HindPhotostat

GET IT ON Google Play

MADE EASY , IES MASTER , ACE ACADEMY , KREATRYX

ESE , GATE, PSU BEST QUALITY TOPPER HAND WRITTEN NOTES
MINIMUM PRICE AVAILABLE @ OUR WEBSITE

1. ELECTRONICS ENGINEERING
2. ELECTRICAL ENGINEERING
3. MECHANICAL ENGINEERING
4. CIVIL ENGINEERING
5. INSTRUMENTION ENGINEERING
6. COMPUTER SCIENCE

IES ,GATE , PSU TEST SERIES AVAILABLE @ OUR WEBSITE

- IES –PRELIMS & MAINS
- GATE

➢ NOTE;- ALL ENGINEERING BRANCHS

➢ ALL PSUs PREVIOUS YEAR QUESTION PAPER @ OUR WEBSITE

PUBLICATIONS BOOKS -

MADE EASY , IES MASTER ,ACE ACADEMY ,KREATRYX ,GATE ACADEMY, ARIHANT ,GK

RAKESH YADAV, KD CAMPUS , FOUNDATION , MC –GRAW HILL (TMH) ,PEARSON...OTHERS

HEAVY DISCOUNTS BOOKS AVAILABLE @ OUR WEBSITE

F230, Lado Sarai New Delhi-110030 Phone: 9311 989 030	Shop No: 46 100 Futa M.G. Rd Near Made Easy Ghitorni, New Delhi-30 Phone:9711475393	F518 Near Kali Maa Mandir Lado Sarai New Delhi-110030 Phone: 9560 163 471	Shop No.7/8 Saidulajab Market Neb Sarai More, Saket, New Delhi-30

Website: www.hindPhotostat.com

Contact Us: 9311 989 030

Courier Facility All Over India

(DTDC & INDIA POST)

Array :

(Ex1) `int a[10] = {10, 20, 30, 40, 50 ..... 100}`

(declaration)

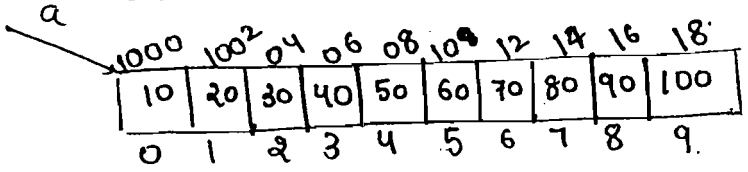
↓ identifier / variable / fn name.

iden

→ scan from left due to LL or LR parser (Top down, Bottom up)

`int a [10]` a is array of (size 10) having elements as integer.

winner



a is array which contain 10 elements where everyone is integer

integer = 2 Bytes

j - i → gives elements before j

j - i + 1 → gives elements including j

(Brackets) ( ) have Left to Right associative. (If come first it will be done).

X { `(int a) [10]` stop. a is a integer

print a : 1000 will print (array name print → gives Base address will print)

array name.

print variable name : print the value of that value

$$LOC(a[3]) = 1000 + (5-0) * 2 = 1010$$

for LOC → (&) reference.

* Dereference

$$LOC(a[9]) = 1000 + (9-0) * 2 = 1018$$

using this formula anyone can be accessed

⇓

Random access.

EX2 A[75 ..... 330]

starting index → 75 ; ending index → 330

330 − 75 + 1 = 276 elements

BA = 1000 , C = 10 (size)

[a[290]] = 1000 + ~~290*10~~ ..... (290 − 75)

1000 + ~~2900~~ ..... 215 * 10

(a[290]) = ~~3900~~ (3150) .

2150
1000
3150

EX3 A[-90 ... 500] (591)

500 − (−90) + 1 ⇒ 591

BA = 0 , C = 5 Bytes

LOC(A[393]) = 0 + (92 − 393) * 5

= 0 + (393 + 90) * 5

= 2415

393
92
485
*5
25

NOTE

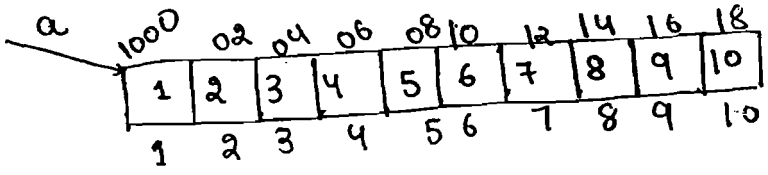
A[lb .... ub]

BA , C.

$$LOC[A[i]] = BA + (i - lb) * C$$

LOC(a[3]) = 1000 + (5 − 1) * 2 = 1008

extra.



NOTE.

