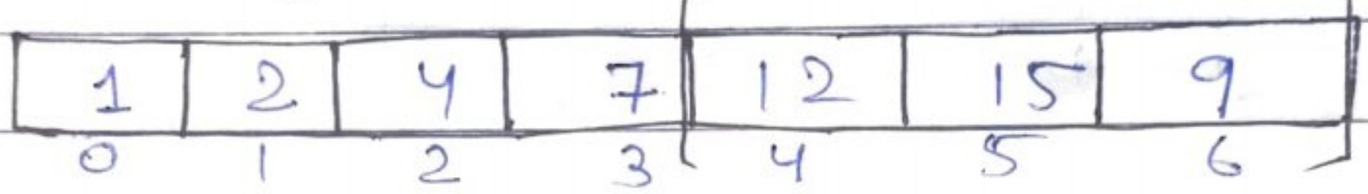
Min = 0
index = A[0]
value = 2
Swapping value = 1

1	9	7	4	12	15	2
0	1	2	3	4	5	6

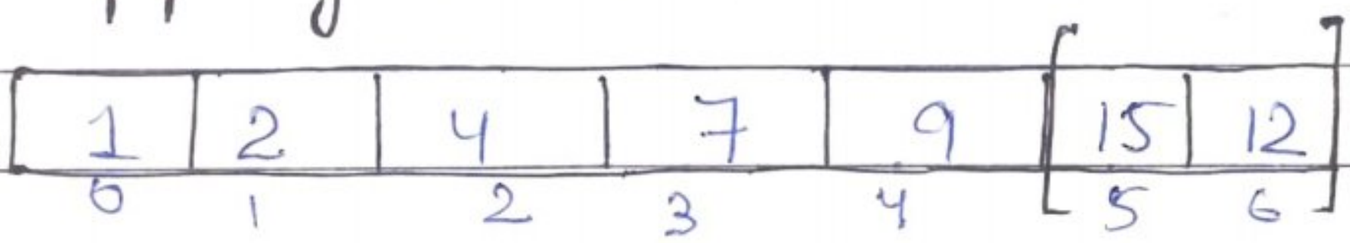
Min = 0
index = A[1]
value = 9
Swapping value = 2



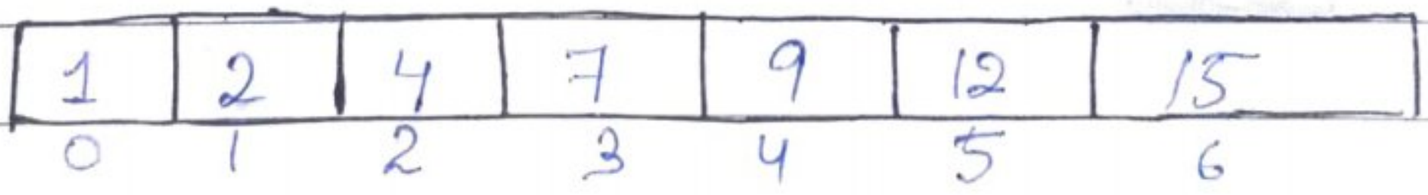
Min = 0
index = A[2]
value = 7
Swapping value = 4



Min = 0
index = A[4]
value = 12
Swapping value = 9



Min = 0
index = A[4]
value = 12
Swapping value = 9



The List is Sorted

2. Insertion Sort

* Not suitable for large item .
* Comparing item based on range value & insertion it at correct location

9	5	20	19	30

|←— R —→| (spans 9, 5)

5	9	20	19	30

|←— R —→| (spans 5, 9, 20)

5	9	19	20	30

|←— R —→| (spans 5, 9, 19, 20)

5	9	19	20	30

|←— R —→| (spans 5, 9, 19, 20, 30)

