2. Array

- **Array:**

A linear array is a list of finite number ~~of~~ a 'n' of homogenous data element such that:
a) The element of the array are referenced respectively by an index set consisting of n consicutive numbers
b) The element of the array are stored respectively in succesive memory location.

- **Syntax:-** int i [s]

i is the name of an array of integer data type having size s.

- **Length of Array :-**

[UB - LB + 1]

int i[5] ; 1 + 0 5 ∴ L = 1 - 5 + 1
↓ L.B ↓ U.B

e.g An automobile company uses an array auto to record the number of automobile sold each year from 1932 to 1984. So identify no. of elements in an array auto.

1932 to 1984
↓ L.B ↓ U.B

∴ L = UB - LB + 1
= 1984 - 1932 + 1
= 52 + 1
= 53.

- Representation of Linear array in memory.

Loc(LA[K]) = Base(LA) + W(K - LB)
(LB) (W.B)

e.g. Array Auto :- 1932 to 1984.
Base[Auto] = 200
W = 4 words per memory cell
K = 1965

Loc(Auto[1965]) = Base(Auto) + W(K - LB)
= 200 + 4(1965 - 1932)
= 200 + 4(33)
= 200 + 132
= 332.

e.g.
Base[Auto] = 200
W = 4
K = 1933

Page No.
Date :

$$\text{Loc}(\text{Auto}[1933]) = \text{Base}(\text{Auto}) + w(k - LB)$$
$$= 200 + 4(1933 - 193$$
$$= 200 + 4$$
$$= 204$$

e.g Consider the linear array
AAA (5 : 50)
BBB (-5 : 10)
CCC (18)

a) find the no. of element in each array
b) Suppose Base (AAA) = 300, w = 4 per memory cell. find the address of AAA [15], AAA [35], AAA [55]

Ans:- a)
i) Length of Array AAA [5 : 50]

$$= UB - LB + 1$$
$$= 50 - 5 + 1$$
$$= 45 + 1$$
$$= 46$$

ii) Length of Array BBB [-5 : 10]

$$= UB - LB + 1$$
$$= 10 + 5 + 1$$
$$= 16$$

Page No.
Date : / /

iii) CCC[18]

∴ Length of Array = 18

b) Ans:-

i) Array AAA :- (50 to 50)
Base[AAA] :- 300
W :- 4
K :- 15.

Loc(Auto AAA[15]) = Base[AAA] + W(K-LB)
= 300 + 4(15-5)
= 300 + 40
= 340

ii) Array AAA :- (50 to 50)
Base[AAA] :- 300
W :- 4
K :- 35

Loc(AAA[35]) = Base[AAA] + W(K-LB)
= 300 + 4(35-5)
= 300 + 4(30)
= 300 + 120
= 420

Page No.
Date : / /

• Accessing each and every element exactly once
• Traversing in an array.

Beacon
Devid
John
Smith
Coagnee

• Algorithm
Linear (LA, K, LB, UB)
① Set K := LB
② Repeat step ③ & ④ While K ≤ UB
③ Apply process (function of traversing) to LA[K]
④ Set K : K + 1
⑤ Exit.

• Insertion in an array.

1 Beacon Insert => fored
2 Devid
3 John
4 Smith
5 Coangee
6
7
8
9

Page No.
Date : / /

[LA]

⇒ Here ^ array ^ is an linear array with N elements and k is a positive integer such that $K \leq N$.

This algorithm insert an element item into k^{th} position in LA.

