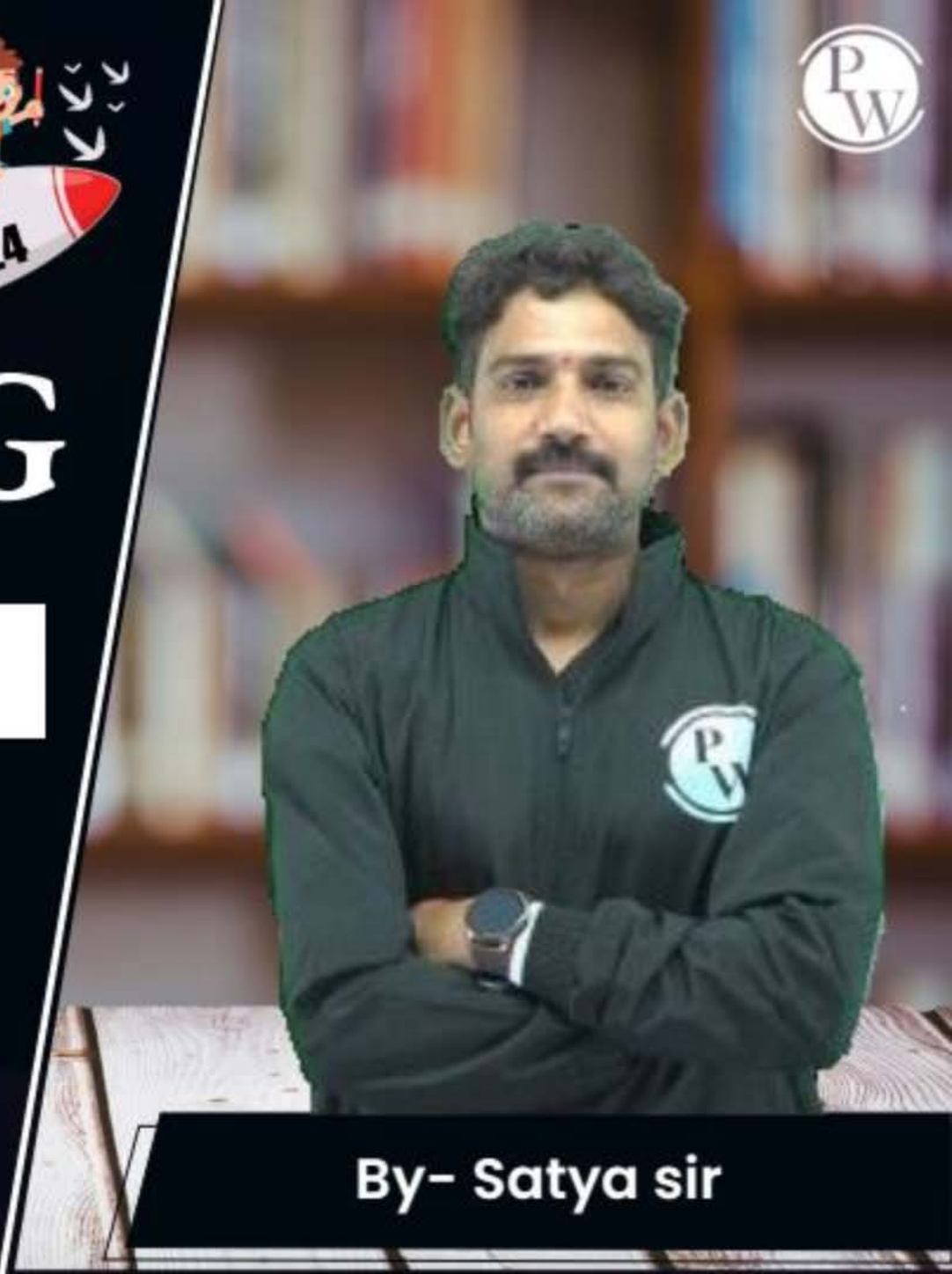
CS & IT ENGINEERING

'C' Programming

Pointers & Arrays

Lecture No.- 05



By- Satya sir

Recap of Previous Lecture

- String Handling Functions
- 2-D arrays
  - 2-D array Declaration
  - 2-D array Initialization

Topics to be Covered

- Address of 2-D array Element
- Access Elements of 2-D array
- Pointers & Arrays

