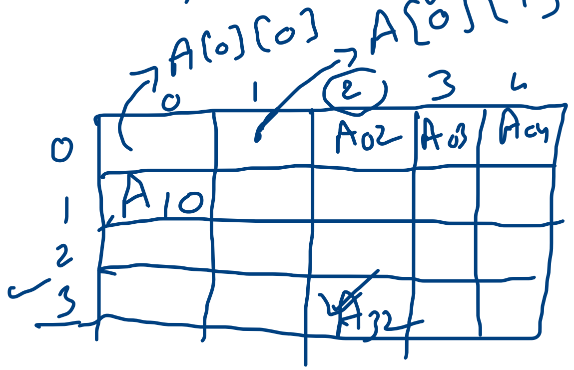
Unit-2:

Multidimensional Arrays:

2D Arrays

→ int a[4][5];

n rows x m columns.



$4 \times 5 = 20$

$r \times c$

1D-Arrays.

int age[10];

index/subscript

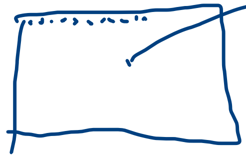
→ age[0]

age[1]

⋮

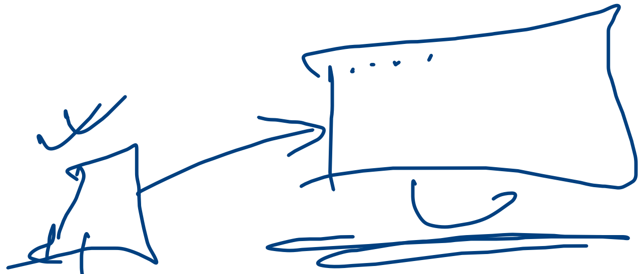
age[9]

2D Arrays:



→ (x, y)

- Excel.
- Mathematical objects → Matrices
- Frame Buffer.



Pic Def.

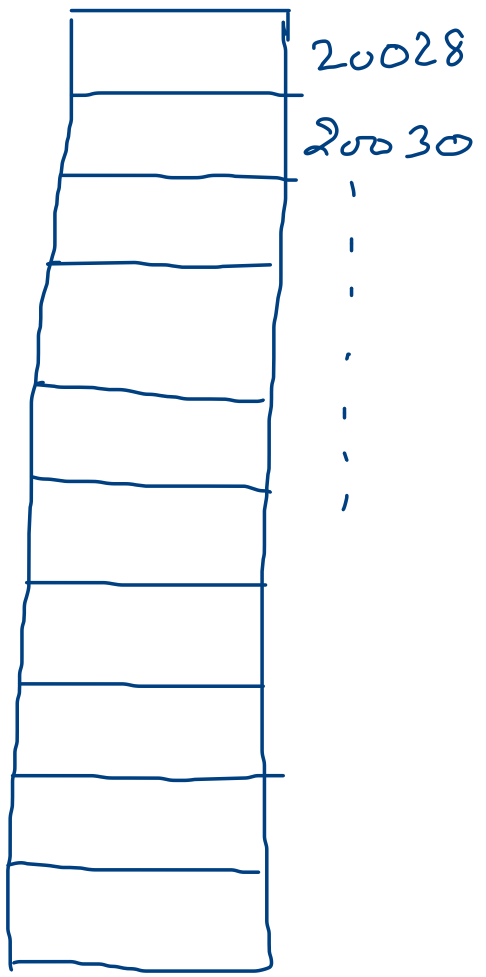
Representing 2D Arrays in Memory :

→ Row Major order.

→ Column Major order.

2D

a[4][5]



Rep. of 2D Arrays.

① Row Major order :