INSERT(LA, N, K, item) [J is counter variable]

1] Set J = N
2] Repeat Step ③ & ④ while $J \geq K$.
3] set LA[J+1] :- LA[J]
4] set J := J - 1
5] set LA[K] := item.
6] set N = N + 1
7] Exit

Deletion in an array

Here LA is an linear array with N elements and k is a posite integer such that $K \leq N$ this algorithm delete the k^{th} element from LA

DELETE (LA, K, N, item)

1] Set item := LA[K]
2] Repeat for J := K to N-1

3] Set LA[J] := LA[J+1]
4] Set N := N - 1
5] Exit

• Binary Search (Array should be in ascending order)

If not then L.S

Array :-

Q. DATA = 11, 22, 30, 33, 40, 44, 55, 60, 66, 77, 80, 88, 99.

⇒ item = 40

(Begin) ① L.B ⇒ 1

(End) U.B ⇒ 13

$$mid = \frac{LB + UB}{2} = \frac{1+13}{2} = 7$$

DATA[7] = 55 ≠ item

Now

item < data[mid]

40 < 55 (TRUE)

End = mid - 1 ⇒ 7 - 1 = 6

② Beg ⇒ 1, End = 6

$$mid = \frac{Beg + mid}{2} = \frac{7}{2} = 3.5 = 3$$

Data[3] = 30 ≠ item.

item < data[item]

40 < 30 (false)

Page No.
Date : / /

Beg ⇒ mid + 1 ⇒ 3 + 1 = 4

③ Beg ⇒ 4 End ⇒ 5

$$mid = \frac{4+5}{2} = \frac{10}{2} = 5$$

DATA [5] = 40 = item

Item is present in an array DATA at location 5

If (item < Data (mid))

2]

item = ~~100~~ 77
Begin = 1
End = 13

① Beg = 1, End = 13

$$mid = \frac{Beg + End}{2} = \frac{1+13}{2} = 7$$

DATA [7] = 55 ≠ item,
Now

item < data (mid)
77 < 55

Beg ⇒ mid + 1 ⇒ 7 + 1 = 8

② Beg ⇒ 8, End = 13.

$$mid = \frac{8+13}{2} = 21/2 = 10.5 = 10$$

Page No.
Date : / /

DATA[10] = 77 = item

Item is present in an array
@ DATA at location 10

3) item = 100
Begin = 1
End = 13

① Beg = 1 , End = 13
mid = $\frac{Beg + End}{2} = \frac{1+13}{2} = \frac{14}{2} = 7$

DATA[7] = 55 ≠ item
Now,
item < data[mid]
100 < 55

Beg ⇒ mid + 1 ⇒ 7 + 1 = 8

② Beg = 8 , End = 13
mid = $\frac{8+13}{2} = 21/2 = 10.5 = 10$

DATA[10] = 77 ≠ item
Now,
item < data[item]
100 < 77

Beg ⇒ mid + 1 ⇒ 10 + 1 = 11

Page No.
Date : / /

③ Beg = 11 End = 13

$mid = \frac{11+13}{2} = \frac{24}{2} = 12$

DATA[12] = 88 ≠ item

Now

item < data(item)

100 < 88

Beg ⇒ mid + 1 = 12 + 1 = 13

④ Beg = 13 End = 13

$mid = \frac{13+13}{2} = \frac{26}{2} = 13$

DATA[13] = 99 ≠ item

Now

item < data(item)

100 < ~~88~~ 99

Beg = mid + 1 = 13 + 1 = 14

Beg = 14 and End = 13.

Since begining is greater than end we cannot perform searching operation further 100 is not located in this series.

Page No.
Date : / /

Algorithim

BINARY [DATA, LB, UB, ITEM, LOC)

1) Set BEG := LB, END := UB,
   MID = INT $\left(\frac{BEG + END}{2}\right)$
2) Repeat steps 3 & 4 while
   BEG $\leq$ END and DATA[mid] $\neq$ item
3) if ITEM < DATA[mid] then
   END = mid - 1
   else
   BEG = mid + 1
4) Set mid = INT $\left(\frac{BEG + END}{2}\right)$
5) if DATA[mid) = item then set
   loC := mid else
   Set loC := NULL
6) Exit

Page No.
Date : / /

* Sorting

Bubble Sort :-

Let, A be a list of n numbers. Sorting A refers to the operation of rearranging the elements of A so they are in increasing order.

Q.1 32, 51, 27, 85, 66, 23, 13, 57.