By default array index start from 0 not from 1 because no need to calculate offset value (substraction)

(No need to perform extra substraction)

2D-Array:

$4-1+1 = 4$ rows

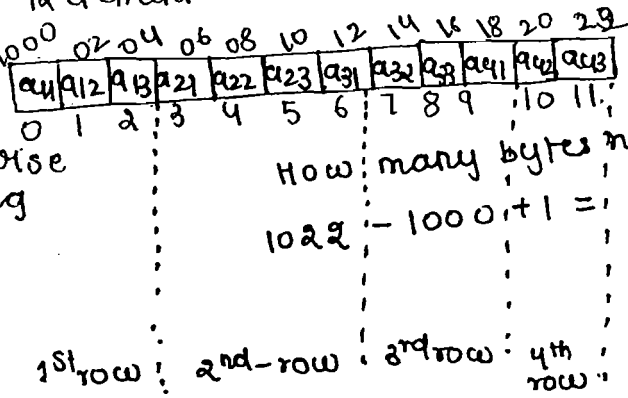
int A[1 ..... 4, 1 ..... 3]

$3-1+1 = 3$ columns.

	1	2	3
1	a_{11}	a_{12}	a_{13}
2	a_{21}	a_{22}	a_{23}
3	a_{31}	a_{32}	a_{33}
4	a_{41}	a_{42}	a_{43}

Row major order

$4 \times 3 = 12$ elements = 12 slots needed to store



$LOC(A[4][3]) = 1000 + 3*4(2)$

$= 1000 + 12*2$

$= 1024.$

(4-1) (3-1)
3 * 2 * 2
6 * 2
= 12

$LOC[A[4][3] = 1000 + [(4-1)*3 + (3-1)]*2$

$= 1022$

(3-1) * 4 (2)
2 * 8

Ans $LOC(a[2][3]) = 1000 + [(2-1)*3 + (3-1)]*2$

$= 1000 + [3 + 2]*2$

$1000 + 10$

$= 1010$

r
(3) * 3
9 + 2
11 * 2
1000 + 22
= 1022

(j-i) * j +

ex(2) $A = [29 \ldots 76, 93 \ldots 200]$.

row $\Rightarrow 76-29+1 = 48 \Rightarrow 108$ c.

BA = 1000, c = 10 Row major order

76 − 29 = 47; 200 − 93 = 107

$LOC(A[70][190]) = 1000 + [(70-29)*108 + (190-93)]*10$

$= 1000 + [4428 + 20]*10$

$= 1000 + 44480$

$= 45480.$

44480 + 1000 = 454

10 − 29 = 41; 108 − 41 = 108 x

432 / 4428

40 250

ex(3) $A[-200 \cdots +200, -300 \cdots -150]$

$200-(-200)+1$ $\quad -150-(-300)+1$

$-150+300+1$

$150+1$

$=151$

BA = 0 , C = 1 , RMO

$LOC[A[-3][-170]] = D + [(-3-(-200))*151 + [-170-(-300)]]$

$= 197*151 + 130$

$= 29747 + 130$

$= 29877$

200 − 3 = 197

300 − 5 = 295

300 − 170 = 130

197 × 151 = 29747

NOTE: $ub_1 - lb_1 + 1 = nr$

$A[lb_1 \cdots ub_1, lb_2 \cdots ub_2]$ nc

BA, C, RMO

$LOC(A[i][j])$

$= BA + [(i-lb_1)*nc + (j-lb_2)]*C$

NOTE:

column Major order = 48 = 108

ex(4) $A[29 \cdots 76, 93 \cdots 200]$

BA = 1000 , C = 10 CMO

$L(A[70][190]) = 1000 + [(190-93)*48 + (70-29)]*10$

$= 1000 + [87*48 + 41]*10$

$= 47970$

190 − 93 = 87

70 − 29 = 41

87 × 48

last element address in RMO = last elem add in CMO

$A = [-200 \cdots +200 \;(\Rightarrow 401),\; -300 \cdots -150 \;(\Rightarrow 151)]$

BA = 0 c = 1 cmo

$Loc(A[-3][-170]) = 0 + [(-170+300)*401 + (-3+200)]*1$

$= 130*401 + 197$

$= 52327$

300 − 170 = 130

401 − 13 = 203

$401 \times 130 = 52130$

$52130 + 197 = 52327$

NOTE: $A(lb_1 \cdots ub_1 \; lb_2 \cdots ub_2)$ BA c cmo $n_r = ?$

$$Loc(A[i][j]) = BA + [(j - lb_1) * n_r + (i - lb_1)]c$$

3D-array

Ex①

$A[23 \cdots 49, 2 \cdots 19, 11 \cdots 29]$

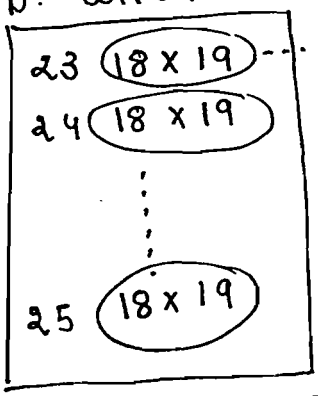
no of 2D = 49−23+1 = 26+1 = 27

$n_r = 19-2+1 = 17+1 = 18$

$n_c = 29-11+1 = 18+1 = 19$

$18 \times 19 \Rightarrow$ 2D.

3D: collections of 2D.



① collects of elements ⇓ 1D

② collection of 1D ⇓ 2D

BA = 1000, c = 10, RMO

size of this 18 x 19.

$Loc(A[40][15][20]) = 1000 + [(40-23)*27 + (15-2)*19 + (20-11)]*10$

40 − 23 = 17

27 − 9

$= 61700$