CS & IT

[redacted]

ENGINEERING

C Programming

Pointers & Arrays

Lecture No.- 06

By- Satya sir

Recap of Previous Lecture

- 2-D Arrays memory layout
  - RMO
  - CMO
- Pointers & Arrays
  - Pointer, to an individual element of array ✓
  - Pointer to whole array
  - Array of Pointers

Topics to be Covered

- PYQ solving
- Pointers & 2-D arrays
- Array of Pointers

Topic : Arrays and Pointers - 2

#Q. What does the following fragment of C-program print?

```
char c[] = "GATE2011";
char *p =c;
printf("%s", p + p[0] - p[3]) ;
```

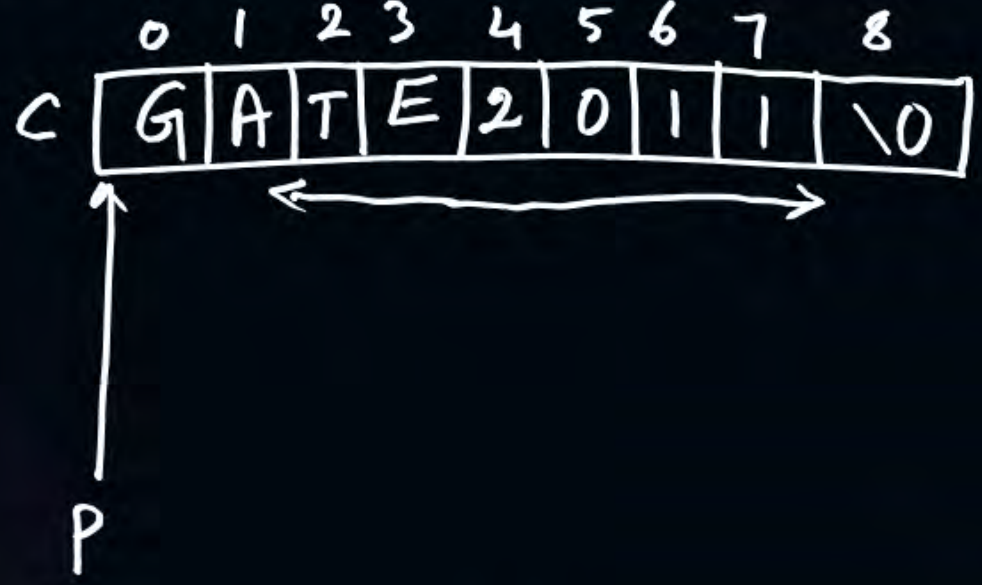
$p + \text{'G'} - \text{'E'}$

(A) GATE2011

(B) ATE2011 $\quad p + 71 - 69$

[✓ (C) TE2011] $\quad p+2$

(D) 2011



0	1	2	3	4	5	6	7	8
G	A	T	E	2	0	1	1	\0

$p[0] = *(p+0) = \text{'G'}$

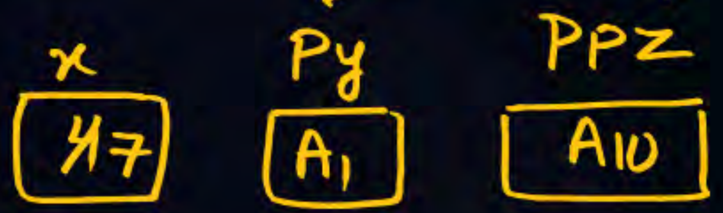
$p[3] = *(p+3) = \text{'E'}$

GATE 2011

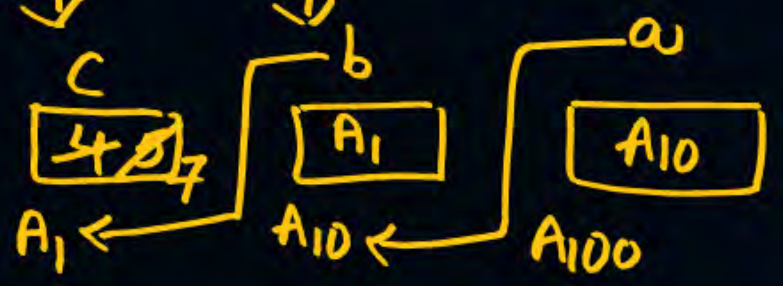
Topic : Arrays and Pointers - 2

#Q. What is printed by the following C program?

```
#include <stdio.h>
int f(int x, int *py, int **ppz)
{
  int y, z;
  **ppz += 1;
  z  = **ppz;      z = 5
  *py += 2;
  y = *py;         y = 7
  x += 3;x=x+3     x = 7
  return x + y + z;
}                  7+7+5 = 19
```



```
void main()
{
  int c, *b, **a;
  c = 4;
  b = &c;
  a = &b;
  printf( "%d", f(c,b,a));
  getchar();         returned 19
}
```