Pass - I → 32, 51, 27, 85, 66, 23, 13, 57
32, 51, 27, 85, 66, 23, 13, 57
C = 7 32, 27, 51, 85, 66, 23, 13, 57
I = 5 32, 27, 51, 85, 66, 23, 13, 57
32, 27, 51, 66, 85, 23, 13, 57
32, 27, 51, 66, 23, 85, 13, 57
32, 27, 51, 66, 23, 13, 85, 57
32, 27, 51, 66, 23, 13, 57, 85

Pass - II → 32, 27, 51, 66, 23, 13, 57, 85
27, 32, 51, 66, 23, 13, 57, 85
C = 6 27, 32, 51, 66, 23, 13, 57, 85
I = 4 27, 32, 51, 66, 23, 13, 57 85
27, 32, 51, 23, 66, 13, 57, 85
27, 32, 51, 23, 13, 66, 57, 85
27, 32, 51, 23, 13, 57, 66, 85
27, 32, 51, 23, 13, 57, 66, 85

2 . 0
1

Page No.
Date : / /

Pass III → 27, 32, 51, 23, 13, 57, 66, 85
27, 32, 51, 23, 13, 57, 66, 85
C = 5 27, 32, 51, 23, 13, 57, 66, 85
I = 2 27, 32, 23, 51, 13, 57, 66, 85
27, 32, 23, 13, 51, 57, 66, 85
27, 32, 23, 13, 51, 57, 66, 85

Pass IV → 27, 32, 23, 13, 51, 57, 66, 85
27, 32, 23, 13, 51, 57, 66, 85
C = 4 27, 23, 32, 13, 51, 57, 66, 85
I = 2 27, 23, 13, 32, 51, 57, 66, 85
27, 23, 13, 32, 51, 57, 66, 85

Pass V → 27, 23, 13, 32, 51, 57, 66, 85
23, 27, 13, 32, 51, 57, 66, 85
I = 2 23, 13, 27, 32, 51, 57, 66, 85
C = 3 23, 13, 27, 32, 51, 57, 66, 85

Pass VI → 23, 13, 27, 32, 51, 57, 66, 85
C = 2 13, 23, 27, 32, 51, 57, 66, 85
I = 1 13, 23, 27, 32, 51, 57, 66, 85.

Pass VII → 13, 23, 27, 32, 51, 57, 66, 85.

C = 1
I = 0

∴ comparisons = 7 + 6 + 5 + 4 + 3 + 2 + 1 = 28
Interchange = 5 + 4 + 2 + 2 + 3 2 + 1 + 0
= 16

Page No.
Date: / /

Q.2 FOREST

Pass I :- F O R E S T
F O R E S T
C = 5 F O R E S T
I = 1 F O E R S T
F O E R S T

Pass II :- F O E R S T
F O E R S T
C = 4 F E O R S T
I = 1 F E O R S T
F E O R S T

Pass III :- F E O R S T
E F O R S T
C = 3 E F O R S T
I = 1 E F O R S T

Pass IV :- E F O R S T
E F O R S T
C = 2 E F O R S T
I = 0

Pass V :- E F O R S T
C = 1 E F O R S T
I = 0

∴ Comparisons = 5 + 4 + 3 + 2 + 1 = 15
Interchange = 1 + 1 + 1 + 0 + 0 = 3

The complexity of bubble sort algorithm is O(

Page No.
Date : / /

• Bubble Sort (DATA, N)

Here DATA is an array with 'N' elements. this algorithm sort the elements in DATA.