A

A11, A12, A33

	1	2	3	4
1	10	11	12	16
2	21	19	17	11
3	23	28	25	30

Logical representation

3×4 array A

$r = 3$

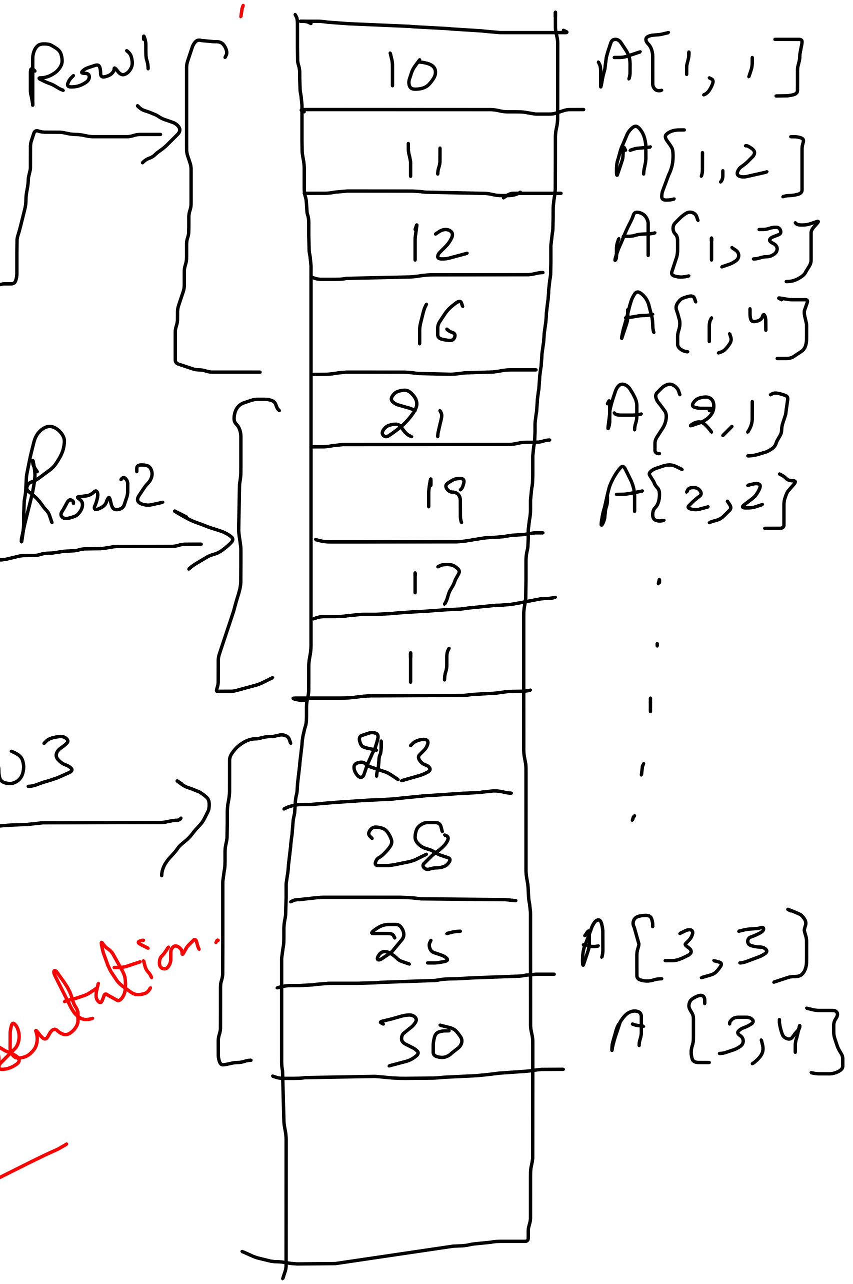
$C = 4$

Total elements $= 3 \times 4 = 12$

used in various HLLs.

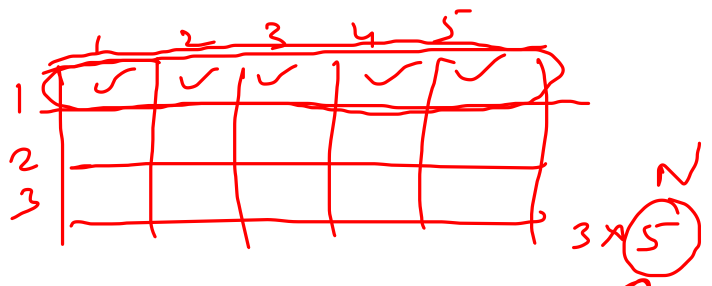
C, C++, Java, Pascal, Module2, Ada, etc.

mapping



Physical representation.

How to calculate the address of an element of a 2D-Array? in case of Row-Major order?



Let there be an $M \times N$ array A (ie $A[M, N]$)

Let it's base address be $Base(A)$

s = size of an element of the Array.