A. 18

B. 19 ✓

C. 21

D. 22

GATE 2008

Topic : Arrays and Pointers - 2

Pointers & 2-D arrays

Let B=1000, 1 int = 4 Bytes

int A[3][4] = { 2,4,6,8,10,12,14,16,18,20, 22,24 };

A+0 = 1000 = &A[0][ ]

A+1 = 1016 = &A[1][ ]

A+2 = 1032 = &A[2][ ]

$$\boxed{A+i = \&A[i][\ ] \text{ address of Row } i}$$

A => 1000

(A+0)+1 => 1004 == &A[0][1]

(A+1)+2 => 1024 = &A[1][2]

(A+2)+3 => 1044 = &A[2][3]

$$\boxed{(A+i)+j \Rightarrow \&A[i][j]}$$

A	0	1	2	3
0	1000 2	1004 4	1008 6	1012 8
1	1016 10	1020 12	1024 14	1028 16
2	1032 18	1036 20	1040 22	1044 4

X = A+0

0	1	2	3
2	4	6	8
1000 x+0	1004 x+1	1008 x+2	1012 x+3

Y = A+1

0	1	2	3
10	12	14	16
1016 y+0	1020 y+1	1024 y+2	1028 y+3

Z = A+2

0	1	2	3
18	20	22	24
1032 z+0	1036 z+1	1040 z+2	1044 z+3

Topic : Arrays and Pointers - 2

$$\boxed{A+0+1 \neq (A+0)+1}$$

$A+1 \neq (A+0)+1$

$\&A[1][\,] \neq \&A[0][1]$

$1016 \neq 1004$

int A[3][4] = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24};

$$\boxed{A+i+j \neq (A+i)+j}$$

A	0	1	2	3
0	1000 2	1004 4	1008 6	1012 8
1	1016 10	1020 12	1024 14	1028 16
2	1032 18	1036 20	1040 22	1044 24

$\frac{A+0}{x}$	0	1	2	3
	2	4	6	8

1000 ↑ P

A+1	0	1	2	3
	10	12	14	16

1016

A+2	0	1	2	3
	18	20	22	24

1032

1-D array row 0

$A+0 == *(A+0) ==$ Pointer Pointing to row 0 == 1000

1-D array Row-1

$A+1 == *(A+1) ==$ Pointer Pointing to row1 == 1016

1-D array Row 2

$A+2 == *(A+2) ==$ Pointer Pointing to row2 == 1032

$$\boxed{\underline{A+i} == *(A+i) == \text{Pointer Pointing to Row } i == \&A[i][\,]}$$

Name of 1-D array, Row i

Topic : Arrays and Pointers - 2

$(A+i) == *(A+i) == \&A[i][\ ]$ = Address of Row i

$(A+i)+j == \&A[i][j]$ = Address of Row i, col j

$*(A+i)+j == \&A[i][j]$ = Address of Row i, col j

$*(*(A+i)+j) == A[i][j]$ = value at Row i, col j

$$\underset{\text{Value at address}}{\underline{*}}\left(\underset{\text{Pointer to Row } i}{\underline{*}}(A+i)+j\right)$$

Topic : Arrays and Pointers - 2

**x == *(*(x+0)+0) = x[0][0]

*x = *(x+0)

Example

int x[4][5] = { { 10, 15, 5, 9, 7 },
{ 1, 3, 5, 7, 9 },
{ 2, 4, 6, 8, 10 },
{ 11, 22, 33, 44, 55 } };

B = 2000

	col 0	1	2	3	4
Row 0	2000 10	2004 15	2008 5	2012 9	2016 7
1	2020 1	2024 3	2028 5	2032 7	2036 9
2	2040 2	2044 4	2048 6	2052 8	2056 10
3	2060 11	2064 22	2068 33	2072 44	2076 55

x+1 == Row 1 == 2020

*x +1 == *(x+0) +1 == 2000 + 1*4 == 2004

**x+2 == *(*(x+0)+0) +2 = x[0][0] +2 = 10+2 = 12

*(*x+1)+2 == *(*(x+0)+1) +2 = x[0][1]+2 == 15+2 = 17

((x+1)+2) == x[1][2] == 5

x+2+6 ≠ (x+2)+6 → == x+8 ⇒ Row 8 Base address
2160
~~(out of bounds)~~

*(x+2)+6 == 2040 + 6*4 = 2064 ≅ &x[3][1]