BUBBLE (DATA, N)

1] Repeat step 2 and 3 for K = 1 to N-1 (Passes)

2] Set PTR = 1

3] Repeat while PTR ≤ N-K

a] if DATA [PTR] > DATA [PTR + 1] then interchange DATA [PTR] & DATA [PTR+1]

b] Set PTR = PTR + 1 (increment)

4] Exit

Q.1 P, E, O P L E

⇒ Pass - I → P E O P L E
E P O P L E
E O P P L E
E O P P L E
E O P L P E
E O P L E P

C = 5
I = 4

Pass - II → E O P L E P
O E P L E P
O O P L E P
O O L P E P
O O L E P P

C = 4
I = 2

Page No.
Date : / /

PASS - III → E O L E P P
E O L E P P
C = 3 E L O E P P
I = 2 E L E O P P
E L E O P P

PASS - IV → E L E O P P
E L E O P P
C = 2 E E L O P P
I = 1

PASS - V → E E L O P P
E E L O P P
C = 1
I = 0

Total Comparision = 5 + 4 + 3 + 2 + 1
= 15
Total Interchange = 0 + 1 + 2 + 2 + 4
= 9

The complexity of Bubble Sort Algorithm is $O(n^2)$

Q. TADOBA

⇒ We apply the bubble sort to the string TADOBA

Pass I :- T A D O B A

A T D O B A

A D T O B A

C = 5 A D O T B A

I = 5 A D O B T A

A D O B A T

A D O B A T

At the end of first pass the Biggest alphabet (T) Take it's positi

Page No.
Date : / /

Pass II :- A D O B A T
A D O B A T
C = 4 A D O B A T
I = 2 A D B O A T
A D B A O T

Pass III :- A D B A O T
A D B A O T
C = 3 A B D A O T
I = 2 A B A D O T

Pas IV :- A B A D O T
A B A D O T
C = 2 A A B D O T
I = 1

Pass V :- A A B D O T
A A B D O T
C = 1
I = 0

∴ Total comparesion = 1 + 2 + 3 + 4 + 5
= 15

Total Interchanges = 0 + 1 + 2 + 2 + 5
= 10

The complexity of bubble sort algorithim is $O(n^2)$

Page No.
Date : / /

* Multidimensional Array.

1] Two dimensional Array :-

A two dimensional m×n array A is a collection of m×n data elements such that each element is specified by a pair of integer called Subscript with the property that $1 \leq J \leq m$ and $1 \leq K \leq n$

The element of A with first Subscript J and second Subscript K will be denoted by A_{jk} or $A[J,K]$

Two dimensional array are called matrices in mathematics.

- Memory Representation of two dimensional Array.

int A [3,3] → (R C)

(1,1) ↓1	(1,2) ↓2	(1,3) ↓3 ← 1
(2,1)	(2,2)	(2,3) ← 2
(3,1)	(3,2)	(3,3) ← 3

Row	column
A(1,1)	A(1,1)
A(1,2)	A(2,1)
A(1,3)	A(3,1)
⋮	⋮
A(3,3)	A(3,3)

Page No.
Date : / /

Formulae :-

Column :-

$$Loc(A[J,K]) \Rightarrow Base(A) + W[M(K-1) + (J-1)]$$

Row :-

$$Loc(A[J,K]) \Rightarrow Base(A) + W[N(J-1) + (K-1)]$$

Ex. Consider

Formulae :

Column :

loc (A [J K]) ⇒ Base (A) + W [M (K-1) + (J-1)]

Row :

loc (A [J K]) ⇒ Base (A) + W [N (J-1) + (K-1)]

Ex. Consider [25. 4] Array score. suppose Base of score = 200 and there are W = 4 words per memory cell. Further more suppose programming language store 2D array using row major order. Then determine address of score [12 3]

⇒ Score [25 4]

loc [score (12 3)] = Base (score) + W [N (J-1) + (K-1)]

= 200 + 4 [4 (12-1)] + (3-1)]

= 200 + 4 [46]

= 384

- Multidimensional Array :

More than 2 dimensional is called as Multidimensional array

Ex. Suppose a 3-D array MAZE is use declaring

MAZE (2:8, -4:1, 6:10)

Then the length of 3 dimensional array MAZE (2:8, -4:1, 6:10)

PAGE NO.:

$L_1 = UB - LB + 1 \rightarrow 8 - 2 + 1 = 7$

$L_2 = 6 \quad L_3 = 5$

$L = L_1 \times L_2 \times L_3$

$L = 7 \times 6 \times 5$

$L = 210$

Suppose programming language store MAZE in memory row major order & suppose base of MAZE = 200 & w = 4 words per ~~mem~~ memory cell then identify address of MAZE [5 -1 8]