Q: To find address of specific element $A[i, j]$

(1) Address of i^{th} Row: ie How many rows to skip over ie $(i-1) \times$ size of a row

multiply

size of a row $\rightarrow N \times s$

$(i-1) \times N \times s$

(2) Address of j^{th} Column: $\boxed{(j-1) \times s}$

$$\text{Address of } A[i, j] = \boxed{Base(A)} + \boxed{(i-1) \times N \times s} + \boxed{(j-1) \times s}$$

$$= \boxed{Base(A) + s((i-1)N + (j-1))}$$

Formula:

$$\text{Address of } A[i,j] = \text{Base}(A) + s[(i-1)N + (j-1)]$$

Example:

$$\text{Address of } A[3,2] = 200 + 2[(3-1)4 + (2-1)]$$
$$= 200 + 2[(2)4 + 1]$$
$$= 200 + 2(8+1)$$
$$= 200 + 18$$
$$= 218$$

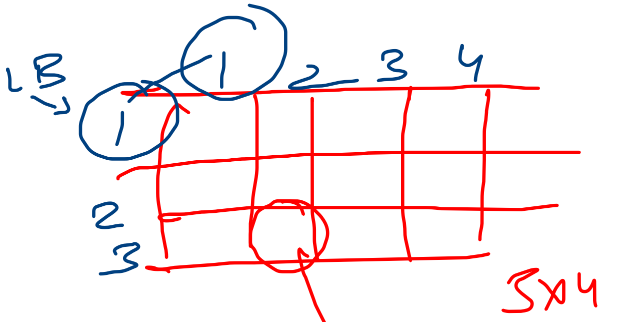
A

$A(0,0)$

$A(10, 18)$

LB UB

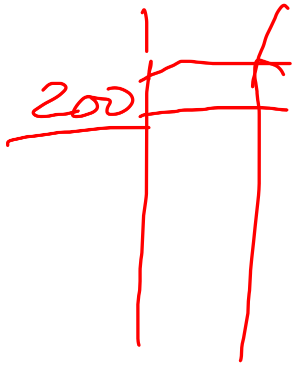
$A[3][4]$ M N



$A[3,2]$

$\text{Base}(A) = 200$

$s = 2$ bytes



General Formula:

$$\text{Address of } A[i,j] = \text{Base}(A) + s[(i-lb)N + (j-ub)]$$

2nd Method of Representing 2D Arrays in Memory:

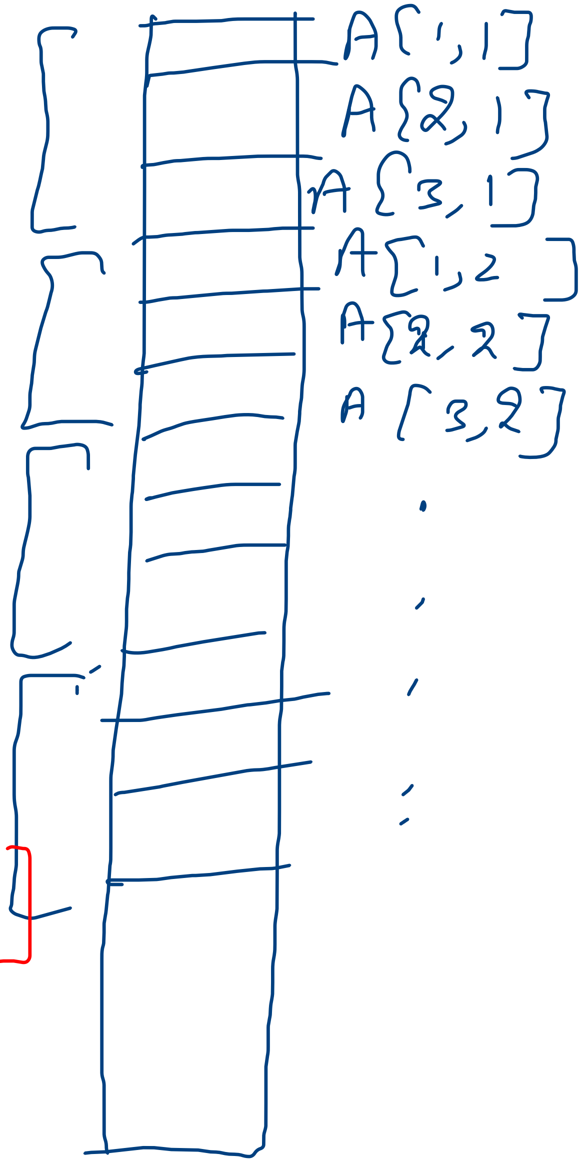
(2) Column Major Order.