Topic : 2-D Arrays

Let Base address = 100

Ex: Let 1 int = 4Bytes

```
int x[3][4] = {5,10,15,20,25,30,35,40,45,50,55,60};
```

Representation

x	0	1	2	3
	100	104	108	112
0	5	10	15	20
	116	120	124	128
1	25	30	35	40
	132	136	140	144
2	45	50	55	60

- In Memory, array elements always store in Consecutive locations.
- 2-D array elements can be stored in either of 2-ways:
  1) Row-Major Order (RMO) – Row-wise C, C++, Java
  2) Column-Major Order (CMO) – Col-wise FORTRAN, MATLAB, SCILAB

RMO

[0][0]	[0][1]	[0][2]	[0][3]	[1][0]	[1][1]	[1][2]	[1][3]	[2][0]	[2][1]	[2][2]	[2][3]
5	10	15	20	25	30	35	40	45	50	55	60
100	104	108	112	116	120	124	128	132	136	140	144

48 Bytes

CMO

[0][0]	[1][0]	[2][0]	[0][1]	[1][1]	[2][1]	[0][2]	[1][2]	[2][2]	[0][3]	[1][3]	[2][3]
5	25	45	10	30	50	15	35	55	20	40	60
100	104	108	112	116	120	124	128	132	136	140	144

48 Bytes

Topic : 2-D Arrays

Find address of 2-D array Element

Let n = Size of Each Element (Scale factor), 'A' be array

i = row index, j = col index

r = Total rows, c = Total cols

B = Base address

row indexing starts from 'x', col indexing starts from 'y'

RMO $\quad loc(A[i][j]) = B + \{[(i-x)*c] + (j-y)\} * n$

CMO $\quad loc(A[i][j]) = B + \{[(j-y)*r] + (i-x)\} * n$

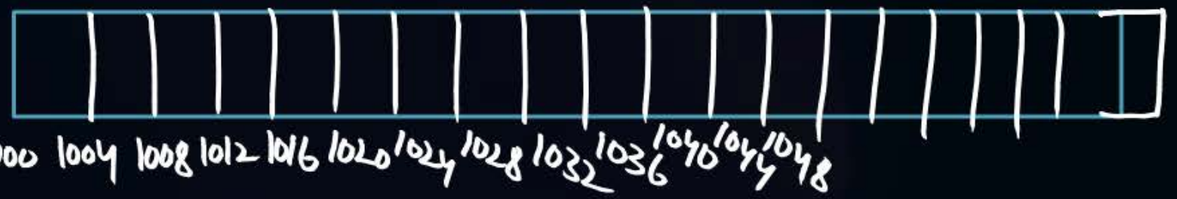
Topic : 2-D Arrays

$$RMO = B + [(i-x)*c + (j-y)]*n$$
$$CMO = B + [(j-y)*r + (i-x)]*n$$

Example :

Let row index starts from -2, $\Rightarrow x = -2$
Col index starts from -7 $\Rightarrow y = -7$

int $A[^{r}_{4}][^{c}_{5}]$; $B = 1000$



f A[0][-4]

$RMO = 1000 + [[0-(-2)]*5 + -4-(-7)]*4$
$= 1000 + (10+3)*4 = 1052$
$CMO = 1000 + [[3*4]+2]*4 = 1056$

f A[-1][-3]

$RMO = 1000 + [[-1-(-2)]*5 + -3-(-7)]*4$
$= 1000 + (1*5+4)*4 = 1036$
$CMO = 1000 + [(-3-(-7))*4 + -1-(-2)]*4$
$= 1000 + [4\times4+1]\times4 = 1068$

f A[-2][-5]

H|W

Topic : 2-D Arrays

Accen 2-D array Elements

```
int x[10][20];
int r, c, i, j;
Printf("Enter No of rows & cols");
scanf("%d %d", &r, &c);
// i/p values
 for(i=0; i<r; i++)
    for(j=0; j<c; j++)
    {
      Printf("\nEnter x[%d][%d] value", i, j);
      Scanf("%d", &x[i][j]);
    }
```

```
i=0   j=0, 1, 2, 3, ---
i=1   j=0, 1, 2, 3, ---
i=2   j=0, 1, 2, 3 ---
```

```
// Printing array values
  for(i=0; i<r; i++)
     for(j=0; j<c; j++)
        Printf("\nx[%d][%d] value is %d", i, j, x[i][j]);
```

Topic : Arrays and Pointers - 1

Practice Programs

1. Matrix Addition
2. Matrix Multiplication
3. Transpose of given Matrix
4. Inverse of given Matrix
5. Find Maximum and minimum Element of array (1-D & 2-D arrays)

:
:

Topic : Arrays and Pointers - 1

- Every array is a Constant Pointer.