To identify address of MAZE (5 -1 8) will need effective indices

⇒ Base [MAZE] = 200 W = 4

MAZE = [5 -1 8]

$E_1 = 5 - LB = 5 - 2 = 3$

$E_2 = -1 - (-4) = 3$

$E_3 = 8 - 6 = 2$

Row major order

⇒ $(E_1 L_2 + E_2) L_3 + E_3$

$= (3 \times 6 + 3) 5 + 2$

$= 21 \times 5 + 2$

$= 107$

$LOC[MAZE(5 -1 8)] = Base(MAZE) + w(107)$

$= 200 + 4(107)$

$= 200 + 428$

$= 628$

Suppose multidimensional array A and B are declared using A(-2:2, 2:22) B(1:8, -5:5, -10:5)

1) Find length of each dimension of and no of element in A & B

2) Consider element B[3,3,3] Find effective indices

Base of B = 400, and w=4 words per memory cell using column major.

A(-2:2, 2:22)

→ $L_1 = 2-(-2)+1 = 5$

$L_2 = 22-2+1 = 21$

$L = 5 \times 21 = 105$

B(1:8, -5:5, -10:5)

$L_1 = 8-1+1 = 8$ $L_2 = 5-(-5)+1 = 11$

$L_3 = 5-(-10)+1 = 16$

$L' = L_1 \times L_2 \times L_3$

$L' = 8 \times 11 \times 16 = 1408$

$B = 3-2 = 2 = E_1$

$E_2 = 3+5 = 8$

$E_3 = 3+10 = 13$

Column major $= (E_3 L_2 + E_2) L_2 + E_1$

$= (13(11)+8)11+2$

$= 1663$

PAGE NO.:

1) Consider a multi-dimensional array X (-5:5, 3:33)
Y (3:10, 1:15, 10:20)
Find the length of each dimension and number of elements in X and Y.

2) Suppose Base(Y) = 400
w = 4 Determine effective indices E_1 E_2 E_3 and address of Y[5, 10, 15]
Assuming- 1) Y is stored in row major ~~measured~~ order
2) Y is stored in column major~~measured~~ order.

Ans 1) For X

$L_1 = 5-(-5)+1 = 11$

$L_2 = 33-3+1 = 31$

No of elements in X = $L_1 L_2$
$= 11 \times 31 = 341$

2) For Y

$L_1 = 10-3+1 = 8$

$L_2 = 15-1+1 = 15$

$L_3 = 20-10+1 = 11$

No of elements in Y = $L_1 L_2 L_3$
$= 8 \times 15 \times 11$
$= 1320$

Ans 2) For Effective indices

$E_1 = 5 - 3 (LB) = 2$

$E_2 = 10 - 1 = 9$

$E_3 = 15 - 10 = 5$

For Row major order

Row major $\Rightarrow (E_1 \cdot L_2 + E_2) L_3 +$

$\Rightarrow (2 \times 15 + 9) 11 + 5$

$= 39 \times 11 + 5$

$= 429 + 5$

$= 434$

Row ~~minor~~ major $Y[5\ 10\ 15]$

$=$ Base $(Y) + w(434)$

$= 400 + 4(434)$

$= 400 + 1736$

$= ~~736~~\ 2136$

order

For Column major $\Rightarrow (E_3 \cdot L_2 + E_2) L_1 +$

$= (5 \times 15 + 9) 8 + 2$

$= (75 + 9) 8 + 2$

$= 84 \times 8 + 2$

$= 672 + 2$

$= 674$

Column major $= Y[5\ 10\ 15]$

$=$ Base $(Y) + w(674)$

$= 400 + 4 \times 674$

$= 400 + 2696$

$= 3096$

Page No.
Date : / /

• Record Structure

Collection of Data are frequently organized into a higher feild, record and file. Specifically record is a collection of simillar data atom each of which is called feild or atribute. and a file is a collection of Simillar record. although a record is collection of data atom. it differs from the linear array in the following ways.
A record may be a collection of non-homogenous