column 1

$$\begin{matrix} 10 & 11 & 12 & 16 \\ 21 & 19 & 17 & 11 \\ 23 & 28 & 25 & 30 \end{matrix}$$

$M \times N$

3×4



Formula:

Address of $A[i,j] = \text{Base}(A) + s[(j-1)M + (i-1)]$

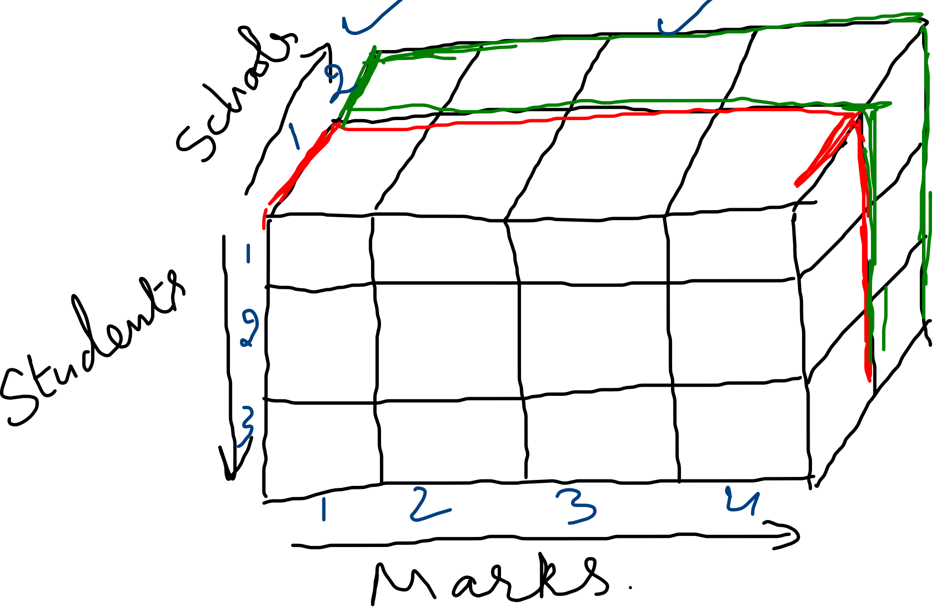
Eg: Address of $A[3,2] = 200 + 2[(2-1)3 + (3-1)]$

in column major order.

$= 200 + 2[3+2]$

$= 210$

3-D Arrays:



Marks[2, 3, 4]

Marks → M

School 1

Student 1	Marks[1,1,1]	M[1,1,2]	M[1,1,3]	M[1,1,4]
Student 2	M[1,2,1]	M[1,2,2]	M[1,2,3]	M[1,2,4]
Student 3	M[1,3,1]	M[1,3,2]	M[1,3,3]	M[1,3,4]