- Array will get memory at Compile-time, Static
- Pointer will get Memory at Run-time, Dynamic

- array cannot be dereferenced, while Pointer can be.

- Arrays and Pointers, together can be implemented in 3 ways:

1) Pointer to an Element of array
2) Pointer to whole array (Array Pointer)
3) Array of Pointers

datatype *Pointer

datatype (*Pointer)[size]

datatype *Pointer[size]

Ex:

```
int *p;

int (*p)[5];

int *p[5];
```

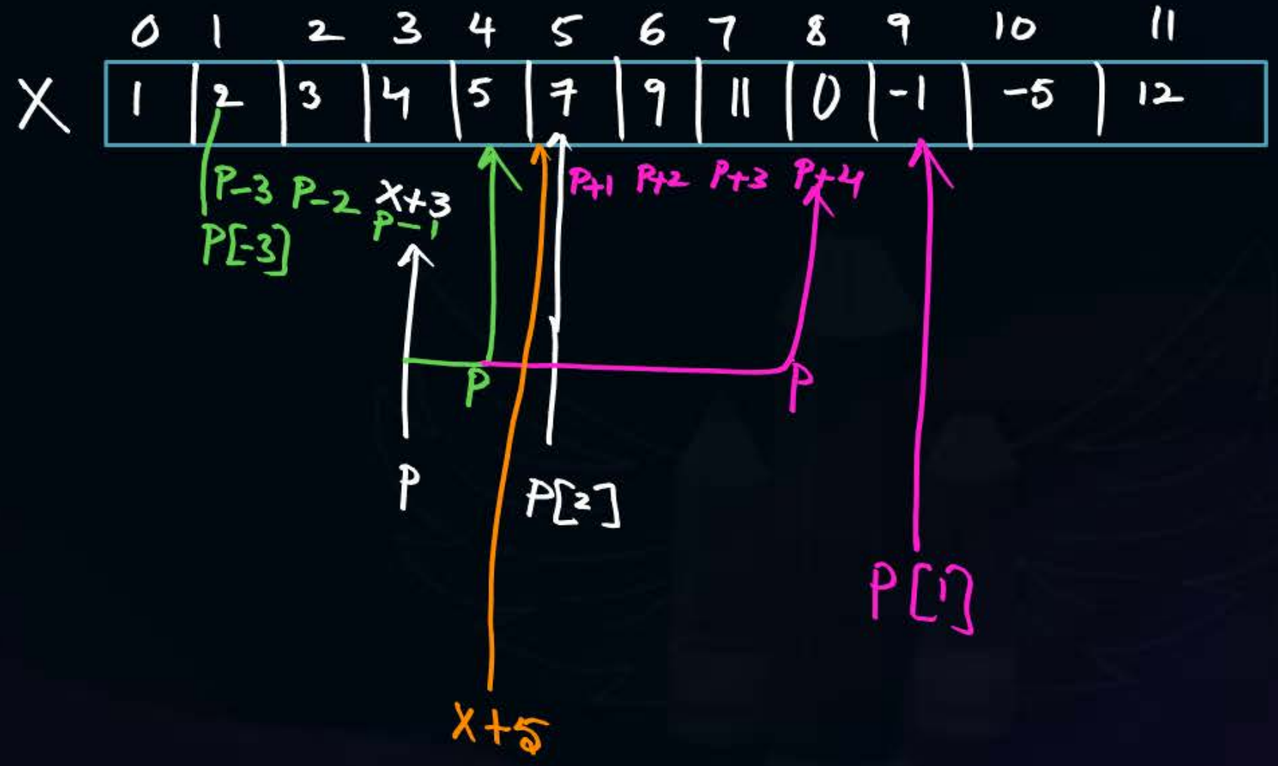
Topic : Arrays and Pointers - 1

Examples :