② The Data atom in a record are index by atribute names, so their may not be a natural ordering of it's element

* e.g consider a hospital keeps a record on each new-born baby which contain the following data atoms :-
Name, gender, birth date, mother, father. Suppose that birth date is a group atom with subatom month, day, and year. father

Page No.
Date : / /

& mothee also aee geoup atom each coith sub atom name and age Deaco the Eeecoed steuctuee

Ans:- 1. Neeboen baby (20) → if mention the then wete

2. Name
2. Gendee
2. Bieth day
3. day
3. moth
3. Yeae
2. Fathee
3. Name
3. age
2. Mothee
3. Name
3. age

To accesses the infoemation of babibe:
Neeboen. fathee. Name. (6)

Q. The following is an list of entey with a level numbee in a Student Eecoed. One Student, Teoo numbee, Tcso name 3 lasst, 3 FIEST, 3 Middle name., 2 gendee, 2 biethday, 3 Day, 3 month, 3 yeae, 2 test, 3 math, 3 veebal

i) Deaco the coeeesponding

Page No.
Date : / /

Structure

ii) which of the atoms are elementary atoms.

Soln:- i)

1. Student
   2. Number
   2. Name
      3. Last
      3. first
      3. Middle
   2. Gender
   2. Birthday
      3. Day
      3. math
      3. Year
   2. Test
      3. Math
      3. Verbal

ii) => Number, Gender, Last, first middle, Day, month, year, Math, and verbal

Q. A college uses following structure for graduating class.

1. Student [20]
   2. Name
      3. First Name
      3. Middle Name
      3. Last Name

Date : / /

2. Major
2. Sat
3. Math
3. Verbal
2. GPA (4)
2. COM.

i) How many elementry atoms are their in the file.
ii) a) How does one accesses the major of eigth student.
iii) b) The sophomore GPA of 45th Student
iv) c) Find each output
a) Write ; Name [15]
b) Write : COM
c) Write : GPA [2]
d) Write : GPA [1,3]

Ans:- i) Since GPA is counted 4 times per student their are 11 elementry atom per students therefore the total no. of elementry atom in recoerd student = 11 x 200
= 2200

ii) Student. Major [8]
iii) Student. GPA [45, 2]
iv) → a) It will print the name of 16th Student.

Page No.
Date : / /

b)

- Representation of polynomial using array.

The polynomial expression $3x^4 + 5x^3 + 6x^2 + 10x - 14$ can be stored in a single dimenssion array as follows.

0	-14
1	10
2	6
3	5
4	3

- Pointer and pointer array.

Let DATA be any array a variable p is called pointer if p points to an element in data. i.e if ~~co~~ P