3. Quick Sort

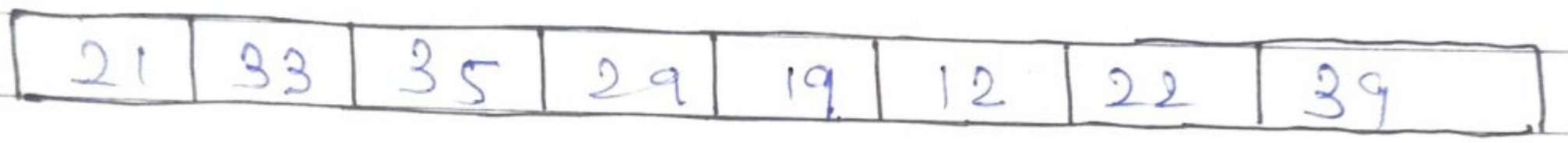
یہ سب popular sorting method ہے -
Quick sort جو ہے وہ selection, Insertion sort کے ~~[struck out]~~ مقابلے list کو fastly sort کر لیتا ہے -

4. Merge Sort

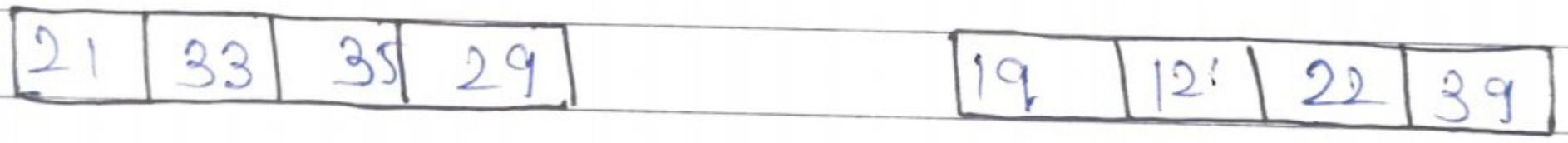
دو الگ الگ list کو ایک ~~sorted list~~ sorted list بناتے ہے۔

دو الگ الگ list کو sort کر کے Merge کیا جاتا ہے۔

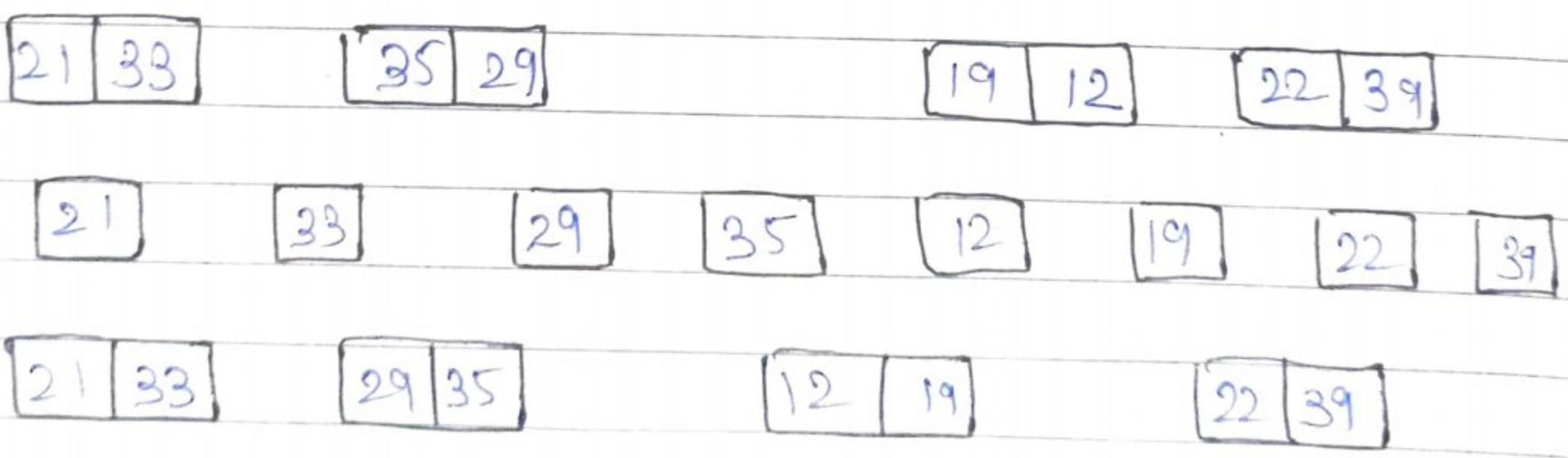
Divide & conqre.