① `int x[ ] = {1, 2, 3, 4, 5, 7, 9, 11, 0, -1, -5, 12};`

```
int *p = x+3;

Printf("%d", p[2]);   //7

p++;

Printf("%d", p[-3]);  //2

p = p+4;

Printf("%d", p[1]);   //-1

Printf("%d", *(x+5));  //7
```

o/p: 7, 2, -1, 7

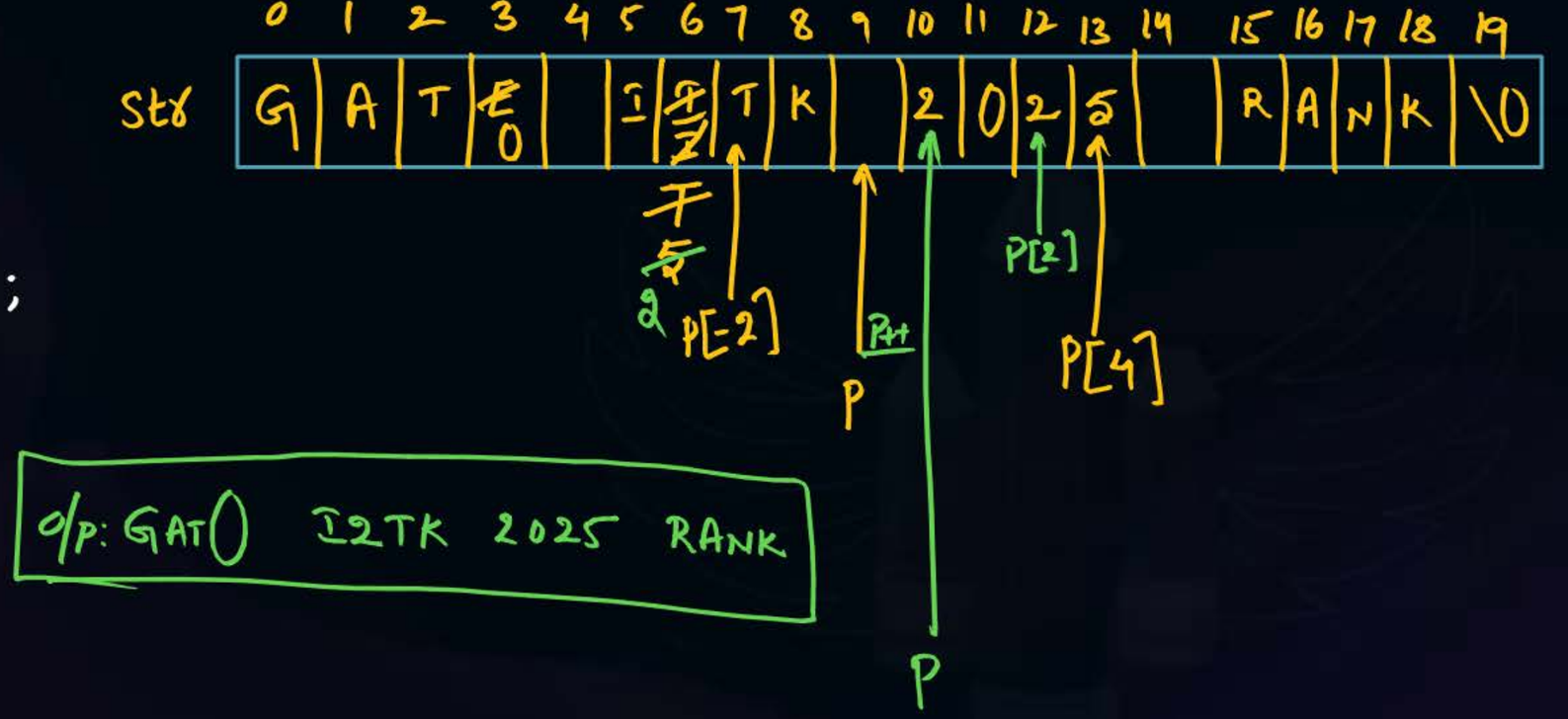


Topic : Arrays and Pointers - 1

Example-2

```
int i=6;
Char Str[ ] = "GATE IITK 2025 RANK";
Char *p = Str+9;
Str[3] = Str[11];
Str[6] = Str[5]   Str[i--] = Str[i++];
Str[6] = 'T'      Str[++i] = P[-2];
Str[6] = '5'      Str[i] = P[4];
P[-4] = '2'       P++;
                  P[-4] = P[2];
Printf("%s", Str);
```

Single space



o/p: GAT() I2TK 2025 RANK

2 mins Summary

- Finding address of 2-D array element
- Access 2-D array elements
- Pointers & Arrays

To be contd ... @ TOMORROW @ 6PM

THANK - YOU