Page No.
Date : / /

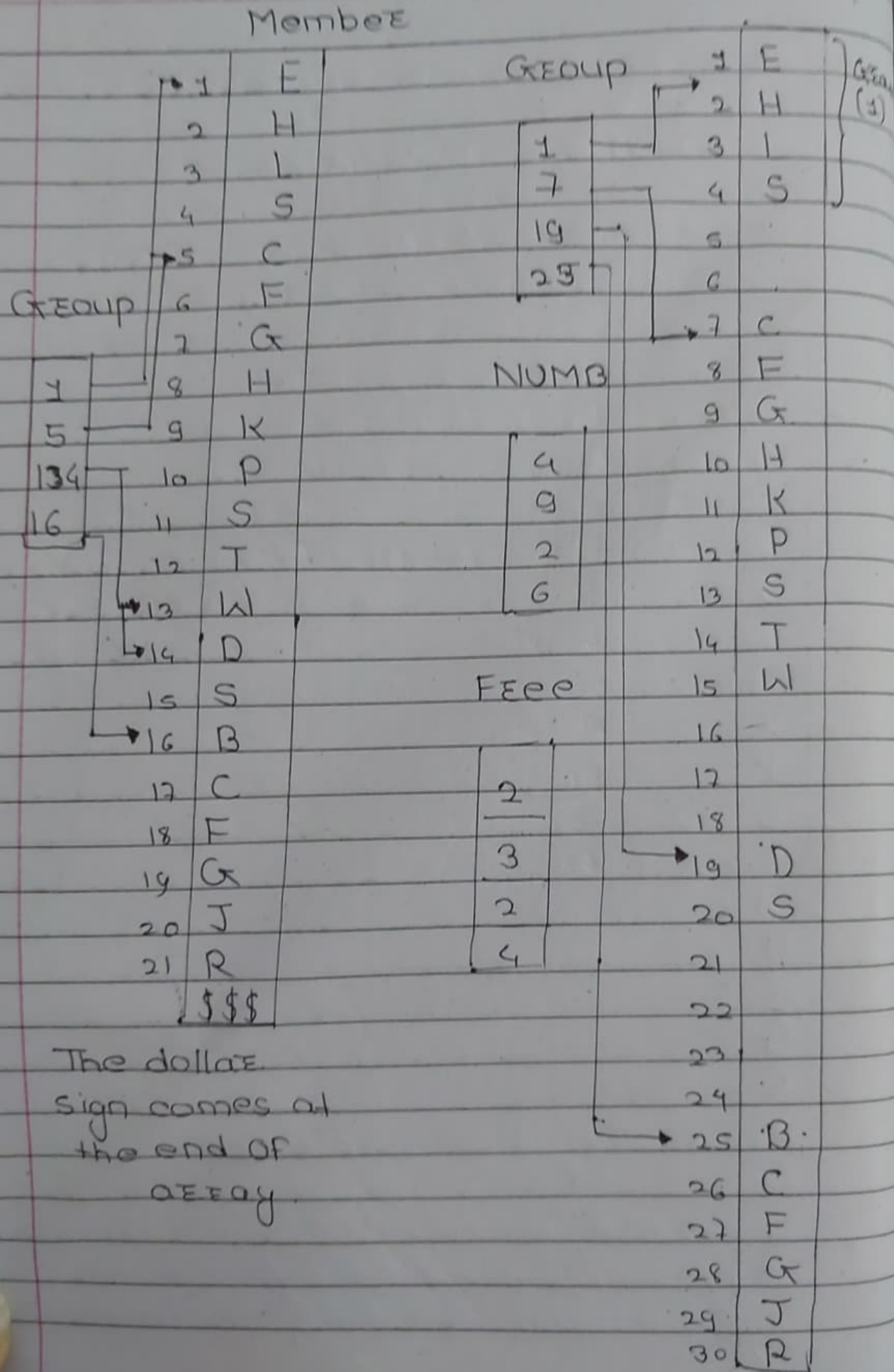
contain address of an element in DATA. and An array PTR is called called pointer array if each element of PTR is a pointer. pointer and pointer array are use to facillate the processing of the information in data. consider an organisation which divide it's membership into four groups. where each group contain an alphabetical list of those member living in certain area.

Group 1	Group 2	Group 3	Group 4
Even	Fonead	Davis	Baker
	felt	Segal	Coofer
Levis	glass		food
	Hill		Gray
	King		Jones
	Pew		Reed
	Silver		
	Tray		
	Wagner		

PAGE NO.:

Group 1	Group 2	Group 3	Group 4
Evan	Conrad	Davis	Baker
Harris	Felt	segal	Coofer
Levis	Glass		Ford
Shawn	Hill		Gray
	King		Jones
	Paen		Reed
	Silver		
	Troy		
	Wagner		

E	Evan
H	Harris
L	Levis
S	Shawn
C	$ $ $ → End of Group 1
F	C
G	F
H	G
K	H
P	K
S	P
T	S
W	T
D	W
S	$ $ $ → End of Group 2
:	D
:	S
:	$ $ $



The dollar sign comes at the end of array.