*(*x+3) == *(*(x+0)+3) = x[0][3] == 9

**x+2+1 == **x+3 = 10+3 = 13

(**x+2)+1 == (10+2)+1 = 13

Topic : Arrays and Pointers - 2

#Q. What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires four bytes of memory.

```
#include <stdio.h>
int main()
{
  unsigned int x[4][3] = {{1, 2, 3}, {4, 5, 6},
                          {7, 8, 9}, {10, 11, 12}};
  printf("%u, %u, %u", x+3, *(x+3), *(x+2)+3);
}
```

Row 3
=2036

2036

2024+3*4
=2036

	0	1	2
0	2000 1	2004 2	2008 3
1	2012 4	2016 5	2020 6
2	2024 7	2028 8	2032 9
3	2036 10	2040 11	2044 12

*(x+2) → Pointer to Row2

x+3

*(x+3)
Pointer to Row 3

[✓ A. 2036, 2036, 2036]

B. 2012, 4, 2204

C. 2036, 10, 10

D. 2012, 4, 6

GATE 2015

Topic : Arrays and Pointers - 2

#Q. What will be the output of the C program, if executed on a 32-bit processor?

```
#include<stdio.h>
void function(char**);
→int main()
{       char *arr[] = { "ant", "bat", "cat", "dog", "egg", "fly" };
        function(arr);
        return 0;               }
void function(char **ptr)             {
        char *ptr1;
        ptr1 = (ptr += sizeof(int)[-2] - 2);
        printf("%s\n", ptr1);         }
```

Ptr1 = Ptr+0

$$\begin{aligned} Ptr &= ptr + sizeof(int)[-2] - 2 \\ &= Ptr + 4[-2] - 2 \\ &= Ptr + (4-2) - 2 \\ &= Ptr + 0 \end{aligned}$$

Ptr → arr

Ptr1 → ptr+0

arr	0	1	2	3
0	a	n	t	\0
1	b	a	t	\0
2	c	a	t	\0
3	d	o	g	\0
4	e	g	g	\0
5	f	l	y	\0

A. bat B. cat [✓ C. ant] D. dog

Topic : Arrays and Pointers - 2

```
int x[5] = {11,22,33,44,55}, *P, (*q)[5];
```

→ q is Pointer, Pointing to array of 5 Elements, Array Pointer.

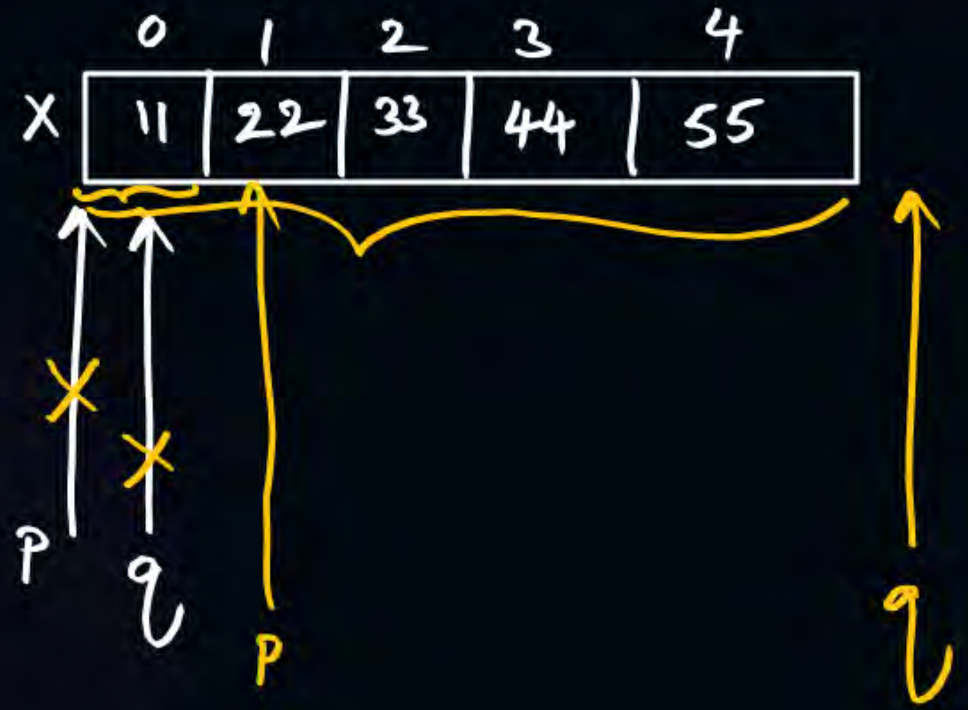
Let
B = 1000
1 int = 4 Bytes

```
P = x;

q = &x;
```

```
Printf("%u", P); //1000
Printf("%u", q); //1000

P++; //1004
q++; //1020
```



- P Points to only an Element of array
- q Points to whole array.

2 mins Summary

- PYQ Practice
- Pointers with 2-D arrays
- Pointer to an Element
- Pointer to whole array

To be contd . . . ☺

THANK - YOU