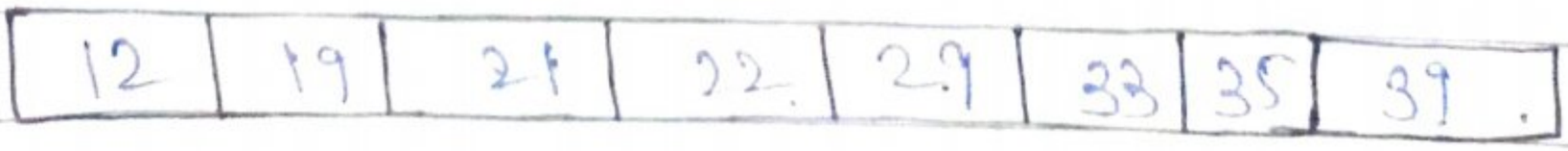


Divide



Divide





5. Bubble Sort

یہ arrange کے Element کا Method ہے جو order Ascending کا ہے۔

First Iteration

25	17	31	13	2

17	25	31	13	2

17	25	31	13	2

17	25	13	31	2

17	25	13	2	31

Second Iteration

17	25	13	2	31

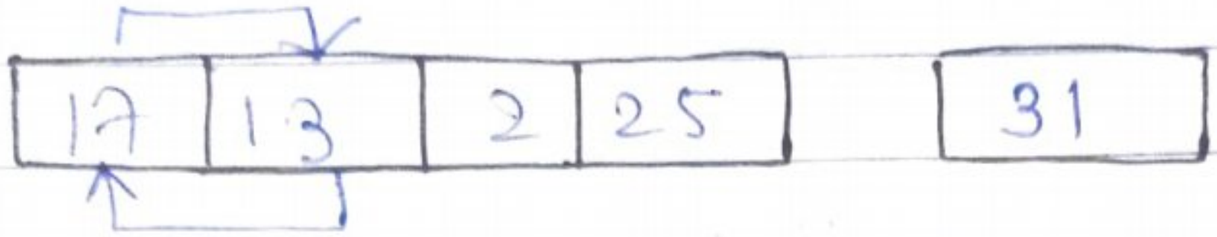
17	25	13	2	31

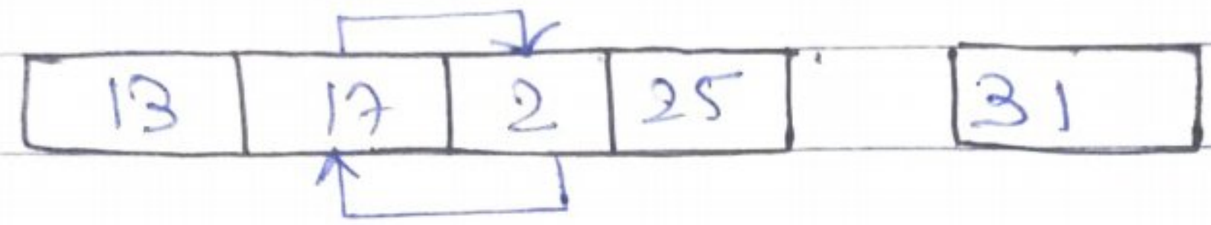
17	13	25	2	31

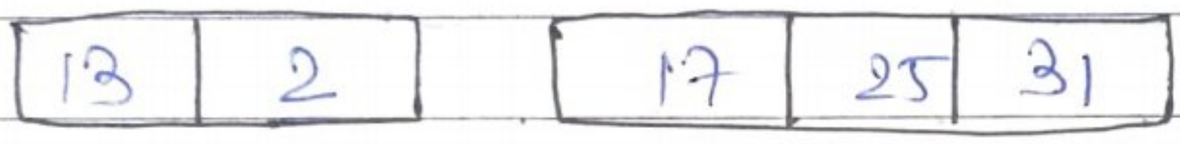
17	13	2	25	31

Date / /
Page

Third Iteration







fowith Iteration