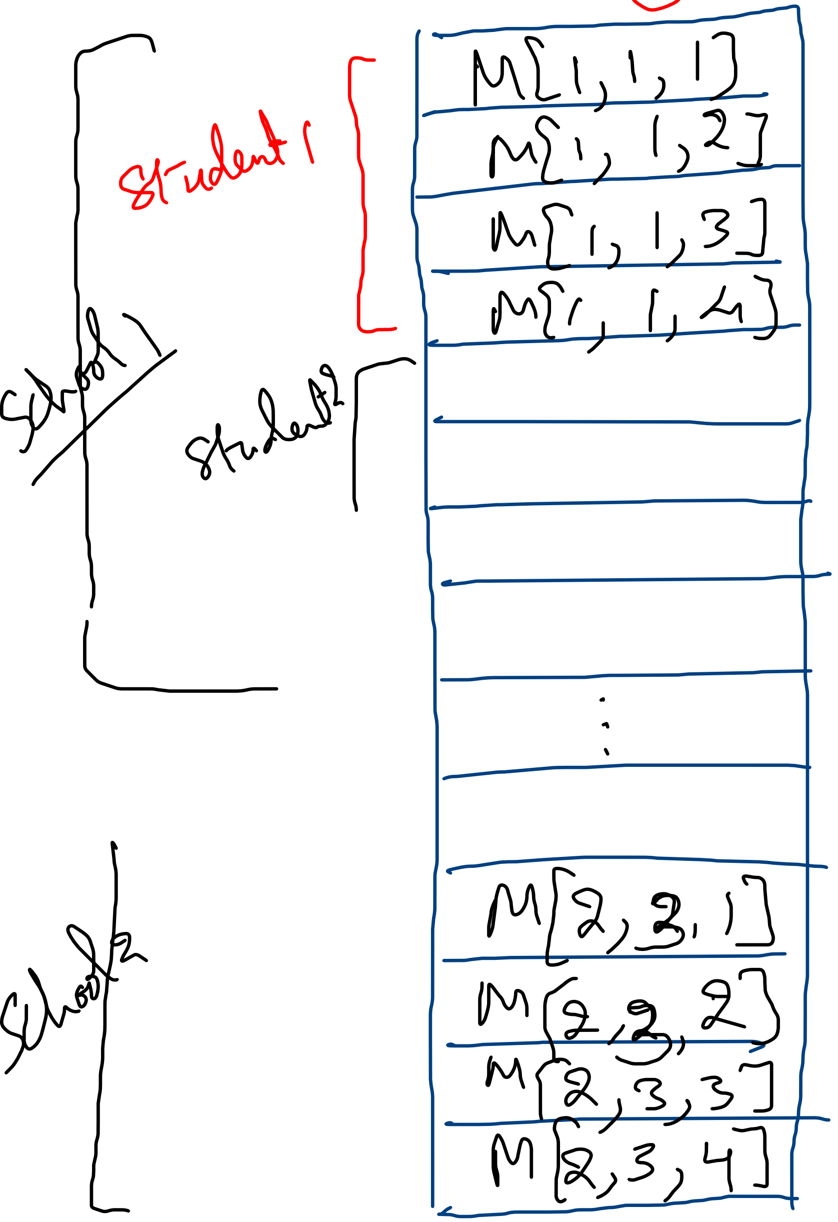
School 2

M[2,1,1]	M[2,1,2]	-	-
-	- -	- -	

complete it.

```
for(i=1; i<=M; i++)
{ for(j=1; j<=N; j++)
  { for(k=1; k<=Q, k++)
    {
      cin>> M[i][j][k];
    }
  }
}
```

Representing 3D Arrays in memory:



↓ Physical memory.

I) Row Major order

II) Column Major order

H/W.

Finding address of an element in 3D-Array.

$1 \le i \le M$, $1 \le j \le N$, $1 \le k \le Q$.

$(i-1)NQ + (j-1)Q$

Formula:

$$\text{Address of } A[i,j,k] = \text{Base}(A) + s\left[((i-1)N + (j-1))Q + (k-1)\right]$$

eg: $A[2,2,3]$ (i, j, k)

$B(A) = 200$

$s = 2$

$M = 2$

$N = 3$

$Q = 4$

$$= 200 + 2[((2-1)3 + (2-1))4 + (3-1)]$$

$$= 200 + 2((3+1)4 + 2]$$

$$= 200 + 2[16 + 2]$$

$$= 200 + 2(18) = 236$$

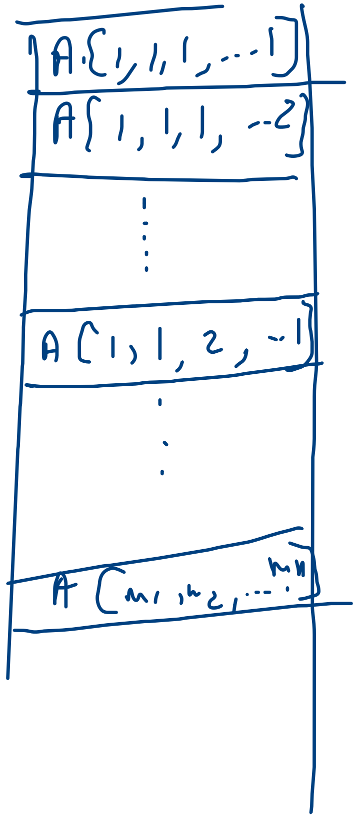
Generalization of Multidimensional Array:

↳ n-D array.

Range: $1 \le k_1 \le m_1,\ 1 \le k_2 \le m_2,\ \ldots\ 1 \le k_n \le m_n$.

Element: $A[k_1, k_2, \ldots k_n]$

Memory Rep. (Row Major order)



Formula for finding address of an element.

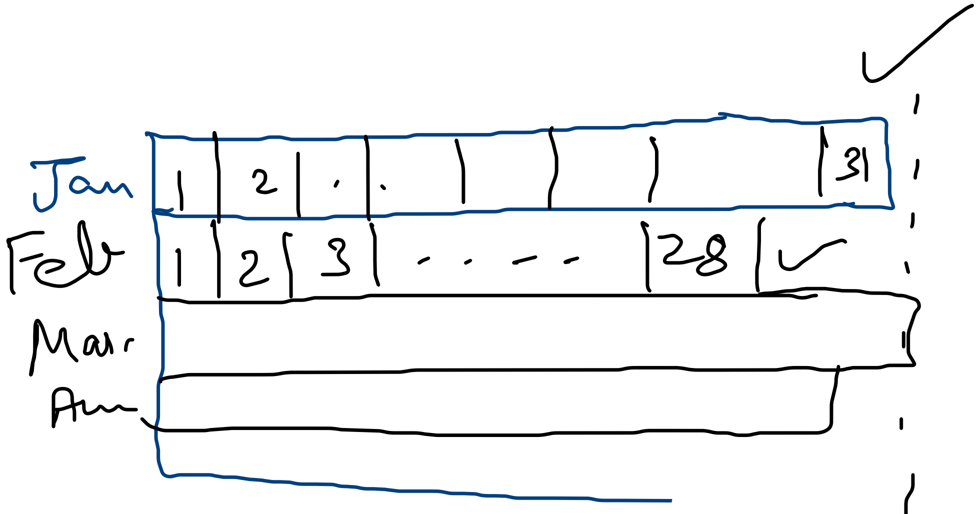
$$A[k_1, k_2, \ldots k_n] = \text{Base}(A) + \delta\left[(k_1-1)\, m_2 m_3 \cdots m_n + (k_2-1)\, m_3 m_4 \cdots m_n + \cdots + (k_{n-1}-1)\, m_n + (k_n-1)\right]$$

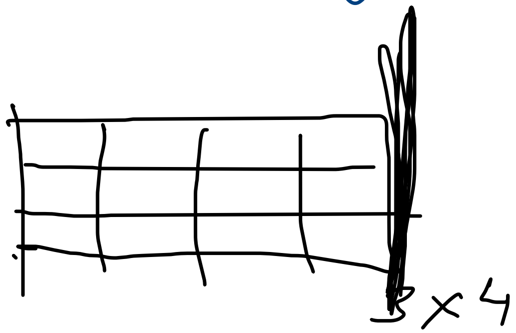
$$= \text{Base}(A) + \delta\left[\sum_{1 \le i \le n} p_i\,(k_i - 1)\right]$$

$$p_i = \prod_{i < x \le n} m_x$$

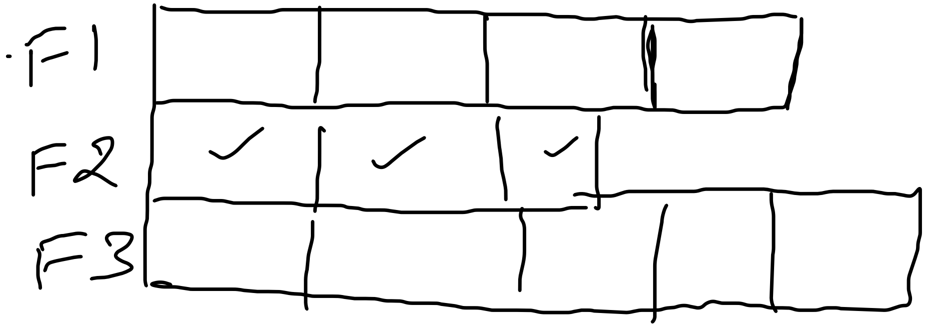
Jagged Arrays and Pointer Arrays:

eg:- Calander for a partialar year.





An array in which no. of elements in each row <u>need not</u> be of same size is called a jagged array.



Storing Jagged arrays in Memory: (We must know about: Pointer and array of pointers)

P is a var that points to an element of an array.

→ contains address of an element of an array.

PTR is an array of pointers.